



Extend Salesforce with Clicks, Not Code



© Copyright 2000–2026 salesforce.com, inc. All rights reserved. Salesforce is a registered trademark of salesforce.com, inc., as are other names and marks. Other marks appearing herein may be trademarks of their respective owners.

CONTENTS

Extend Salesforce with Clicks, Not Code	1
Setup Home Page	3
Customize Your Salesforce Org	7
Set Up Your Data Your Way.....	170
Build Your Own Salesforce App	599
Get Help for Lightning App Builder.....	617
Manage Your Notifications with Notification Builder.....	712
Custom Domains	724
Extend the Reach of Your Organization	824
Build Your Own Web Site	1379
Develop and Deploy with DX Inspector.....	1670
Organization Sync	1681
Resources for the Point & Click Administrator.....	1682

Extend Salesforce with Clicks, Not Code

Ready to go beyond the basics of Salesforce administration? Want to customize your org, push its boundaries, and enhance its functionality? You can do that and so much more without writing a single line of code. All you need is your mouse and a sense of adventure. Enhance your objects, data, and fields, customize your org's look and feel, augment your business processes, create websites, and even create apps—all using point-and-click tools.

Setup Home Page

The Home page in Setup is the jumping off point for customizing your org, finding important setup tools and information, and building and managing applications.

Customize Your Salesforce Org

You can customize each of the standard tabs and types of records, including adding custom fields and setting page layouts. You can also customize search, tagging, and user interface options for your org. In addition, every Contact Manager, Group, Professional, Enterprise, Unlimited, and Performance Edition user can customize various personal display options.

Set Up Your Data Your Way

Optimize your Salesforce data to fit the unique needs of your users. You can create your own objects with data that fits together in the ways that make the most sense for you.

Build Your Own Salesforce App

An *app* is a collection of items that work together to serve a particular function. Salesforce apps come in two flavors: Classic and Lightning. Classic apps are created and managed in Salesforce Classic. Lightning apps are created and managed in Lightning Experience. You can customize both types of app to match the way your users work.

Manage Your Notifications with Notification Builder

Keep your users in the know with timely notifications, whether they're at their desks or on the go. Create custom notifications to give your users new information and reminders. Choose whether Salesforce notifications appear on desktop, mobile, Slack, or at all.

Custom Domains

Provide a branded experience for users who access your external-facing Salesforce content by serving your Digital Experiences or Salesforce Sites on a domain that you own, such as <https://www.example.com>.

Extend the Reach of Your Organization

Sometimes your users need to work with data and services that are outside your Salesforce org. There's a variety of ways you can provide seamless access across org boundaries.

Build Your Own Web Site

Site.com and Salesforce Sites are legacy systems to create sites using Salesforce.

Develop and Deploy with DX Inspector

DX Inspector is a management tool that provides quick access to metadata change tracking and visualization. It's available at the top of a page or builder in your Salesforce sandbox or scratch org.

[Organization Sync](#)

This feature is now retired. In previous releases, Organization Sync let you set up a secondary, synced Salesforce org. Users could access a subset of Salesforce data in that org when the primary org was unavailable.

[Resources for the Point & Click Administrator](#)

In addition to online help, Salesforce creates guides and tip sheets to help you learn about our features and successfully administer Salesforce.

Setup Home Page

The Home page in Setup is the jumping off point for customizing your org, finding important setup tools and information, and building and managing applications.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions except **Database.com**

You access the Home page from Setup.

In Lightning Experience:

- The left sidebar, which you can browse or search, provides access to all setup actions, tasks, and tools.
- The Quick Find lets you quickly navigate to any node using a keyword. Quick Find is the best way to find what you're looking for if you know its name.
- A collection of quick-access tiles give you instant access to important setup tools and feature information.
- The Most Recently Used list shows your most recently used records or customization features in Setup. You can quickly link back to what you were working on by clicking its name.
- The Object Manager provides a one-stop shop for managing all objects in your org, both standard and custom.

In Salesforce Classic:

- The left sidebar, which you can browse or search, provides access to all setup actions, tasks, and tools.
- The Getting Started box contains a tool for generating a basic app in a single step and links to information about extending and managing apps. This box doesn't appear if you've previously dismissed it.
- The Recent Items list shows recent metadata items that you've viewed, edited, or created and their related objects.
- The System Overview messages box displays messages to remind you when your org reaches its usage limits. The System Overview messages box isn't enabled by default.
- The Quick Links box provides links for managing tools, users, apps, security, and data.
- The Community box showcases available resources. If you've previously dismissed this box, it reappears with each new release.
- The right pane includes external links that are useful for developers and administrators.

Configure System Overview Messages

Add system overview usage messages to the Salesforce Home page to remind you when your org approaches its limits. You can expand, collapse, and dismiss the system overview messages that appear on the Home page. By default, the system overview home page messages are enabled.

Select Which Fields Appear in the Recently Viewed List

As a Salesforce admin, you can customize the Recently Viewed list that appears on the Setup home page for most standard and custom objects. Choose and order the fields to display so that your users see the information that's most important for your company.

Recent Items List (Beta)

The Recent Items list in Salesforce Classic shows recent metadata items that you've viewed, edited, or created and their related objects.

Quick Access Menu

The quick access menu offers handy shortcuts to customization features.

Configure System Overview Messages

Add system overview usage messages to the Salesforce Home page to remind you when your org approaches its limits. You can expand, collapse, and dismiss the system overview messages that appear on the Home page. By default, the system overview home page messages are enabled.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: All Editions except **Personal** and Database.com

USER PERMISSIONS NEEDED

To configure system messages: Customize Application

 **Note** The system overview page shows only the items enabled for your org. For example, your system overview page shows workflow rules only if workflow is enabled for your org.

1. From Setup, enter *System Overview* in the Quick Find box, then select **System Overview**.
2. Click **Configure Messages**.
3. Select or deselect the types of system overview messages to show or hide on the Home page.
4. Click **OK**.

 **Important** System overview messages only appear on the Salesforce Home page when your org approaches its limits.

When you enable or dismiss system overview messages, it only impacts your individual view of the messages.

Select Which Fields Appear in the Recently Viewed List

As a Salesforce admin, you can customize the Recently Viewed list that appears on the Setup home page for most standard and custom objects. Choose and order the fields to display so that your users see the

information that's most important for your company.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Essentials, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To customize recent records list

Customize Application

 **Note** These steps work in Lightning Experience. If you see the App Launcher icon (grid icon) on the left side of the navigation bar at the top of your screen, you're in Lightning Experience. If not, you're in Salesforce Classic.

For most list views, your users can select which fields to display and how to order the view columns. However, they can't edit the recent records quick list on object home pages. Only Salesforce admins can select and order the fields to display for the recent records quick list. Admins can't make any other changes to this default list.

1. From Setup, at the top of the page, select **Object Manager**.
2. Click the label name of the object for the Recently Viewed list you want to modify.
3. From the menu of links at the top of the page, click **Search Layouts**.
4. In the far right of the Search Results row, click  and select **Edit**.
Recently viewed lists use the Search Results search layout in Lightning. In Classic, recently viewed lists use the Tab search layout.
5. To add columns to the Recently Viewed list, select one or more fields from Available Fields and click **Add**. To remove columns, select one or more fields from Selected Fields and click **Remove**.
6. Order columns by selecting one or more fields from Selected Fields and clicking **Up** or **Down**.
7. Click **Save**.

 **Example** Your users collaborate on opportunities. To make it easy to see who worked on a recent opportunity last, select **Last Modified By** from the Available Fields list. Click **Add** to move it to Selected Fields. Now this information appears on the Recently Viewed list on the Opportunities home page.

In the Salesforce mobile app, the Recently Viewed list is the same list view as the desktop list view, with a few differences.

- The Recently Viewed list shows up to two fields for each record. To see more than two fields, tap .
- The fields shown come from the search results layout for the object. Any user with the Edit user permission for an object can add up to 10 fields. On desktop, go to **Setup | Customize | {Object Name} | Search Layouts | Search Results**. A user's changes also affect which fields are shown on the search results page.

-  **Note** In the Salesforce mobile app, a Recent list appears below the list views on an object's home page. It's an automatically generated list of recently accessed records. It isn't a list view and can't be modified.

Recent Items List (Beta)

The Recent Items list in Salesforce Classic shows recent metadata items that you've viewed, edited, or created and their related objects.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: all editions except **Database.com**

-  **Note** The Recent Items list is in beta. It's production quality but has known limitations.

The Recent Items list includes:

- Apex classes
- Apex triggers
- Approval processes
- Apps
- Custom report types
- Email templates
- Fields
- Lightning pages
- Objects
- Page layouts
- Permission sets
- Profiles
- Record types
- Static resources
- Tabs
- Users
- Validation rules
- Visualforce pages
- Visualforce components
- Workflow email alerts
- Workflow field updates
- Workflow outbound messages
- Workflow rules
- Workflow tasks

-  **Note** The Recent Items list in Salesforce Classic Setup is independent of the Recent Items section in the sidebar column of many Salesforce pages. The list in Setup shows items that administrators use, while the Recent Items section in the sidebar displays records with which end users have worked.

Quick Access Menu

The quick access menu offers handy shortcuts to customization features.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To view the quick access menu: Customize Application

When you're working on apps or objects, use this menu to jump to relevant app customization features. It's available from object list view pages and record detail pages.

-  **Note** If drag-and-drop scheduling on list views is enabled, the quick access menu isn't visible for list views on accounts, contacts, and custom objects.

- To expand or collapse the menu, click  (or press ALT+;).
- To scroll down the list of the menu, press TAB.
- To select an option on the menu, press ENTER.
- To remove the menu from all list views and record pages, click **Turn off menu**.

To restore the quick access menu:

- From your personal settings, enter *Advanced User Details* in the Quick Find box, then select **Advanced User Details**. No results? Enter *Personal Information* in the Quick Find box, then select **Personal Information**.
- Click **Edit**.
- Select the **Quick Access Menu** checkbox.
- Click **Save**.

See Also

[Personalize Your Salesforce Experience](#)

Customize Your Salesforce Org

You can customize each of the standard tabs and types of records, including adding custom fields and setting page layouts. You can also customize search, tagging, and user interface options for your org. In addition, every Contact Manager, Group, Professional, Enterprise, Unlimited, and Performance Edition user can customize various personal display options.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

The available customization options vary according to which Salesforce Edition you have.

USER PERMISSIONS NEEDED

To view setup options: View Setup and Configuration

To customize your org: Customize Application

Watch a Demo:  [Creating a Workflow Rule \(Salesforce Classic\)](#)

Quick demo of how to customize the way Salesforce looks for your organization.

To tailor Salesforce for your org, you can customize the display of the various tabs and other items. Select a link to get started on any task.

[Find Object Management Settings](#)

Salesforce lets you personalize your object model with features like custom fields, page layouts, and validation rules. Depending on which experience of Salesforce you have enabled, these customizations are located in different areas of Setup.

[Ways to Control User Access to Fields](#)

Use field-level security to control user access to fields. Use page layouts to control the layout and organization of detail and edit pages in Salesforce, the Self-Service Portal, and the Salesforce Customer Portal.

[Set Page Layouts and Field-Level Security](#)

Use field-level security as the means to restrict users' access to fields; then use page layouts primarily to organize detail and edit pages within tabs. This reduces the number of page layouts for you to maintain. For example, if a field is required in the page layout and read-only in the field-level security settings, the field-level security overrides the page layout and the field is read-only for the user.

[Page Layouts](#)

Page layouts control the layout and organization of buttons, fields, s-controls, Visualforce, custom links, and related lists on object record pages. They also help determine which fields are visible, read only, and required. Use page layouts to customize the content of record pages for your users.

[Compact Layouts](#)

A compact layout displays a record's key fields at a glance in the Salesforce mobile app, Lightning

Experience, and in the Outlook and Gmail integrations.

Custom Help Content

Tailor help so that users understand how to work within your unique implementation of Salesforce. You can add learning in the flow of work in several ways. Add custom help for a page, app, object, or org level, or provide links to help in the Utility Bar, Path, or as a text box on the page. Show users custom resources in a Lightning Experience welcome mat when they first log in. Or, to reach users with important news, training, and on-boarding information, add micro-learning prompts and walkthroughs to your app.

Tailor Business Processes to Different Record Types Users

Record types let you offer different business processes, picklist values, and page layouts to different users. You can create record types to differentiate your regular sales deals from your professional services engagements, offering different picklist values for each. Or you can display different page layouts for your customer support cases versus your billing cases.

Manage Your Translations

If your Salesforce org has multiple languages enabled, manage translations so that your global users can use Salesforce in their language.

Find Object Management Settings

Salesforce lets you personalize your object model with features like custom fields, page layouts, and validation rules. Depending on which experience of Salesforce you have enabled, these customizations are located in different areas of Setup.

Find Object Management Settings in Lightning Experience

Salesforce lets you customize your object model with features like custom fields, page layouts, and validation rules. Most objects are available from the Object Manager in Setup.

Find Object Management Settings in Salesforce Classic

Salesforce lets you personalize your object model with features like custom fields, page layouts, and validation rules. Depending on which type of object you want to find, these customizations are located in different areas of Setup.

Standard Object Limits

Standard object limits include usage details for object customizations, such as the custom fields you've added or sharing rules you've applied to an object.

App Setup Overview

The App Setup and Build pages list optional tasks to customize Salesforce, and build, customize, and manage applications.

Find Object Management Settings in Lightning Experience

Salesforce lets you customize your object model with features like custom fields, page layouts, and validation rules. Most objects are available from the Object Manager in Setup.

REQUIRED EDITIONS

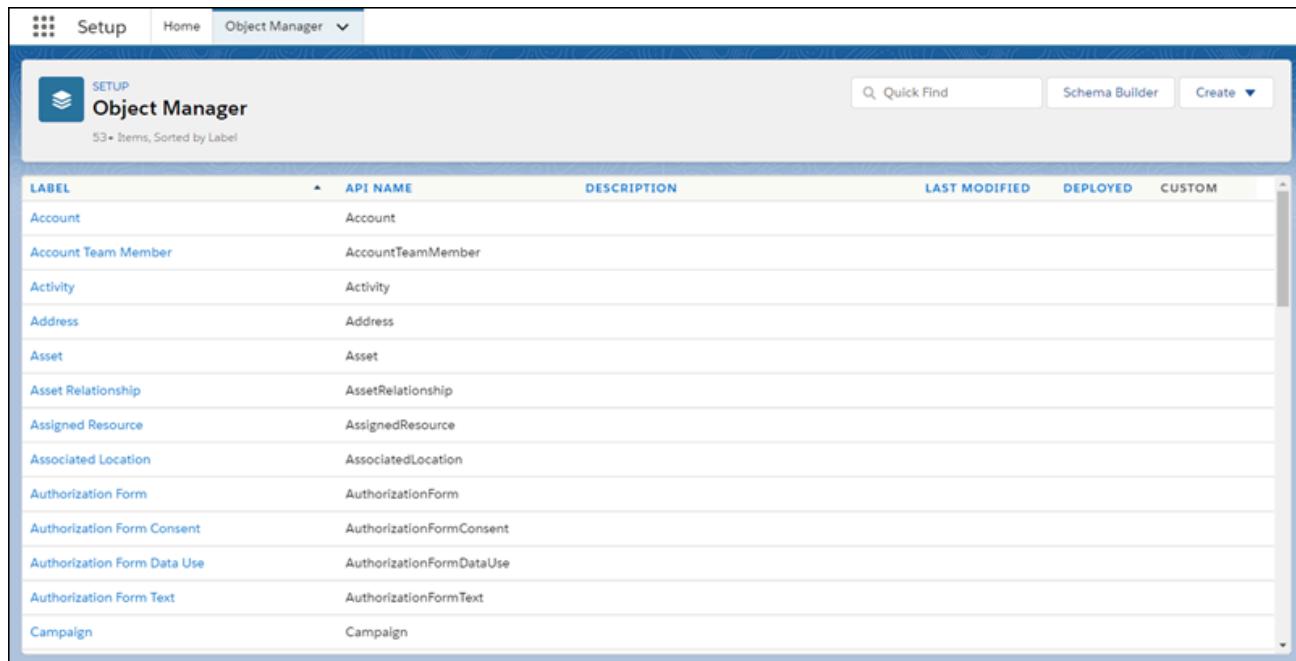
Available in: Lightning Experience

Available in: **All** editions

Standard Objects and Custom Objects

A *standard object*, such as Account or Contact, comes out of the box with your Salesforce organization. A *custom object* is an object that you or another administrator created.

From Setup, at the top of the page, select **Object Manager**. Select one of the objects in the list, and then select a specific customization from the left pane.



The screenshot shows the Salesforce Object Manager page. At the top, there's a navigation bar with 'Setup' and 'Object Manager'. Below it is a header with 'Object Manager' and search/filter buttons ('Quick Find', 'Schema Builder', 'Create'). The main area is a table listing objects:

LABEL	API NAME	DESCRIPTION	LAST MODIFIED	DEPLOYED	CUSTOM
Account	Account				
Account Team Member	AccountTeamMember				
Activity	Activity				
Address	Address				
Asset	Asset				
Asset Relationship	AssetRelationship				
Assigned Resource	AssignedResource				
Associated Location	AssociatedLocation				
Authorization Form	AuthorizationForm				
Authorization Form Consent	AuthorizationFormConsent				
Authorization Form Data Use	AuthorizationFormDataUse				
Authorization Form Text	AuthorizationFormText				
Campaign	Campaign				

For example, to add a custom field to the Account object, select **Object Manager** from the top of the Setup page. Next, select **Account**, and then **Fields & Relationships**. Select **New**.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Account Name	Name	Name		✓
Account Number	AccountNumber	Text(40)		
Account Owner	OwnerId	Lookup(User)		✓
Account Site	Site	Text(80)		
Account Source	AccountSource	Picklist		
Active	Active__c	Picklist		
Annual Revenue	AnnualRevenue	Currency(18, 0)		
Billing Address	BillingAddress	Address		
Clean Status	CleanStatus	Picklist		✓
Created By	CreatedById	Lookup(User)		
Customer Priority	CustomerPriority__c	Picklist		

- Note** If you've renamed any objects or fields, the Object Manager uses their assigned custom names.

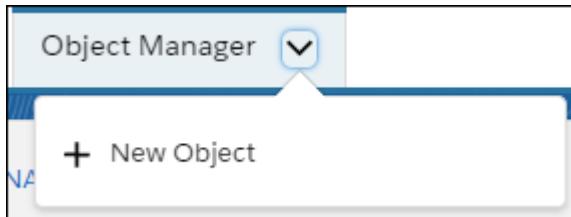
Other Standard Objects

Some standard objects aren't housed in the Object Manager. Standard objects with more specific purposes and customizations can be found in Setup using the Quick Find box.

From Setup, enter the object name in the Quick Find box, then select the customization.

Custom Objects

You can also create custom objects from the Object Manager. From Setup, at the top of the page, select **Object Manager**. Then select **New Object**.



External Objects

An *external object* is similar to custom objects, except that it maps to data that's stored outside your Salesforce organization.

From Setup, enter *External Objects* in the Quick Find box, then select **External Objects**. Next, click one of the external objects in the list. Then scroll to the section for the specific customization.

For example, to add a custom field to the Orders external object, enter *External Objects* in the Quick Find box, then select **External Objects**. Click **Orders**, and then scroll to Custom Fields and Relationships.

See Also

[Gaps Between Salesforce Classic and Lightning Experience](#)

Find Object Management Settings in Salesforce Classic

Salesforce lets you personalize your object model with features like custom fields, page layouts, and validation rules. Depending on which type of object you want to find, these customizations are located in different areas of Setup.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **All** editions

Standard Objects

A *standard object*, such as Account or Contact, comes out of the box with your Salesforce organization.

From Setup, enter the name of the appropriate object in the **Quick Find** box, then select the specific

customization.

For example, to add a custom field to the Case object, enter *Case* in the **Quick Find** box, then select **Fields** under Cases.

Custom Objects

A *custom object* is an object that you or another administrator created.

From Setup, enter *Objects* in the **Quick Find** box and select **Objects**. Next, click one of the custom objects in the list. Then scroll to the section for the specific customization.

For example, to add a custom field to the Job Applications object, enter *Objects* in the **Quick Find** box, then select **Objects**. Click **Job Applications**, and then scroll to Custom Fields and Relationships.

External Objects

An *external object* is similar to custom objects, except that it maps to data that's stored outside your Salesforce organization.

From Setup, enter *External Objects* in the **Quick Find** box, then select **External Objects**. Next, click one of the external objects in the list. Then scroll to the section for the specific customization.

For example, to add a custom field to the Orders external object, enter *External Objects* in the Quick Find box, then select **External Objects**. Click **Orders**, and then scroll to Custom Fields and Relationships.

Standard Object Limits

Standard object limits include usage details for object customizations, such as the custom fields you've added or sharing rules you've applied to an object.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All Editions except Database.com**

The list varies depending on the object. When a customization exceeds the allowed limit for the object, or reaches 75% of the limit, a tip displays that suggests what you can do next.

Refer to the standard object limits page when you're planning to customize a particular standard object or to monitor the usage and limits of customizations on that object.

To see an object's usage:

- If you're using Lightning Experience, from Setup, open the Object Manager, click the name of the

- object you want to see the usage for, and then go to Object Limits.
- If you're using Salesforce Classic, from Setup, enter the object name in the Quick Find box, and then select Limits under that object.

 **Note** The object limit percentages are truncated, not rounded. For example, if your org uses 95.55% of the limit for a particular customization, the object limit displays 95%.

See Also

[Knowledge Article: Increase the maximum relationships \(master-detail plus lookup\) allowed per object](#)

App Setup Overview

The App Setup and Build pages list optional tasks to customize Salesforce, and build, customize, and manage applications.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: All Editions.

Click any task in the list to get started.

Rename Object, Tab, and Field Labels

You can change the name of almost any object, field, or tab in Salesforce. This simple adjustment lets you continue using the terminology your users already know and helps them transition to using Salesforce. However, Salesforce Help and most pages in Setup always display the original names for standard objects, fields, and tabs.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To rename a tab and field:

Customize Application

OR

Manage Translation

USER PERMISSIONS NEEDED

OR

If you're designated as a translator, you need:

View Setup and Configuration

To reset renamed tabs:

Customize Application

OR

Manage Translation

OR

If you're designated as a translator, you need:

View Setup and Configuration

Before renaming tabs, objects, fields, and other related labels, review the [implementation tips](#) for administrators.

For example, you can change the name label for the “Accounts” object and related “Accounts” tab to “Companies”, and change the field “Account Name” to “Company Name.” When you rename an object, tab, or field, the new name appears on all pages the user sees, in Salesforce for Outlook, and in Connect Offline.

If you use person accounts in your Salesforce org, see [Renaming Person Account Labels](#).

1. From Setup, enter *Rename Tabs and Labels* in the Quick Find box, then select **Rename Tabs and Labels**.
2. Select your default language from the Select Language dropdown list at the top of the page.

 **Note** In Hebrew, we recommend keeping tab renaming to a minimum because variable gender in verbs isn't supported and verbs can lose gender agreement.

3. Click **Edit** next to the tab you want to rename. Click **Reset** to revert to a tab's original name.

 **Note** You can't reset custom object tab names.

4. Enter the singular and plural forms of the new tab name. Also, if applicable for the language, select **Starts with a vowel sound** for labels that start with a vowel to ensure that Salesforce uses the proper article (such as “a” or “an”). Then click **Next**.

When you rename a tab or an object, you can't use the name of another standard tab, custom object, external object, or custom tab. When you rename an object, a field, or a tab, you can't use these characters: #, \$, %, ;, <, =, >, [,], ^, ` , |, and ~.

5. Enter your labels for the standard field labels and other user interface elements. Be sure to enter both a singular and plural form for each label that requires it. Select **Starts with a vowel sound** for labels

that start with a vowel.



Note Some standard fields, such as Created By and Last Modified By, are purposely omitted from renaming because they track system information.

6. Click **Save**.

Repeat this procedure to translate labels into all other languages used in your organization.



Tip After renaming a tab or object, rename any custom reports, dashboards, profiles, permission sets, custom fields, and list views that contain the original name. You can modify labels using the Translation Workbench. To rename a standard report, click **Save As** and save it to a folder designed for your new name.

Other tab customization options include:

- Individual users can control which tabs and related lists display for their own logins.
- In addition to the standard tabs provided by Salesforce, users can create entirely new custom tabs depending on their Edition.
- In Enterprise, Unlimited, Performance, and Developer Edition organizations, you can override the tab home page that is displayed when a user clicks a standard, custom, or external object tab. For more information, see [Override Standard Buttons and Tab Home Pages](#).

See Also

[Considerations for Renaming Tab and Field Labels](#)

[Rename the Chatter Tab](#)

[Find Object Management Settings](#)

Renaming Person Account Labels

If your org uses person accounts, you can rename three standard fields.

REQUIRED EDITIONS

Available in: Salesforce Classic

Person accounts available in: **Enterprise, Performance, Unlimited, and Developer** Editions

Field	Tab	Description
Business Account	Accounts	An account that isn't a person account because it doesn't have a record type specific to person accounts. This label is primarily used to clarify the type of accounts you're importing.
Person Account	Accounts	A person account.
Business Contact	Contacts	A contact that is associated with a business

Field	Tab	Description
		account. This label is primarily used to clarify the type of accounts you're importing.

When you rename the Person Account field label, the renamed label appears in Salesforce:

- As a prefix to differentiate person account fields such as Birthdate and Home Phone from business account fields. For example, Person Account: Birthdate is available as an account column in opportunity reports.
- In the name of the Is Person Account field and icon. For example, if you rename the Person Account field to “Consumer,” then Is Person Account becomes Is Consumer.

 **Note** The Person Account and Business Account field labels are independent from actual record type names.

- To customize person account record types, from the object management settings for person accounts, go to Record Types.
- To customize business account record types, from the object management settings for accounts, go to Record Types.

See Also

- [Rename Object, Tab, and Field Labels](#)
[Considerations for Renaming Tab and Field Labels](#)

Considerations for Renaming Tab and Field Labels

Before renaming standard and custom tabs and fields, learn how those changes affect your users.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To rename tab and field labels:	Customize Application
To reset renamed tabs:	Customize Application

Most standard tabs and objects can be renamed but not all. For example, the Forecasts tab isn't available for renaming. From Setup, enter *Rename Tabs and Labels* in the Quick Find box, then select **Rename Tabs and Labels** to view a list of the tabs and objects you can rename.

- The renamed labels appear on all user pages in Salesforce, including Personal Setup. In Lightning Experience, all Lightning-based pages in the Setup area use the renamed labels. However, Salesforce

Classic Setup pages, whether they're in Salesforce Classic or embedded in a Lightning Experience Setup page, use the default, original labels.

- Some standard fields, such as Created By and Last Modified By, are purposely omitted from renaming because they track system information.
- When you rename a tab or an object, you can't use the name of another standard tab, custom object, external object, or custom tab.
- When you rename a tab, a field, or an object, you can't use include these characters: #, \$, %, ;, <, =, >, [,], ^, ` , |, and ~ .
- After renaming tabs, objects, or fields, check the following additional items to determine which items need manual updates:
 - Review all list view names. Built-in list views where you haven't previously modified the list view name, such as All Accounts, automatically update to use the new name. Built-in list views where you've previously modified the list view name and user-created list views continue to show the original object name until you change them manually.
 - Check standard report names and descriptions for the objects you renamed.
 - Update the titles and descriptions of any email templates that contain the original object or field name.
 - Manually change any other items you customized with the new object or field name. For example, change any custom fields, page layouts, and record types that contain the original tab or field name.
- Connect Offline, the Outlook integration, the Gmail integration, and Salesforce for Outlook use your new names.
- When you rename custom tabs for custom objects, the tab label normally appears in the requested language. If there's no translation for the requested language, the tab label appears in the default language of the custom object. The custom object default language is the default language of the org when the object was created.
- If you have renamed tabs, objects, or fields, you can also replace Salesforce Help with another URL. Users can view this URL whenever they click any context-sensitive help link on an end-user page or within their personal settings. After you replace the help, the Help & Training link at the top of every page and all Setup pages will continue to display Salesforce Help.
- To rename custom object labels without changing all of the custom object's original labels, edit the labels from **Setup | Rename Tabs and Labels**.
- In Hebrew, we recommend keeping tab renaming to a minimum because variable gender in verbs isn't supported and verbs can lose gender agreement.

See Also

[Rename Object, Tab, and Field Labels](#)

Ways to Control User Access to Fields

Use field-level security to control user access to fields. Use page layouts to control the layout and organization of detail and edit pages in Salesforce, the Self-Service Portal, and the Salesforce Customer Portal.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Page layouts and search layouts available in: **All Editions**

Field-level security available in: **Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

! **Important** When you use page layouts to hide fields from detail and edit pages, users can still see these fields via reports, search results, list views, and the API. To restrict field access, use field-level security. Search doesn't return results for records with fields protected by field level security. In some rare situations, when search terms match field values protected by field-level security, the associated records are returned but without the protected fields and their values. Don't use page layouts to secure data. For example, removing the Edit button from a page layout doesn't prevent users from using inline editing. To prevent users from editing data, use sharing rules, field-level security, page layout field properties, validation rules, object permissions, and Visualforce pages.

Field-Level Security

- Restrict users' access to view and edit fields. For example, restrict access in reports, search results, list views, related lists, email, and mail merge templates, custom links, Connect Offline. Also restrict API access and when synchronizing data or importing personal data.
- Override less-restrictive field access settings in page layouts and mini page layouts. For example, if a page layout requires a field that's read-only in field-level security settings, the field remains read-only for the user.
- Override less-restrictive field settings in search layouts. For example, if a field is visible in the search layout but hidden via field-level security settings, the field remains hidden.

Page Layouts

- Control the layout and organization of detail and edit pages.
- Control which fields, related lists, and custom links users see, on detail and edit pages only.
- Control which standard and custom buttons display on detail pages and related lists.
- Determine whether fields are visible, read only, or required, on detail and edit pages only.
- Determine the fields that users can import data into.
- In Personal, Contact Manager, Group, and Professional Editions, page layouts control which fields users can access in:
 - related lists and list views
 - reports
 - Connect Offline
 - email and mail merge templates
 - custom links

Page layouts also control field access when synchronizing data.

- In Professional, Enterprise, Unlimited, Performance, and Developer Editions, determine aspects of

mini page layouts, including:

- record type
- profile associations
- related lists
- fields and field access settings

The visible fields and related lists of the mini page layout can be further customized. But other items inherited from the associated page layout can't be changed on the mini page layout. Mini page layouts display selected fields and related lists of records in the mini view of the console.

 **Tip** To automatically add a field to all page layouts and make it visible and required everywhere regardless of field-level security, make it a universally required field.

See Also

[Page Layouts](#)

[Customize Search Layouts to Show Results Users Want](#)

Set Page Layouts and Field-Level Security

Use field-level security as the means to restrict users' access to fields; then use page layouts primarily to organize detail and edit pages within tabs. This reduces the number of page layouts for you to maintain. For example, if a field is required in the page layout and read-only in the field-level security settings, the field-level security overrides the page layout and the field is read-only for the user.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Page layouts and search layouts available in: **All Editions**

Field-level security available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To view field accessibility:

[View Setup and Configuration](#)

1. Create custom fields.
2. Create any custom buttons or links.
3. If you're in a Personal, Contact Manager, or Group Edition org, skip to step 8.
4. Create any custom profiles.
5. Create record types for different business scenarios.
6. Assign which record types are available to users with different profiles.
7. Set the field-level security for each profile to restrict users' access to specific fields.
8. Define page layouts to organize your pages.
9. Set the related objects and the mini page layouts that display in the console.

10. Assign page layouts to users based on profiles and record types.
This step isn't applicable for Personal, Contact Manager, and Group Editions. In Personal, Contact Manager, and Group Edition orgs, all users automatically use the same page layout for each object.
11. To verify that all field access settings are correct, check the field accessibility grid.
This step isn't applicable for Personal, Contact Manager, and Group Editions.
12. Define search layouts. All users use the same search layouts.

 **Tip** Click **Preview** while editing a page layout to see how the page looks for users with different profiles. This preview includes any extra security that is set in field-level security.

See Also

[Field-Level Security](#)

Page Layouts

Page layouts control the layout and organization of buttons, fields, s-controls, Visualforce, custom links, and related lists on object record pages. They also help determine which fields are visible, read only, and required. Use page layouts to customize the content of record pages for your users.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Page layouts available in: all editions

Creation and deletion of page layouts available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Page layouts can include s-controls and Visualforce pages that are rendered within a field section when the page displays. You can control the size of the s-controls and Visualforce pages, and determine whether a label and scroll bars display.

Salesforce has two drag-and-drop tools for editing page layouts: the original page layout editor and an enhanced page layout editor. The enhanced page layout editor is enabled by default, and provides all the functionality of the original editor, as well as additional functionality and an easier-to-use interface.

You can enable the original page layout editor in the User Interface settings. Your Salesforce org can use only one page layout editor at a time.

From within a page layout, you can access a mini page layout. The mini page layout defines the hover details that display when you mouse over a field on an object's detail page in the Agent console or in the Recent Items section of the sidebar in Salesforce Classic.

Salesforce automatically creates a default page layout when you create a custom object. If you don't use any page layout with your custom object, you can still interact with it by using the Lightning Platform API to manage custom data or build a custom user interface.

Create Page Layouts

With the enhanced page layout editor, you can tailor record page layouts to the needs of your users.

The Enhanced Page Layout Editor

The enhanced page layout editor is a tool for customizing your page layouts in Salesforce, the Self-Service Portal, and the Salesforce Customer Portal. The enhanced page layout editor has all the functionality of the original page layout editor, but has more features and an easier-to-use interface.

Assign Page Layouts to Profiles or Record Types

After defining page layouts, assign which page layouts users see. A user's profile determines which page layout he or she sees. In addition, if your organization is using record types for a particular object, the combination of the user's profile and the record type determine which page layout is displayed when a user views records for that object.

Edit Multi-Line Layouts for Opportunity Products

Customize the columns that display when users add or edit items in the Products related list of an opportunity detail page.

Configure Fields on Multi-Line Layouts for Opportunity Products

Before you can add a field to the Opportunity Product multi-line layout, the field must be visible on the Opportunity Product page layout. You make the field visible via the Product related list on an Opportunity object page layout.

Customize Related Lists

You can customize the buttons, columns displayed, column order, and record sort order of related lists on record detail pages in Salesforce and the Salesforce Customer Portal.

Customize Standard and Custom Buttons on Page Layouts

When customizing page layouts, you can control which standard and custom buttons are displayed and the order in which the custom buttons appear.

How Page Layout Elements Display in Lightning Experience

When you change your page layouts in Salesforce Classic, it can impact the content shown on object record pages in Lightning Experience. However, in Lightning Experience, the page elements look different, and some features are not supported.

Page Layout Tips

Here are a few tips to keep your page layouts organized and easy to use.

Page Layout Considerations

Keep these considerations in mind when working with page layouts in the enhanced page layout editor.

Page Layout Limitations

Keep these limitations in mind when working with page layouts in the enhanced page layout editor.

How Page Layouts Work in the Salesforce Mobile App

Use the enhanced page layout editor to customize the layout of an object's record detail pages, configure actions, and adjust which fields and related lists appear in the Salesforce mobile app.

Manage Mobile Cards in the Enhanced Page Layout Editor

Add expanded lookups, components, and Visualforce pages to the Mobile Cards section of your page layout to have them show up as mobile cards in the Salesforce mobile app.

Feed-Based Layouts in Salesforce Classic

Feed-based page layouts make it easier to work with records by providing two separate views: one for the record's feed, and one for its details, including related lists.

Salesforce Classic Home Tab Page Layouts

You can customize the Home tab in Salesforce Classic to include components such as sidebar links, a company logo, a dashboard snapshot, or custom components that you create. A dashboard snapshot is a clipping of the top row of a dashboard's components. Just like other tabs, you can also assign different home page layouts to different users based on profile.

Customize Page Layouts with the Original Page Layout Editor

Use the original page layout editor to customize page layouts in Salesforce, the Self-Service Portal, and the Salesforce Customer Portal.

Considerations for Using the Original Page Layout Editor

When working with the Original Page Layout Editor in Salesforce Classic, keep these considerations in mind.

See Also

Ways to Control User Access to Fields

Customize Search Layouts

Trailhead: Customize Record Details with Page Layouts

Create Page Layouts

With the enhanced page layout editor, you can tailor record page layouts to the needs of your users.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: Enterprise, Performance, Unlimited, and Developer Editions

USER PERMISSIONS NEEDED

To create page layouts:

Customize Application

1. From the management settings for the object that you want to edit, go to **Page Layouts**.
 2. Create a page layout in one of these ways.
 - Click **New** from the Page Layouts list page.
 - Clone an existing layout by clicking **New** from the Page layouts list page, then selecting a layout from the Existing Page Layout menu that you want to base the new layout on.
 - Clone an existing layout by using Save As inside the enhanced page layout editor.
 3. Give the layout a name.
 4. Click **Save**.

See Also

Customize Page Layouts with the Enhanced Page Layout Editor

Assign Page Layouts to Profiles or Record Types

The Enhanced Page Layout Editor

The enhanced page layout editor is a tool for customizing your page layouts in Salesforce, the Self-Service Portal, and the Salesforce Customer Portal. The enhanced page layout editor has all the functionality of the original page layout editor, but has more features and an easier-to-use interface.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

The enhanced page layout editor has two parts: a palette on the upper portion of the screen and the page layout on the lower portion of the screen. The palette contains the user interface elements that you can add to your page layout, such as fields, actions, buttons, links, and related lists.



With the enhanced page layout editor you can:

- Add custom console components, customize the Mini-Page Layout for record pages in Salesforce Classic, configure Multi-Line layouts (on supported objects), and select which objects appear in the mini view of the console.
- Customize the highlights panel on Salesforce Classic pages (2). This section isn't used in Lightning Experience.
- Manage the quick actions that appear in the Salesforce Classic publisher (3) and in Lightning Experience and the Salesforce mobile app (4). See [Quick Actions](#).
- Add, remove, and reorder:
 - Custom and standard buttons on the record page (5).
 - Fields in the record details (6).
 - Custom links.
 - Report charts.
- Customize the related lists that appear on record pages, including configuring related list buttons and selecting which fields appear on a related list. See [Customize Related Lists](#).

Note You can only view object page layouts in Lightning Experience using the enhanced page layout editor. Accessing an object page layout in Lightning Experience with the original page layout editor can cause an "Insufficient Privileges" error. To confirm that the enhanced page layout editor is enabled, in Setup, go to **User Interface | User Interface**, then ensure that **Enable Enhanced Page Layout Editor** is selected.

[Customize Page Layouts with the Enhanced Page Layout Editor](#)

Tailor your page layouts to the needs of your users with the enhanced page layout editor. Add, remove, or reorder actions, buttons, fields, and sections on a record's detail page.

[User Interface Elements for the Enhanced Page Layout Editor](#)

This list describes the enhanced page layout editor user interface elements and how you can use them in your page layout. To add elements, drag them from the palette to the layout. Valid drop locations show up in green. To remove elements, drag them off the layout and back to the palette.

Standard Object Record Page Save Options on Page Layouts

With save options, you can make object-specific checkboxes appear on the edit pages for certain standard objects. Save options trigger additional processes when a record is saved. You can also set the checkboxes to be selected by default.

Tips for Using the Enhanced Page Layout Editor

Keep these tips in mind when customizing page layouts in the enhanced page layout editor.

See Also

[Page Layouts](#)

Customize Page Layouts with the Enhanced Page Layout Editor

Tailor your page layouts to the needs of your users with the enhanced page layout editor. Add, remove, or reorder actions, buttons, fields, and sections on a record's detail page.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create page layouts: Customize Application

1. From the management settings for the object that you want to edit, go to [Page Layouts](#).
2. Click ▾ on a layout's row, and select **Edit**, or simply click the name of the layout.
3. Add items to the layout by clicking the element type in the palette and dragging an item in that category onto the layout.

4. Remove an item by dragging it from the layout back to the palette, or hover over it and click the remove icon (⊖).
5. Make a field ready-only or required by double-clicking it in the page layout and selecting the appropriate checkbox.
6. Save the layout.

See Also

[User Interface Elements for the Enhanced Page Layout Editor](#)

[Page Layout Tips](#)

User Interface Elements for the Enhanced Page Layout Editor

This list describes the enhanced page layout editor user interface elements and how you can use them in your page layout. To add elements, drag them from the palette to the layout. Valid drop locations show up in green. To remove elements, drag them off the layout and back to the palette.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

 **Important** The Mobile Cards section of the page layout editor is no longer supported as of Spring '20. Items in the Mobile Cards section, such as components and expanded lookups, no longer display in the Salesforce mobile app. For more information, see [Mobile Cards Are Not Available in the New Salesforce Mobile App](#) in the Salesforce Spring '20 release notes.

Analytics Assets

You can add and move an Analytics dashboard to any section on the page layout, except Mobile Cards.

For an Analytics dashboard element, use field mapping to map data fields in the dashboard to the object's fields so that the dashboard shows only the data that's relevant for the record being viewed. For more about field mapping or filtering Analytics dashboards, see [Embed Analytics Dashboards in Lightning Pages](#) or [Embed Analytics Dashboards in Salesforce Classic Pages](#).

Blank Spaces

You can find blank spaces in the Fields and the Custom Links categories of the page layout editor palette.

You can add and move blank spaces to any section on the page layout, except Mobile Cards. Use blank spaces to visually align and distinguish elements on the page.

 **Note** If you use the original page layout editor to view a page layout that was created in the enhanced page layout editor, the original page layout editor shows the blank spaces that you added. You can't move or add blank spaces in the original page layout editor, but you can remove them by dragging them to the box on the right.

Buttons

You can control which standard and custom buttons are displayed and the order in which the custom buttons appear. You can't rearrange standard buttons.

Standard and custom buttons are available as actions in the Salesforce mobile app and Lightning Experience.

Canvas Apps

For the Canvas Apps category to appear in the palette, set the canvas app location to Visualforce Page when you create the canvas app in Salesforce.

If you add a canvas app to any section other than the Mobile Cards section, the canvas app appears in the page layout in the full Salesforce site.

Canvas apps added as mobile cards don't appear in the Salesforce mobile app. To have a canvas app appear on a mobile record page, add it to the page as a component using the Lightning App Builder.

Components and Expanded Lookups

Components and expanded lookups added as mobile cards don't appear in the Salesforce mobile app. Expanded lookups are no longer supported in mobile as of Spring '20.

To have a News or Twitter component appear on a mobile record page, add it to a Lightning page using the Lightning App Builder.

Fields

A field can display one or more of these icons:

- —The field must have a value to save the record, but isn't required on the page layout itself.
- —The field must be included on the page layout because an administrator configured the field as universally required or Salesforce automatically requires the field. Although you can't remove such fields, you can move them to different locations.
- —The field is a controlling field.
- —The field is a dependent field.
- —The field is read-only.

To set which fields are required and read-only, select one or more fields and click on any selected field.

- You can't change the field properties of some standard fields. You can change custom fields only if they aren't universally required fields.
- Administrators and users with the Edit Read Only Fields permission can always edit fields marked as read-only.
- If you make a picklist field read-only, all new records contain the default value for that picklist.
- Auto-number fields are always read-only.
- If you mark the opportunity Probability field as read-only, the Probability value is still updated when a user changes the Stage value of an opportunity.

When working with fields:

- In Personal, Contact Manager, and Group Editions, page layouts control which fields users can access

- in related lists, list views, reports, Connect Offline, email and mail merge templates, custom links, and when synchronizing data. In Professional, Enterprise, Unlimited, Performance, and Developer Editions, field-level security controls this access. Field-level security settings override field properties that you set on the page layout if the field-level security is more restrictive than the page layout setting.
- Users can import values into a field only if they have read and edit access. User permissions, page layout assignments, and field-level security settings determine field access.

Mobile & Lightning Actions

The items in this category of the page layout editor palette are a mixture of different types of actions, such as global quick actions, object-specific actions, custom actions, and Chatter actions like Post and Poll.

All of the items in this category are supported for Lightning Experience. Add items from this category to the Salesforce Mobile and Lightning Experience Actions section.

Actions in Lightning Experience display in different places on a record page, such as the highlights panel, Activity tab, and the Chatter tab. Where they appear depends on their type.

In the Salesforce mobile app, actions of all types appear in the action bar, the action bar's action menu, and as list-item actions.

Quick Actions

This category of the page layout editor palette contains standard and custom quick actions that are supported for Salesforce Classic. In Salesforce Classic, quick actions appear in the Chatter publisher when Chatter Settings are enabled. Some actions work in Salesforce Classic and Lightning Experience, so they appear in both categories.

When customizing a page layout for Salesforce Classic record pages, drag actions from here into the Quick Actions in the Salesforce Classic Publisher section.

Mobile smart actions appear as a single action element in the page layout editor. In the Salesforce mobile app, the Mobile Smart Actions element expands to distinct create actions that enable users to create records directly from the action bar. The create actions included in the set of mobile smart actions vary depending on the page layout's object.

 **Note** If you delete an action, the action is removed from all layouts that it's assigned to.

Related Lists

A page layout can have up to 100 related lists. You can place related lists at the bottom of the page layout. To move a related list on the page layout, drag the handle located above the related list.

To customize a related list, double-click the related list handle or click  inside the handle. Use the

related list properties to:

- Specify which fields display as columns on the related list, the order in which they appear, and the sort order of the records in the related list. In Professional, Enterprise, Unlimited, and Performance Editions, you can also opt to apply the column information to other page layouts for the same type of object.
- Specify which standard and custom buttons appear on the related list.

When working with related lists on page layouts, note that:

- Some related lists aren't customizable because they link to data rather than store it. Related lists that aren't customizable are indicated on the page layout.
- You can't add related lists to the page layouts for the User object.
- In Professional, Enterprise, Unlimited, Performance, and Developer Editions, individual users can customize which related lists display for their personal use. Administrators can overwrite these user customizations and apply the related list configuration in the page layout to all users, even if they already customized their display. To overwrite users' related list customizations, click **Yes** on the Overwrite Users' Customized Related Lists window, which appears when saving a page layout if you moved or added a related list.

Report Charts

Report charts are supported in Salesforce Classic and Lightning Experience.

S-Controls

A page layout can have up to 20 s-controls.

To change the properties of an s-control, double-click the s-control or click its wrench icon () , and then set these attributes.

- **Width** sets the horizontal size in pixels or a percent.
- **Height** sets the vertical size in pixels.
- **Show scrollbars** determines whether the iFrame in which the s-control displays contains scroll bars when necessary.
- **Show label** determines whether the page layout includes the Label of the custom s-control. Remove the label to display the s-control in a wider area.

S-controls aren't supported in Lightning Experience.

Sections

You can add and move sections anywhere above the related lists on the page layout. The sections you add can contain fields, s-controls, and blank spaces. In addition, each page layout has a default section that can only contain custom links and blank spaces. You can change the location of the custom link section, but you can't remove it from the page.

The Section user interface element is the second option in the palette when you select the Fields or Custom S-Controls category on the palette.

To change the attributes of a section, double-click the section or select its associated wrench icon (). You can:

- Enter a name for the section. Names of some standard page sections can't be changed.
- Specify whether the section has one or two columns.
- Specify the order in which users can tab through the items in that section.
- Specify whether the section heading is shown on the detail and edit pages.

Tags

If tags are enabled, click **Layout Properties**, and configure personal and public tags in the header section of the page layout. Users can't tag a record if personal or public tags aren't included in the header section. Also, the positioning of personal and public tags in the header can't be modified.

Tags aren't supported in Lightning Experience.

Visualforce Pages

 **Important** Due to third-party cookie restrictions in modern web browsers, Visualforce pages embedded in page layouts sometimes don't render properly in Salesforce Classic. See [Embedded Visualforce page not displayed in record page layout in Classic UI](#).

You can add Visualforce pages to any section on the page layout, except for sections reserved for custom links and related lists. A page layout can have up to 20 Visualforce pages.

You can add a Visualforce page to a page layout only if the standard controller on the Visualforce page is set to the object for which you're creating the page layout. If you don't have any Visualforce pages with a standard controller set to that object, the Visualforce Pages category doesn't appear in the palette.

Visualforce pages added as mobile cards in the page layout editor don't appear in the Salesforce mobile app. To have a Visualforce page appear on a mobile record page, add it to the page as a Visualforce component using the Lightning App Builder.

See Also

- [Page Layout Considerations](#)
- [Page Layout Limitations](#)
- [The Enhanced Page Layout Editor](#)

Standard Object Record Page Save Options on Page Layouts

With save options, you can make object-specific checkboxes appear on the edit pages for certain standard objects. Save options trigger additional processes when a record is saved. You can also set the

checkboxes to be selected by default.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **All** Editions

To configure how Salesforce displays the checkboxes, when customizing the page layout in the enhanced page layout editor, click **Layout Properties**.

Use the **Select by default** option associated with a checkbox if you want Salesforce to automatically select the option when a user accesses the edit window.

If both **Show on edit page** and **Select by default** are selected, the save option checkbox is selected by default, but users can deselect it to disable the option.

Object	Available Save Options
Account	<p>“Evaluate this account against territory rules on save” checkbox –Displays the Evaluate this account against territory rules on save checkbox on account edit pages. Available only if Enterprise Territory Management is enabled for your org.</p> <p>Territory assignment rules run automatically when the Default checkbox is selected.</p>
Case	<ul style="list-style-type: none"> Case Assignment Checkbox– Displays the Assign using active assignment rules checkbox on case edit pages. Case assignment rules run automatically when the Default checkbox is selected. Email Notification Checkbox – Displays the Send notification email to contact checkbox on case edit pages.
Case Close	<ul style="list-style-type: none"> Solution information section – Displays the solution information section on the case close edit pages. Notify contact checkbox – Displays the Notify Contact checkbox on case close edit pages.
Lead	<p>Lead Assignment Checkbox– Displays the Assign using active assignment rule checkbox on the lead edit page.</p> <p>Lead assignment rules run automatically when the Default checkbox</p>

	is selected.
Person Account	Evaluate this account against territory rules on save – Displays the Evaluate this account against territory rules on save checkbox on person account edit pages.
	Territory assignment rules run automatically when the Default checkbox is selected.
Task	Email notification – Displays the Send Notification Email checkbox on the task edit page. A user's personal preference for defaulting the state of the checkbox takes precedence over the org-wide setting.

 **Example** The save options from page layouts appear in the footer of the edit window in Lightning Experience. In this case, the “Evaluate this account against territory rules on save” checkbox option was enabled on the account page layout. When users edit an account record, they now see the checkbox in the edit window.



See Also

[Customize Page Layouts with the Enhanced Page Layout Editor](#)

[The Enhanced Page Layout Editor](#)

[Page Layouts](#)

Tips for Using the Enhanced Page Layout Editor

Keep these tips in mind when customizing page layouts in the enhanced page layout editor.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **All** Editions

- To select multiple elements individually, use Ctrl+click. To select multiple elements as a group, use Shift+click.
- To change the properties of an element on the page layout, double-click the element or click the wrench icon () next to it. You can't change the properties of elements in the palette.
- To make a field read-only or required, double-click the field in the page layout and select the appropriate checkbox.
- To access the other layouts for an object with multiple page layouts, click the page layout name at the top of the page and select another layout to view.
- To change the name of the page layout, add personal and public tags if available, and display standard object checkboxes, click **Layout Properties**.



Note You can't rename a page layout if you're using Salesforce Professional Edition.

- In Enterprise, Unlimited, Performance, and Developer Editions, you can select a profile to preview how the pages look for users with that profile. Most related lists' columns preview without data.
- If you're working with a feed-based page layout, click **Feed View** to customize the tools and components that appear when users are working in the feed on a record.
- To choose which fields display on the record detail page and the order in which they appear, click **Edit Multi-Line Layout**.
- Multi-line layout is available only for page layouts on certain objects, including Opportunity Product, Opportunity Split, Order Product, Contract Line Item, and Sales Agreement Product. If you don't see the **Edit Multi-Line Layout** link in the page layout editor on an object layout, then multi-line layout isn't supported for that object.
- The mini page layout defines the hover details that display when you mouse over a field on an object's detail page, in the Agent console, or in the Recent Items section of the sidebar in Salesforce Classic.
- When you're done customizing the page layout, save it. If you navigate away from your page layout before saving, you lose your changes.

See Also

[Page Layout Considerations](#)

[Page Layout Limitations](#)

[Tips for Optimizing Page Layouts for the Salesforce Mobile App](#)

[The Enhanced Page Layout Editor](#)

Assign Page Layouts to Profiles or Record Types

After defining page layouts, assign which page layouts users see. A user's profile determines which page layout he or she sees. In addition, if your organization is using record types for a particular object, the combination of the user's profile and the record type determine which page layout is displayed when a user views records for that object.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Page layouts are available in: **All** Editions

Record types are available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To assign page layouts:

Manage Profiles and Permission Sets

You can assign page layouts from:

- The object's customize page layout or record type page
- The original or enhanced profile user interface.

1. From the management settings for the appropriate object, go to Page Layouts or Record Types.

2. Click **Page Layout Assignment**.

3. Click **Edit Assignment**.

4. Use the table to specify the page layout for each profile.

The table displays the page layout assignments for each profile. If your organization uses record types, a matrix displays a page layout selector for each profile and record type.

When selecting page layout assignments:

- Click a cell, column, or row heading to select all the table cells in that column or row.
- Press SHIFT+click to select multiple adjacent table cells, columns, or rows.
- Press CTRL+click to select multiple nonadjacent table cells, columns, or rows.
- Click any cell and drag to select a range of cells.
- Click **Next** or **Prev** to view another set of record types.

Selected page layout assignments are highlighted. Page layout assignments you change are italicized until you save your changes.

5. If necessary, select another page layout to assign from the **Page Layout To Use** drop-down list and repeat the previous step for the new page layout.

6. Click **Save**.



Note To verify that users have the correct access to fields based on the page layout and field-level security, you can check the field accessibility grid. From Setup, enter *Field Accessibility* in the Quick Find box, then select **Field Accessibility**. From this page, choose a particular object to view and then select whether you want to check access by profiles, record types, or fields.

See Also

[Page Layouts](#)

[Assign Record Types and Page Layouts in Profiles](#)

[Tailor Business Processes to Different Record Types Users](#)

[Ways to Control User Access to Fields](#)

Edit Multi-Line Layouts for Opportunity Products

Customize the columns that display when users add or edit items in the Products related list of an opportunity detail page.

REQUIRED EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To edit multi-line layouts for opportunity products: Customize Application

In Salesforce Classic, the Multi-Line Layout columns appear when you click **Edit All** from the Products

related list on an opportunity detail page.

In Lightning Experience, the Multi-Line Layout columns appear on the Edit All Products window that appears when you click **Edit Products** or **Add Products | Next** from the Products related list on an opportunity record page.

-  **Note** Multi-line layout is available only for page layouts on certain objects, such as Opportunity Product, Opportunity Split, Order Product, Contract Line Item, and Sales Agreement Product. If you don't see the **Edit Multi-Line Layout** link in the page layout editor on an object layout, then multi-line layout isn't supported for that object.

1. From the object management settings for Opportunity Product, go to **Page Layouts**.
2. Next to the name of an Opportunity Product page layout, click **Edit**.
3. Click **Edit Multi-Line Layout**.
4. Move fields between Available Fields and Selected Fields.

 **Note** The Product and Quantity fields are required and can't be removed from the layout.

5. Click **Save**.

See Also

[Configure Fields on Multi-Line Layouts for Opportunity Products](#)

[The Enhanced Page Layout Editor](#)

[Page Layouts](#)

Configure Fields on Multi-Line Layouts for Opportunity Products

Before you can add a field to the Opportunity Product multi-line layout, the field must be visible on the Opportunity Product page layout. You make the field visible via the Product related list on an Opportunity object page layout.

REQUIRED EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To edit multi-line layouts for opportunity products: **Customize Application**

1. From the object management settings for Opportunity, go to Page Layouts.
2. Next to the name of an Opportunity page layout, click **Edit**.
3. Scroll down to the Related Lists section.
4. On the Products related list, click the wrench icon ().
5. Move fields between Available Fields and Selected Fields.



Note The fields in the Selected Fields area become the list of column options available when you [configure multi-line layouts for Opportunity Products](#).

6. Click **OK**.
7. Save the page layout.

See Also

[The Enhanced Page Layout Editor](#)
[Page Layouts](#)

Customize Related Lists

You can customize the buttons, columns displayed, column order, and record sort order of related lists on record detail pages in Salesforce and the Salesforce Customer Portal.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All Editions except Database.com**

USER PERMISSIONS NEEDED

To customize related lists: Customize Application

1. From the management settings for the object you want to edit, go to Page Layouts.
2. Click on a layout's row in the list view, and select **Edit**, or click the name of the layout.
3. To edit a related list, double-click its tab. If you're using the enhanced page layout editor, you can also click .

A related list configuration window appears.



Note You can't customize the History related list because it links to data stored elsewhere.

4. Select the fields to include in the related list, define the order to display the fields, and select the record sort order.

The default sort order is by record ID. You can include up to 10 fields per related list. To display more than four fields in Lightning Experience, edit the related list component in the Lightning App Builder and choose **Enhanced List** as the related list type.

5. If desired, select other page layouts to apply your related list customizations to.

Only layouts that include this related list appear in the list. Layouts that include related lists with the same customizations as the current layout had when you opened it are selected by default.

Salesforce's assessment of which layouts have the same customizations is based on the state of the current layout prior to editing it. This can lead to previously deselected layouts becoming re-selected when an edited layout is reopened. To avoid unintended changes across multiple layouts, always verify the list of selected layouts.

6. If you have quick actions or custom list buttons configured for the related list's object, from the Buttons section header, click to customize which standard buttons, custom buttons, and quick

actions appear on the related list.



The quick action or custom button must be defined for the object contained in the related list. For example, to display a quick action on the Contacts related list of an account, define the quick action for contacts, not accounts. For custom buttons, the button Type must be List Button.

7. If necessary, click **Revert to Defaults** to undo any customizations and use the default Salesforce settings in the related list.

8. To store your customizations, click **OK**.

The related list changes aren't saved until you save the page layout.

9. Save the page layout.

If you add, move, or remove related lists from the layout, then click to save the page layout, you have the option to apply those related list changes to all users, even if they already customized their display. This action isn't reversible.

To update the related list type, move the related list on the record page, and more, select the related list on the record page in the Lightning App Builder. To further customize the list or to reorder the list's actions from the Lightning App Builder, upgrade to the Dynamic Related List - Single component.



Note From the User Interface settings in Setup, you can also enable related list hover links so that record detail pages include links for each related list at the top of the page. Users can hover over the link to display the corresponding related list in an interactive overlay to view and manage the related list items. Users can also click the link to jump to the content of the related list without scrolling down the page.

See Also

[Page Layout Limitations](#)

[Page Layout Tips](#)

[Ways to Control User Access to Fields](#)

[Knowledge Article: Unable to Edit a Related List](#)

Customize Standard and Custom Buttons on Page Layouts

When customizing page layouts, you can control which standard and custom buttons are displayed and the order in which the custom buttons appear.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To customize detail page buttons:

Customize Application

1. From the management settings for the object whose page layout you want to customize, go to **Page Layouts**.
2. Click **Edit** next to the page layout you want to customize.
3. Add, remove, or reorder the buttons on the layout.
This process is different depending on which editor you're using.
 - In the enhanced page layout editor, select the **Buttons** category on the palette and drag one or more buttons from the palette to the buttons section on the page layout. To remove a button, drag it from the layout back to the palette.
 - In the original page layout editor, double-click the **Detail Page Buttons** item in the Button Section.
4. Save the page layout.



Note Standard and custom buttons are also considered actions. In Lightning Experience, buttons are displayed along with quick actions in different places on a record page depending on the button type.

See Also

[Customize Page Layouts with the Enhanced Page Layout Editor](#)

[Provide Actions, Buttons, and Links](#)

[Override Standard Buttons and Tab Home Pages](#)

[Define Custom Buttons and Links](#)

How Page Layout Elements Display in Lightning Experience

When you change your page layouts in Salesforce Classic, it can impact the content shown on object record pages in Lightning Experience. However, in Lightning Experience, the page elements look different, and some features are not supported.

REQUIRED EDITIONS

Available in: Lightning Experience

Page layouts are available in: **All Editions**

Creation and deletion of page layouts is available in: **Enterprise, Performance, Unlimited, and Developer Editions**

Here's a sample contact record in Lightning Experience. The highlights panel (1) contains key record fields and is the only part of a record page that you can't customize using the page layout editor. The fields in the highlights panel are customized using a compact layout.

These page layout elements are supported in Lightning Experience.

Feature	Description
Actions	<p>Actions display in different places, such as the Highlights Panel (1), Activity tab (2), and the Chatter tab (3). The actions are derived from the list of actions in the Salesforce Mobile and Lightning Experience Actions section of the page layout.</p> <p>Important: Some actions aren't supported in Lightning Experience.</p>
Blank Spaces	Blank spaces are supported in Lightning Experience.
Canvas Apps	Canvas apps are supported in Lightning Experience.
Custom Links	Custom links display under the Details tab (4).
Fields	<p>On pages that don't use Dynamic Forms, fields from the page layout display in a block under the Details tab (4). You can remove or reorder fields on a page layout only via the page layout editor.</p> <p>On Dynamic Forms-enabled pages, fields can be put almost anywhere on the page, using the Lightning App Builder. A page layout defines which fields are available for use on the page, but the organization of the fields is done in the Lightning App Builder. See Break Up Your Record Details with Dynamic Forms.</p> <p>The top-down tab-key order, which allows users viewing a record detail page to move through a column of fields from top to bottom before moving focus to</p>

Feature	Description
	<p>the top of the next column of fields, isn't supported in Lightning Experience. Even if a page layout is configured for a top-down tab-key order, tabbing moves from left-to-right through field columns in Lightning Experience.</p>
Related Lists	<p>Related lists are included as Lightning components in Lightning Experience, and they appear in the Related tab (5). Not all related lists are supported in Lightning Experience.</p> <p>In Lightning Experience, the related list type determines how many fields are displayed in a related list. The Basic List related list type displays only the first four fields of a related list. The Enhanced List type shows up to 10 fields, lets you resize and sort columns, perform mass actions, and wrap text. To change the related list type, customize the Related List-Single component or the Related Lists component in the Lightning App Builder.</p>
Report Charts	<p>Report charts that you add to a page layout appear under the Details tab (4) in Lightning Experience. When you add a report chart to a page layout, it can take a few moments before the chart appears on Lightning record pages.</p>
Sections	<p>On pages that don't use Dynamic Forms, sections appear along with fields under the Details tab (4). A section with no header is incorporated into the section above it.</p> <p>On Dynamic Forms-enabled pages, sections can be put almost anywhere on the page using the Lightning App Builder.</p> <p>See Break Up Your Record Details with Dynamic Forms.</p> <p>A section with no header is incorporated into the section above it.</p> <p>The Detail Page Visibility setting controls whether the section header appears for both the detail page and the edit page. If the section header is set to display (or hide) on the detail page, the header also displays (or hides) on the edit page.</p>
Standard and Custom Buttons	<p>Standard and custom buttons are treated as actions in Lightning Experience, just like in the Salesforce Mobile App. Buttons can appear in various places on a Lightning page, depending on the button type and the page type.</p> <p>Important:Custom buttons that call JavaScript aren't supported in Lightning Experience.</p>
Visualforce Pages	<p>Visualforce pages that you added to the page layout appear under the Details tab (4). Only Visualforce pages with Available for Lightning Experience,</p>

Feature	Description
	Experience Builder sites, and the mobile app enabled display in Lightning Experience on Lightning pages, utility bars, and the Salesforce Mobile App.
Feature	Description
Actions	<p>Actions display in different places, such as the Highlights Panel (1), Activity tab (2), and the Chatter tab (3). The actions are derived from the list of actions in the Salesforce Mobile and Lightning Experience Actions section of the page layout.</p> <p>Some actions aren't supported in Lightning Experience.</p>
Blank Spaces	Blank spaces are supported in Lightning Experience.
Canvas Apps	Canvas apps are supported in Lightning Experience.
Custom Links	Custom links display under the Details tab (4).
Fields	<p>On pages that don't use Dynamic Forms, fields from the page layout display in a block under the Details tab (4). You can remove or reorder fields on a page layout only via the page layout editor.</p> <p>On Dynamic Forms-enabled pages, fields can be put almost anywhere on the page, using the Lightning App Builder. A page layout defines which fields are available for use on the page, but the organization of the fields is done in the Lightning App Builder. See Break Up Your Record Details with Dynamic Forms.</p> <p>The top-down tab-key order, which allows users viewing a record detail page to move through a column of fields from top to bottom before moving focus to the top of the next column of fields, isn't supported in Lightning Experience. Even if a page layout is configured for a top-down tab-key order, tabbing moves from left-to-right through field columns in Lightning Experience.</p>
Related Lists	Related lists are included as Lightning

Feature	Description
	<p>components in Lightning Experience, and they appear in the Related tab (5). Not all related lists are supported in Lightning Experience. In Lightning Experience, the related list type determines how many fields are displayed in a related list. The Basic List related list type displays only the first four fields of a related list. The Enhanced List type shows up to 10 fields, lets you resize and sort columns, perform mass actions, and wrap text. To change the related list type, customize the Related List-Single component or the Related Lists component in the Lightning App Builder.</p>
Report Charts	<p>Report charts that you add to a page layout appear under the Details tab (4) in Lightning Experience. When you add a report chart to a page layout, it can take a few moments before the chart appears on Lightning record pages.</p>
Sections	<p>On pages that don't use Dynamic Forms, sections appear along with fields under the Details tab (4). A section with no header is incorporated into the section above it. On Dynamic Forms-enabled pages, sections can be put almost anywhere on the page using the Lightning App Builder. See Break Up Your Record Details with Dynamic Forms. A section with no header is incorporated into the section above it. The Detail Page Visibility setting controls whether the section header appears for both the detail page and the edit page. If the section header is set to display (or hide) on the detail page, the header also displays (or hides) on the edit page.</p>
Standard and Custom Buttons	<p>Standard and custom buttons are treated as actions in Lightning Experience, just like in the Salesforce Mobile App. Buttons can appear in various places on a Lightning page, depending on the button type and the page type.</p> <p>Custom buttons that call JavaScript aren't supported in Lightning Experience.</p>

Feature	Description
Visualforce Pages	Visualforce pages that you added to the page layout appear under the Details tab (4). Only Visualforce pages with Available for Lightning Experience, Experience Builder sites, and the mobile app enabled display in Lightning Experience on Lightning pages, utility bars, and the Salesforce Mobile App.

These page layout elements aren't supported in Lightning Experience.

- Expanded lookups
- Mobile cards
- S-controls
- Section header visibility for Edit Page
- Tags

The Lightning App Builder is used to customize the layout of Lightning Experience record home pages, not the enhanced page layout editor.

See Also

[User Interface Elements for the Enhanced Page Layout Editor](#)

[Customize Buttons on the Tabbed Activity Composer](#)

[Custom Record Page Settings](#)

Page Layout Tips

Here are a few tips to keep your page layouts organized and easy to use.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Page layouts available in: all editions

Creation and deletion of page layouts available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

- Use field-level security to restrict users' access to fields; then use page layouts to organize detail and edit pages within tabs. This reduces the number of page layouts for you to maintain. Field-level security settings override the visible and read-only settings on the page layout if the field-level security has a more restrictive setting than the page layout.
- Remove unnecessary fields.
- Keep the number of required fields to a minimum.
- Group similar fields with sections.

- Think about the right TAB key order for each section.
- Check your layouts in Read and Edit modes.
- Add help and description text to custom fields. Use it to explain to users what data you're looking for in the field.
- In Professional, Enterprise, Performance, Unlimited, and Developer Editions, use record types to provide unique layouts for different records.
- Optimize related lists—adjust their overall order, the sorting of the records, and display of relevant columns and buttons.
- If you want to customize the user profile layout in the Salesforce mobile app, create a new layout or edit an existing layout in the User Profile Page Layouts section.
- If a dependent lookup is above its controlling field on a page layout, make its lookup filter optional or redesign the page layout. Placing a required dependent lookup above its controlling field on a page layout could confuse users who typically start from the top of the page when entering data.
- A background process periodically runs that cleans up metadata associated with deleted custom fields. This process will affect the **Last Modified Date** and **Last Modified By** fields on page layouts, record types, and custom objects.
- Salesforce recommends creating no more than 200 page layouts. Although there is no limit, it can be difficult to manage your page layouts if you have more than 200.

Page Layout Considerations

Keep these considerations in mind when working with page layouts in the enhanced page layout editor.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Page layouts available in: all editions

Creation and deletion of page layouts available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

 **Note** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Usage Considerations

Layout changes made by administrators for record pages don't appear immediately in Lightning Experience. The record page displays the updated layout about 15 minutes later when you reload the page.

To see your changes immediately, log out and log back in. Other users don't see the change until up to one hour later when they reload the page. This behavior applies to record pages for these objects.

- Account
- Case

- Contact
- Lead
- Opportunity
- Custom objects

This behavior also applies to record layouts updated through the page layout editor, compact layouts, and Lightning pages. For example, the layout is changed when you add a custom field to an object and add that field to the page layout.

For changes to a Lightning page, such as via the Lightning App Builder, administrators continue to see their changes immediately after a page reload. However, other users and other browsers don't see the change up to one hour later when they reload the page. To see the changes immediately, users can log out and log back in.

Page Layouts

- For Personal, Contact Manager, Essentials, and Group Edition orgs, every user views the same layout. Professional, Enterprise, Unlimited, Performance, and Developer Edition orgs can create different page layouts for use by different profiles and record types and set field-level security settings to further restrict users' access to specific fields.
- In Professional, Enterprise, Performance, Unlimited, and Developer Editions, you can set the mini page layouts and related objects that appear in the Console tab.
- Elements that are already on the page layout still appear on the palette but are inactive. When you click an inactive element on the palette, Salesforce highlights the element on the page layout.
- Removing a field from a page layout doesn't remove it from the object's compact layout. The two layout types are independent.
- If the original page layout editor is enabled, users can click the page layout name to access the detail page of the page layout. The enhanced page layout editor doesn't have detail pages, as all the detail page functionality is always available on the enhanced editor. Salesforce displays a read-only version of the enhanced page layout editor to users with the "View Setup and Configuration" permission.



Note The read-only view of the page layout doesn't display field types and lengths in hover details.

- The Custom Links, Custom S-Controls, and Visualforce Pages categories appear in the palette only if you've defined those types of elements for the object for which you're defining a page layout. When you create a custom link for an object, you add it to the Custom Links section on that object's page layout. In non-English Salesforce orgs, the "Custom Links" section title isn't translated from English automatically for the Territory and Territory Model objects, but you can edit the section title.
- The Canvas Apps category appears in the palette only if you defined at least one canvas app with a location of Visualforce Page.
- The Components category appears in the palette only if the available components are supported by the object for which you're defining a page layout. For example, the Twitter component is supported only on account, contact, and lead page layouts.
- When you edit an account page layout for use in Salesforce Classic, the following applies:
 - On business accounts, you can display a Copy Billing Address to Shipping Address link. On the page layout, in the Address Information section, select the option to display the section header on the

Edit page. Next to the Billing Address field, add the Shipping Address field.

- You can also display a link on person accounts. On the page layout, in the Address Information section, select the option to display the section header on the Edit page. Next to the Billing Address field, add the Shipping Address or the Mailing Address field. The link says Copy Billing Address to Shipping (or Mailing) Address.
- Contact fields and related lists are available on person account page layouts, but contact custom links and custom buttons aren't.
- Currently, you can't change the location of Chatter feeds. However, in Salesforce Classic, users can click the  Hide Chatter link in a Chatter feed to hide the feed, and the  Show Chatter link to show the feed.
- Changes to user layouts override the global publisher layout on user profile pages and the Chatter home page.
- Occasionally, a new object is created without a default page layout. To trigger creation of a default page layout for an object that doesn't have one, visit the object's list view page.
- When you use the Translation Workbench to set a translation for a layout section label, the master layout section label can sometimes be overwritten by a user-language translation label. Here's how it happens: When you open the page layout editor, it displays the translated section labels based on the user's language. Then when you save the layout, the editor sometimes updates the section's master label to match the translated label.
- If one or more profiles have page layout assignments missing for an object, then saving of new records and editing existing ones are prevented for that object.

Mini Page Layouts

- In Professional, Enterprise, Performance, Unlimited, and Developer Editions, you can set the mini page layouts and related objects that appear in the Console tab.
- Field properties on the page layout determine field properties on the mini page layout. For example, if a field is read-only on the page layout, that same field is read-only on the mini page layout. To change the field properties of fields on the mini page layout, you must change the field properties of fields on the page layout.
- Overrides for the Edit and View buttons for an object don't affect the Edit and View buttons in mini page layouts.
- Fields marked **Always Displayed** or **Always on Layout** on page layouts are automatically included on the mini page layout and can't be removed unless they're removed from the page layout.

Knowledge Layouts

- Authoring actions that you add to the Salesforce Mobile and Lightning Experience Actions section of the page layout appear in the highlights panel on record pages in Lightning Experience and the Salesforce mobile app.
- To use inline edit with Knowledge, add the Publication Status field to your standard page layout. The Publication Status field must be in the standard page layout, not in a compact layout. However, the field can appear in both the standard and compact layouts.



Tip If the Publication Status field is in a collapsed layout section, you must expand the section to

load the edit icons before you can use inline editing. To increase the accessibility of inline editing, add the Publication Status field to a layout section that is likely to always be open.

- The Title and URL Name standard fields are required. You can't remove them from the layout.
- To control which audiences can view an article, add these fields to the page layout: Visible in Internal App; Visible to Customer; Visible to Partner; and Visible in Public Knowledge base. The fields appear as checkboxes in the record.

See Also

[Tips for Using the Enhanced Page Layout Editor](#)

[Page Layout Limitations](#)

[User Interface Elements for the Enhanced Page Layout Editor](#)

Page Layout Limitations

Keep these limitations in mind when working with page layouts in the enhanced page layout editor.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Page layouts available in: all editions

Creation and deletion of page layouts available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Page Layouts

- If you're using Salesforce Professional Edition, you can't rename a page layout.
 - You can drag up to 20 s-controls, 20 Visualforce pages, 20 expanded lookups, and 100 related lists onto a page layout. There are no limits on custom links.
-  **Note** You can't place a Visualforce page more than one time on a page layout.
- In Lightning Experience, page layouts support up to 55 lookup fields. In Salesforce Classic, page layouts support up to 40 lookup fields.
 - Don't add more than four external lookup fields to your page layout. On Lightning Experience record pages, a Record Detail component that contains more than four external lookup fields breaks the page at runtime.
 - You can add one dashboard per page layout.
 - Page layouts for the user object only include custom fields, custom links, s-controls, and Visualforce pages. Tagging, related lists, custom buttons, and standard field customizations aren't included on page layouts for the user object. Also, field-level security is only available for custom fields on the user object. Only standard Chatter actions appear on the user profile page, regardless of which actions are assigned to the User Page Layout or the global publisher layout.
 - You can edit only certain attributes when you're working with a page layout that was installed from a managed app. Some changes that you make to managed page layouts, such as adding components,

- work when you're editing the page layout but aren't reflected on the record detail page.
- Chatter group layout changes affect the Salesforce mobile app only. Changes to the group publisher (actions and layout) reflect in both the full Salesforce site and the Salesforce mobile app.
 - Custom fields installed from a managed package aren't translated when they appear in the page layout editor.
 - Due to third-party cookie restrictions in modern web browsers, Visualforce pages embedded in page layouts sometimes don't render properly in Salesforce Classic. See [Embedded Visualforce page not displayed in record page layout in Classic UI](#).

Related Lists

- A page layout can have up to 100 related lists.
- The **View All** button only displays up to 2,000 items in a related list.
- You can't add related lists to the page layouts for the User object.
- You can include up to 10 fields per related list.
- In Lightning Experience, the related list type determines how many fields are displayed in a related list. The Basic List related list type displays only the first four fields of a related list. The Enhanced List type shows up to 10 fields, lets you resize and sort columns, perform mass actions, and wrap text. To change the related list type, customize the Related List-Single component or the Related Lists component in the Lightning App Builder.
- Users can't drag documents to add them to Files or Notes & Attachments related lists when the related list type is **Basic List**.
- You can't move the first field of a related list, because it's a unique identifier for the record.
- You can add custom fields of the long text area type to a related list. However, you can't add some standard fields of the long text area type. For example, you can't add the Description field on an Opportunity to a related list.
- The default sort order varies per record. The Sort By dropdown isn't available for activities and opportunity products.
- Lookup fields aren't available for display on their corresponding lookup related list. For example, the case lookup field on an account page layout isn't available when editing the cases related list.
- You can't customize the History related list because it links to data stored elsewhere.

Mini Page Layouts

- You can't choose Mini Console View for the Close Case layout, Log a Case page, or View Cases page layouts on the Self-Service Portal. You can't choose Mini Console View for opportunity team page layouts.
- You can't define mini page layouts for the Close Case layout, Log a Case page, or View Cases page layouts on the Self-Service Portal. You can't define mini page layouts for opportunity team page layouts.
- You can define mini page layouts for the user object; however, you can't add standard fields or related lists. Also, a customized mini page layout won't display in the Agent console.

See Also

[Tips for Using the Enhanced Page Layout Editor](#)

Page Layout Considerations

How Page Layouts Work in the Salesforce Mobile App

Use the enhanced page layout editor to customize the layout of an object's record detail pages, configure actions, and adjust which fields and related lists appear in the Salesforce mobile app.

REQUIRED EDITIONS

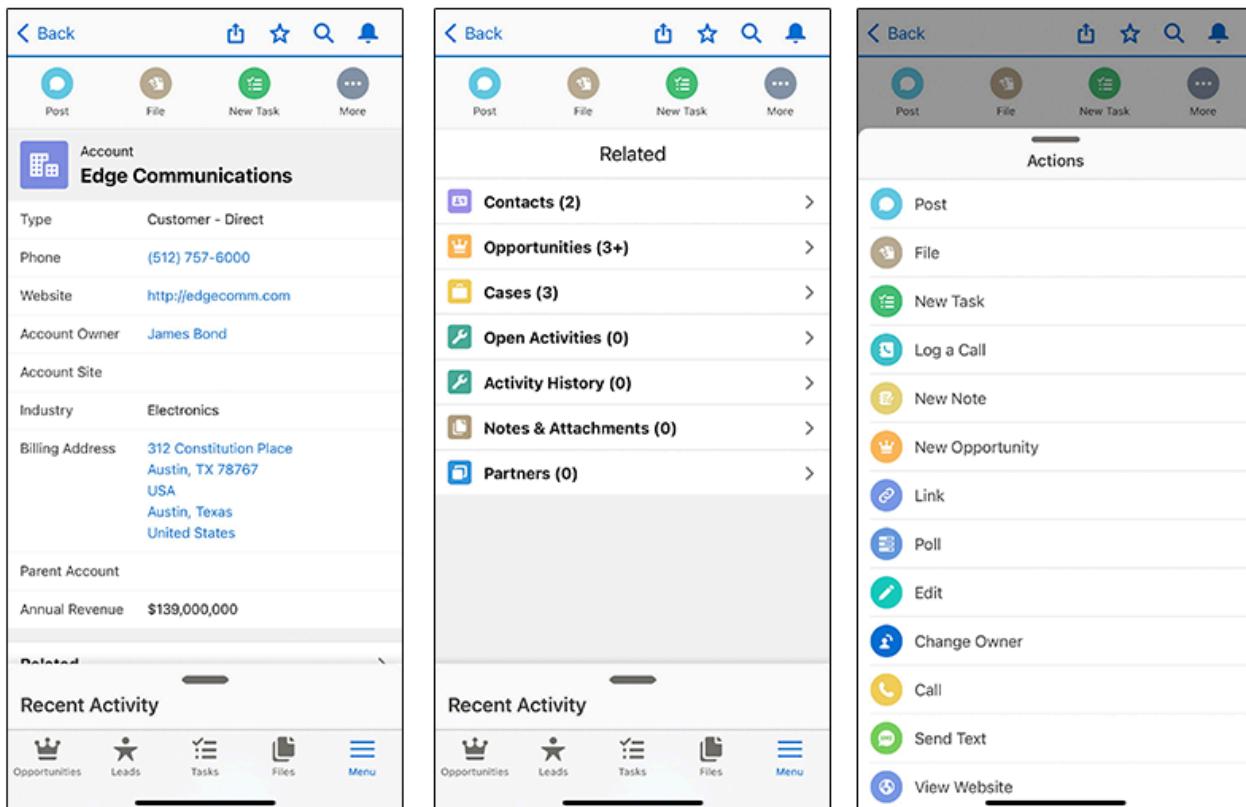
Available in: both Salesforce Classic and Lightning Experience

Available in: All editions except Database.com

In the Salesforce mobile app, page layouts drive these areas of the mobile experience.

- Record Related Information and Detail Pages—When you view a record in the mobile app, you see the fields, Visualforce pages, and related lists that are based on the record type and the user's profile. Related lists show up as single-line cards containing the name of the page or related list. Tapping the related list card displays its details.
- Actions—Actions in the Salesforce Mobile and Lightning Experience Actions section of a page layout appear in the action bar and action menu on the object's record pages.

 **Example** Here are the record details, related lists, and action menu for a sample account, Edge Communications:



Tips for Optimizing Page Layouts for the Salesforce Mobile App

Here are some tips and tricks for making your existing page layouts more mobile-friendly.

Tips for Optimizing Page Layouts for the Salesforce Mobile App

Here are some tips and tricks for making your existing page layouts more mobile-friendly.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: All editions except Database.com

Page layouts containing dozens of fields and lots of related lists might be manageable when viewing records on a computer screen, but on a small mobile device, viewing that same record can be overwhelming. People accessing information using a mobile device are looking for a quick way to get what they need, and making your users sift through hundreds of fields and related lists just doesn't make sense.

When optimizing a page layout, consider:

- What are the important things to see at a glance?
- What are the important moments for your users when they're working in the Salesforce mobile app?
- What actions or processes can you automate so that your users don't have to manually do them?

The Key: Organize and Minimize Fields

- Use sections to organize information logically, putting the most important things at the top of the page so they show up first. Your users don't want to search for fields individually. Organizing similar fields in sections will help your users find what they need. They can then easily scroll down the page to the section they care about.
- For accounts, contacts, and leads, you don't need to put phone or email fields near the top. They're already quickly accessible via the  and  icons on each record page's action bar.
- You don't need to keep fields in one column, as the page renders dynamically. Salesforce for iOS or Android will reorder the fields into a single column, while web browsers can show two columns.
- Put the most important fields into the compact layout—which drives record highlights and record preview cards in the mobile app—so they're available right up front, and so your mobile users don't have to drill into the record detail.
- Keep the number of required fields to a minimum. Setting a field to required means it must appear on the detail page of all page layouts, so consider whether each field is truly required. You might have to convince stakeholders that a field isn't actually necessary for a record to be saved.
- If available in your org, think about using record types so that fields that aren't common to all records don't have to appear on all records.
- To reduce the number of fields on a screen, consider using default values for new records instead of having the user enter the data.

See Also

[How Page Layouts Work in the Salesforce Mobile App](#)

Manage Mobile Cards in the Enhanced Page Layout Editor

Add expanded lookups, components, and Visualforce pages to the Mobile Cards section of your page layout to have them show up as mobile cards in the Salesforce mobile app.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** editions except Database.com

USER PERMISSIONS NEEDED

To create and edit page layouts: Customize Application

! **Important** The Mobile Cards section of the page layout editor is no longer supported as of Spring '20. Items in the Mobile Cards section, such as components and expanded lookups, no longer display in the Salesforce mobile app. For more information, see [Mobile Cards Are Not Available in the New Salesforce Mobile App](#) in the Salesforce Spring '20 release notes.

Feed-Based Layouts in Salesforce Classic

Feed-based page layouts make it easier to work with records by providing two separate views: one for the record's feed, and one for its details, including related lists.

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

Feed-based layouts offer a more streamlined way of working with records, and don't require users to scroll through information they're not interested in to find what they're looking for. Users can easily switch back and forth between the feed view, which includes the publisher and important events on the record, shown in chronological order, and the details view, which shows in-depth information about the record, including related lists.

Unlike standard page layouts, which include all of a record's information—the feed, details, and related lists—on one page, feed-based layouts let you switch between the feed view and the details view so you can focus on the type of information you need at any given moment. For example, to see comments others have made about a record or to create a record that's related to it, you'd use the feed view. To delve into the record's related lists, attachments, and other in-depth information, you'd use the details view.

Feed-based layouts are available on account, asset, case, contact, lead, opportunity, custom, and external objects. To create feed-based layouts for cases, use Case Feed.



The feed view in these layouts includes:

- Tabs or, if you're working in the Salesforce console, toggle buttons (☰) to switch between the feed view and the detail view. (1)
- The publisher, which might include actions that let you do things like create related records or log calls, depending on how your administrator has set up your organization. (2)
- The record feed, which shows activity on the record, such as comments others have made about it. (3)
- Any custom buttons and links your administrator has added. (4)
- A follow button or following indicator and a list of people who follow the record. (5)
Depending on how your administrator has set up the page, these might appear on the left side or on the right side.
- Feed filters, which let you choose which information from the feed you see. (6)
Depending on how your administrator has set up the page, the filters might appear on the left side of the page, in the center, or on the right.

Detail views show in-depth information about the record, including related lists.



[Create Feed-Based Page Layouts in Salesforce Classic](#)

Make it easier for your users to work with account, contact, lead, opportunity, custom, and external object records by creating feed-based layouts. These layouts include two separate views: one for the record's feed and one for its details.

Create Feed-Based Page Layouts in Salesforce Classic

Make it easier for your users to work with account, contact, lead, opportunity, custom, and external object records by creating feed-based layouts. These layouts include two separate views: one for the record's feed and one for its details.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, and Developer Editions**

USER PERMISSIONS NEEDED

To create, edit, and delete page layouts: Customize Application

Before you begin, ensure that feed tracking is enabled for the object on which you want to create a feed-based layout. To enable feed tracking, from Setup, enter *Feed Tracking* in the Quick Find box, then select **Feed Tracking**.

1. [Create a new page layout](#) and select **Feed-Based Layout**.



Note Only new page layouts can be feed-based; you can't change an existing standard layout to a feed-based layout.

2. Click **Edit** next to your layout and [use the enhanced page layout editor to configure it](#).

You can't configure feed-based layouts with the original page layout editor.

3. On the main page layout editor page, [customize the publisher](#) to include the actions you want to make available to users, and add any custom buttons or links.

4. Click **Feed View** in the page layout editor header to customize what appears on the feed page.

You can:

- Enable full-width feed so the feed expands horizontally when users view records in Salesforce console tabs or subtabs.
- Turn on compact feed so users see a cleaner, more streamlined feed view when working with records in Salesforce console tabs or subtabs.
- Choose to automatically collapse the publisher when it's not in use so users can see more of the information below it on the page.
- Add custom components, which are Visualforce pages with functionality you define.
- Choose where on the page custom buttons and links and standard components like the Follow button and followers list appear.
- Hide the standard sidebar.
- Choose which feed filters are available, and where they appear.

5. [Assign the page layout to user profiles](#).

See Also

[Feed-Based Layouts in Salesforce Classic](#)

[Use Case Feed in Salesforce Classic](#)

[Feed Tracking](#)

[Enable Feed Updates for Related Records](#)

Salesforce Classic Home Tab Page Layouts

You can customize the Home tab in Salesforce Classic to include components such as sidebar links, a company logo, a dashboard snapshot, or custom components that you create. A dashboard snapshot is a clipping of the top row of a dashboard's components. Just like other tabs, you can also assign different home page layouts to different users based on profile.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **All** Editions

You can add components to the sidebar or the main panel. You can also determine if custom sidebar components appear only on the Home tab or on all Salesforce pages.

[Create Custom Home Page Components](#)

Use custom components to configure the Salesforce Classic home page for your users. Add HTML, images, links, and more to enhance your users' productivity.

[Design Home Page Layouts in Salesforce Classic](#)

After creating the components you want displayed on the Home tab, design your home page layouts. You can design your layouts based on your unique organizational and user needs.

[Visualforce Area Home Page Components](#)

Use Visualforce Area home page components to add dynamic content to your home page. For example, you can present content from partner apps, display charts with the Reports and Dashboards REST API, or add a canvas app to the home page.

[Home Page Components Tips and Considerations](#)

Keep these considerations in mind when creating custom components that you want displayed on the Salesforce Classic Home tab.

[Assign Home Tab Page Layouts to Profiles](#)

Your home page layouts are only visible to users after you assign them to a user profile.

Create Custom Home Page Components

Use custom components to configure the Salesforce Classic home page for your users. Add HTML, images, links, and more to enhance your users' productivity.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **All** Editions

USER PERMISSIONS NEEDED

To create or change home page layouts: Customize Application

Before you begin:

- If you're creating custom link components, define your Home tab custom links first. See [Custom Buttons and Links](#).
- If you're creating an image component, upload your image to the Documents tab first.
- If you're creating a Visualforce Area component, create your Visualforce page first.

1. From Setup, enter *Home Page Components* in the Quick Find box, then select **Home Page Components**.
2. Click **New**.
3. Enter a name for the component. For custom links, this name is displayed as the section heading in the sidebar on the Home tab.
4. Choose the type of component.
5. Click **Next**, and then complete one or more of these steps.
 - For links, select the appropriate custom links, and then click **Add**.

- For images, click **Insert an image**, choose the document folder, and then select the image file. The image file must be in a public folder and **Externally Available** must be enabled on the document's properties so that users can view the image.
Keep your image size smaller than 20 KB for optimum performance.
- For an HTML Area component, choose where to display it—in the wide or narrow column—and then enter your content in the box below.
HTML Area home page components don't support JavaScript, CSS, iframes, and some other advanced markup. To use JavaScript or other advanced HTML elements in your home page component, we recommend that you use a Visualforce Area component instead.
- For a Visualforce Area component, choose where to display it—in the wide or narrow column—then select the Visualforce page, and assign it a height.

6. Click **Save**.

After creating the home page component, you need to add it to a home page layout.

-  **Note** Components in the narrow column are displayed in the sidebar. They aren't displayed in the sidebar on other pages in Salesforce unless you specify that in your user interface settings or by assigning the Show Custom Sidebar On All Pages permission.

See Also

- [Visualforce Area Home Page Components](#)
[Home Page Components Tips and Considerations](#)

Design Home Page Layouts in Salesforce Classic

After creating the components you want displayed on the Home tab, design your home page layouts. You can design your layouts based on your unique organizational and user needs.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **All** Editions

USER PERMISSIONS NEEDED

To view home page layouts: View Setup and Configuration

To create or change home page layouts: Customize Application

1. From Setup, enter *Home Page Layouts* in the Quick Find box, then select **Home Page Layouts**.
2. Click to edit an existing layout or create one. Alternately, select a layout to copy and click **Clone**.
3. If you create a layout, give it a name and then click **Save**.
4. Select the components to display on the layout.
 - To add the Find Articles component, select **Article Search**. This component is only available for Salesforce Knowledge users.
 - To add the Customer Portal component, select **Customer Portal Welcome**. If the My Profile site

Visualforce page has been enabled, this component contains a personalized welcome message and a link to the portal user's profile. The My Profile page enables users logged into either your Salesforce site, or your Customer Portal from Salesforce sites, to update their own contact information. When they change this page, the corresponding portal user and contact records are updated.

- To allow your users to resume flow interviews that they've paused, select **Paused Flow Interviews**. This component displays only flow interviews that the user has paused. You can add up to 20 components to a home page layout.

5. Click **Next**.
6. Customize the order in which the narrow and wide components appear. Move a component by selecting it and using the arrow buttons.
7. Click **Save**.

See Also

- [Assign Home Tab Page Layouts to Profiles](#)
[Salesforce Classic Home Tab Page Layouts](#)

Visualforce Area Home Page Components

Use Visualforce Area home page components to add dynamic content to your home page. For example, you can present content from partner apps, display charts with the Reports and Dashboards REST API, or add a canvas app to the home page.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **All** Editions

The Visualforce page that you choose for the component can use a standard or custom controller. You can include JavaScript in your Visualforce page, but because the component is rendered in an iframe on the home page layout, the JavaScript can't interact with the page that contains the component.

 **Note** Due to third-party cookie restrictions in modern web browsers, Visualforce Area home page components don't render properly in Salesforce Classic when third-party cookies are disabled. See [Visualforce Limitations in Salesforce Classic When Third-Party Cookies are Blocked](#).

Sample Usage

If your Visualforce Area home page component displays in the sidebar, you can dynamically get the record ID and top-level URL of the page that the component is being displayed on by using the `$CurrentPage` global variable in your Visualforce markup.

Using `$CurrentPage`, you can access the query string parameters for the page by specifying the `parameters` attribute, after which you can access each individual parameter:

```
$CurrentPage.parameters.parameter_name
```

The parameters for record ID and top-level page URL are, respectively, `id` and `sfdcIFrameOrigin`. For more information, see [Getting Query String Parameters](#) in the *Visualforce Developer Guide*.

Home Page Components Tips and Considerations

Keep these considerations in mind when creating custom components that you want displayed on the Salesforce Classic Home tab.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **All** Editions

- Standard components without an **Edit** link are read only.
- The components that you select for the narrow column display in the sidebar. They don't display in the sidebar on other pages within Salesforce unless you specify that in your user interface settings. If you only want certain users to view sidebar components on all pages, grant those users the "Show Custom Sidebar On All Pages" permission.
- When editing the standard Messages & Alerts component, enter the text that you want to display to users. If entering HTML code for your message, make sure that it's self-contained, well-formed HTML.



Note Standard Messages & Alerts home page components don't support JavaScript, CSS, iframes, and some other advanced markup.

- When editing the standard Custom Links home page component, enter the link text to display to users in the Bookmark field. In the URL field, enter the complete website address, such as `http://www.yahoo.com`. To link to a Salesforce page, enter only the part of the URL after `salesforce.com`, for example, `/000x000000esq4`. These links always open within the main Salesforce window, not in a popup window.
- The standard Custom Links home page component is a quick way to add links to the sidebar, but it doesn't support merge fields, functions (such as `URLFOR`), executing JavaScript, or customizable window opening properties. If you need this additional functionality, you can:
 - Create your home page custom links from the **Customize | Home | Custom Links** node in Setup.
 - Create a custom home page component of type **Links** on the **Customize | Home | Home Page Components** page in Setup that includes the custom links that you created in the first step.Creating a custom home page component for your links doesn't change the visual styling for your end users.
- The Dashboard Snapshot component displays the top three components of the last dashboard the user accessed. Users can view a dashboard snapshot on their Home tab if they have access to at least one dashboard.
- When designing home page layouts for your Customer Portal, we recommend adding the following components: Search, Solution Search, Recent Items, Customer Portal Welcome, and a custom HTML Area component that includes your corporate branding in the wide column.

- You can add up to 20 components to a home page layout.
- Due to third-party cookie restrictions in modern web browsers, Visualforce Area home page components don't render properly in Salesforce Classic when third-party cookies are disabled. See [Visualforce Limitations in Salesforce Classic When Third-Party Cookies are Blocked](#).

See Also

[Create Custom Home Page Components](#)

Assign Home Tab Page Layouts to Profiles

Your home page layouts are only visible to users after you assign them to a user profile.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **All** Editions

USER PERMISSIONS NEEDED

To assign home page layouts: Customize Application

1. From Setup, enter *Home Page Layouts* in the Quick Find box, then select **Home Page Layouts**.
2. Click **Page Layout Assignment**.
3. Click **Edit Assignment**.
4. Choose the appropriate page layout for each profile.
Initially, all users, including Customer Portal users, are assigned to the Home Page Default layout.
5. Click **Save**.

 **Tip** Users can customize the dashboard settings on their Home tab in their personal settings.

Customize Page Layouts with the Original Page Layout Editor

Use the original page layout editor to customize page layouts in Salesforce, the Self-Service Portal, and the Salesforce Customer Portal.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **All** Editions

USER PERMISSIONS NEEDED

To customize page layouts: Customize Application

 **Note** We recommend using the enhanced page layout editor instead of the original page layout editor because it offers more features and an easier-to-use interface.

1. From the management settings for the object that you want to edit, go to **Page Layouts**.
2. If tags are enabled, specify whether personal and public tags should be included in the header section of the page layout. Users can tag a record only if personal or public tags are included here.
 - To add personal or public tags, select **Header Items** from the View dropdown list and then drag the Personal Tags or Public Tags items to the header section. You can't change the order in which personal and public tags appear when both are in the header section at the same time.
 - To remove tags, drag the Personal Tags and Public Tags items from the header section to the area under the View dropdown list.
3. To customize buttons, double-click **Detail Page Buttons** in the Button section.
4. To arrange fields, custom s-controls, Visualforce pages, custom links, and related lists on the layout, select one or more items from the box on the right and drag them to the desired location. You can drag up to 20 s-controls, 20 Visualforce pages, 20 expanded lookups, and 100 related lists onto a page layout. There are no limits on custom links.

 **Note** You can add a Visualforce page to a page layout only if the standard controller on the Visualforce page is set to the object for which you're creating the page layout. If you don't have any Visualforce pages with a standard controller set to that object, the Visualforce Pages category doesn't appear in the palette.

5. To set which fields are required and read only, select one or more fields and click **Edit Properties**.
 - You can change custom fields only if they aren't universally required fields.
 - Fields marked as read only are always editable by administrators and users with the Edit Read Only Fields permission.
 - If you make a picklist field read only, all new records contain the default value for that picklist.
 - Auto-number fields are always read only.
 - If you make the opportunity Probability field read only, the Probability value still updates automatically when a user changes the Stage value of an opportunity.
 - In Professional, Enterprise, Unlimited, Performance, and Developer Editions, field-level security settings override the field properties you set here if the field-level security is more restrictive than the page layout setting.
6. To change the properties of an s-control or Visualforce page, double-click it and set the following attributes.
 - Width sets the horizontal size in pixels or a percent.
 - Height sets the vertical size in pixels.
 - Show scrollbars determines whether the iFrame in which the s-control displays contains scrollbars when necessary.
 - Show label determines whether the page layout includes the label of the custom s-control. Remove the label to display the custom s-control in a wider area.
7. To organize the page using sections, click **Edit** next to an existing page section, or click **Create New Section**.
8. To customize related lists on the page layout, double-click a related list in the Related List section. Some related lists aren't customizable because they link to data rather than store it. You can move your cursor over any related list section to see if it's customizable. Also, lookup fields aren't available for

display on their corresponding lookup related list. For example, the case lookup field on an account page layout isn't available when editing the cases related list.

9. To apply the related lists in the page layout to all users, even if they've already customized their display, select **Overwrite users' customized related lists**.
10. To review the page layout, click **Preview**. From the preview in Enterprise, Unlimited, Performance, and Developer Editions, select a profile to see how the pages look for users with different profiles. Most related lists' columns preview without data.
11. Click **Save** to finish. Alternatively, click **Quick Save** to save and continue editing the page layout.

In Professional, Enterprise, Unlimited, Performance, and Developer Editions:

- To choose which related records display in the Console tab's mini view, click **Mini Console View**.
- To define the mini page layouts of the records that appear in the Console tab's mini view, click **Mini Page Layout**.

 **Note** You can't define mini console views or mini page layouts for the Close Case Layout or the Log a Case Page and View Cases Page layouts on the Self-Service Portal.

In Enterprise, Unlimited, Performance, and Developer Editions:

- You can assign page layouts for different profile and record type combinations.
- You can set field-level security to restrict field access further.

See Also

[Considerations for Using the Original Page Layout Editor](#)

[Customize Related Lists](#)

[Page Layouts](#)

Considerations for Using the Original Page Layout Editor

When working with the Original Page Layout Editor in Salesforce Classic, keep these considerations in mind.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: All Editions

- When customizing page layouts for tasks, you can select these save options to appear when users create or edit a task. These options aren't available when user control over task assignment notifications is enabled.
 - **Show Task Email Notification checkbox**—Displays the **Send Notification Email** checkbox when users create or edit a task.
 - **Select Task Email Notification checkbox by default**—Selects the **Send Notification Email** checkbox by default when users create or edit a task. A user's personal preference for defaulting the state of

the checkbox takes precedence over the org-wide setting.

- When customizing page layouts for cases, you can select these save options to appear when users create or edit a case. These options aren't available when user control over case assignment notifications is enabled.
 - **Show on edit page Case Assignment checkbox**—Displays the **Assign using active assignment rules** checkbox when users create or edit a case.
 - **Default Case Assignment checkbox**—Automatically runs case assignment rules. If both **Show on edit page** and **Select by default** are selected, the assignment checkbox is selected by default, but users can deselect it to override the assignment rule.
 - **Show Case Email Notification checkbox**—Displays the **Send Notification Email** checkbox when users create or edit a case.
 - **Select Case Email Notification checkbox by default**—Selects the **Send Notification Email** checkbox by default when users create or edit a case.
- Page layouts for the user object only include custom fields, custom links, s-controls, and Visualforce pages. Tagging, related lists, custom buttons, and standard field customizations are *not* included on page layouts for the user object. Also, field-level security is available only for custom fields on the user object.
- You can define mini page layouts for the user object; however, you can't add standard fields or related lists. Also, a customized mini page layout won't display in the Agent console.
- Users can import values into a field only if they have read and edit access. User permissions, page layout assignments, and field-level security settings determine field access.
- In Personal, Contact Manager, and Group Editions, page layouts control which fields users can access in related lists, list views, reports, Connect Offline, email and mail merge templates, custom links, and when synchronizing data. In Professional, Enterprise, Unlimited, Performance, and Developer Editions, this access is controlled by field-level security.
- In Professional, Enterprise, Unlimited, Performance, and Developer Edition, individual users can customize which tabs and related lists display for their personal use.
- When editing a person account page layout, if you add Shipping Address next to Billing Address in the Address Information section, a link displays on the person account edit page that allows you to copy the billing address to the shipping address. Also, an equivalent link appears if you add Other Address to the Address Information section.
- Some items can only be moved to certain sections on the page layout. For example, you can drag a custom s-control to any field section on the page layout but not a related list section or button section.
- Create the appropriate buttons before editing your page layout. For example, create an account custom button for the detail page and a contact custom list button before putting them both on an account page layout.
- If you use the original page layout editor to view a page layout that was created in the new page layout editor, the original page layout editor will show any blank spaces you added. You can't move or add blank spaces in the original page layout editor, but you can remove them by dragging them to the box on the right.

See Also

[Customize Page Layouts with the Original Page Layout Editor](#)

Compact Layouts

A compact layout displays a record's key fields at a glance in the Salesforce mobile app, Lightning Experience, and in the Outlook and Gmail integrations.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: all editions except **Database.com**

Creating and customizing compact layouts for objects isn't required, because system defaults are provided out of the box. However, we recommend using compact layouts to put important fields into object record headers—and elsewhere—to help your users get the information they need quickly.

In the Salesforce mobile app, the fields that you assign to a compact layout appear in:

- An object record's highlights area (shows up to ten fields)
- Expanded lookup cards on a record's related information page (shows the first four fields)

In Lightning Experience, up to the first seven fields in a compact layout appear in the highlights panel of an object record. (On smaller screens, the highlights panel displays fewer fields.) When a user hovers over a lookup relationship field on the object record page, a highlights panel for that field also displays the first seven fields from the compact layout. Highlights panels display the first field from the compact layout at the top in an accented font.

OPPORTUNITY

Burlington Textiles Weaving Plant Generator

Account Name Burlington Textiles Corp of America	Close Date 9/4/2016	Amount \$235,000.00	Opportunity Owner Madison Ri...
---	------------------------	------------------------	------------------------------------

Prospect... Qualifica... Proposal... Negoti...

ACTIVITY

Opportunity Owner
Madison Rigsby

Private

Opportunity Name
Burlington Textiles Weaving Plant Gen

Account Name
Burlington Textiles Corp of America

Type
New Customer

Lead Source
Web

Order Number
645612

Current Generator(s)
John Deere

Tracking Number
830150301360

CHATTER

Burlington Textiles C...

Type Customer - Direct	Phone (336) 222-7000
Website www.burlington.com	Account Owner Madison Rigsby

Opportunities (1)

Burlington Textiles Weaving Plant Generator

Stage: Prospecting
Amount: \$235,000.00
Close Date: 9/4/2016

[View All](#)

Cases (2)

00001019

Contact Na... Jack Rogers
Subject: Structural failure of generator base
Priority: High

00001020

Contact Na... Jack Rogers
Subject: Power generation below stated level
Priority: Medium

[View All](#)

In the Outlook and Gmail integrations, up to the first three fields in a compact layout appear for records related to an email or event.

As with page layouts, there are separate compact layouts for each object. By default, each object derives its record highlight fields, preview cards, and action-related feed items from the predefined set of fields in the object's read-only, system default compact layout. You can create custom compact layouts on an object-by-object basis. After you create one or more custom compact layouts, you set one as the primary compact layout for the object. The primary compact layout is then used as the new default for that object.

If you have record types associated with an object, you can override the object's primary compact layout and assign different compact layouts to some or all the record types. Each record type can have only one compact layout assigned to it.

Event and task compact layouts determine the fields that appear in the details section when you expand

an activity in the activity timeline in Lightning Experience. When you change the compact layout for tasks in the activity timeline, you also impact the fields that show up in the highlights area on tasks, in tasks lists, and everywhere else the compact layout is used.

Create Compact Layouts

Use compact layouts to customize the fields that display for object records when viewed in the Salesforce mobile app and Lightning Experience.

Assign Compact Layouts to Record Types

As with page layouts, there are separate compact layouts for each object. By default, each object derives its record highlight fields, preview cards, and action-related feed items from the predefined set of fields in the object's read-only, system default compact layout. You can create custom compact layouts on an object-by-object basis. After you create one or more custom compact layouts, you set one as the primary compact layout for the object. The primary compact layout is then used as the new default for that object.

Compact Layout Limitations and Considerations

Keep these limitations and considerations in mind when using compact layouts.

See Also

[Activity Timeline Customization Considerations](#)

Create Compact Layouts

Use compact layouts to customize the fields that display for object records when viewed in the Salesforce mobile app and Lightning Experience.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: All editions except **Database.com**

USER PERMISSIONS NEEDED

To customize compact layouts: Customize Application

To view compact layouts: View Setup and Configuration

1. From the management settings for the object that you want to edit, go to Compact Layouts.
2. Create a new compact layout and give it a label.
3. Add up to 10 fields.

 **Tip** Put the object's Name field first to provide context for your users when they view a record.

4. Sort the fields by selecting them and clicking **Up** or **Down**.
The order you assign to the fields determines the order in which they display.
5. Save the layout.
6. To set the compact layout as the primary compact layout for the object, click **Compact Layout**

Assignment.

-  **Example** Here's a sample compact layout edit page for the Account object. It shows the name of the layout and a list of fields to display.  Here's the related page for the same account object in Lightning Experience, displaying six of the eight fields assigned to the compact layout. You can see the account's name, phone number, business hours, website, owner, and type at the top of the page.  And here's what that same account record looks like in the mobile app. 
-  **Note** Up to ten fields on your compact layout populate the record highlights section at the top of each record view in the Salesforce mobile app. The record highlights section in Lightning Experience uses the first seven fields on the compact layout. However, the number of fields that display can vary based on the width of your screen, which record page is being viewed, and the permissions of the user.

See Also

- [Assign Compact Layouts to Record Types](#)
- [Compact Layout Limitations and Considerations](#)
- [Compact Layouts](#)
- [Customize Case Hovers in Lightning Experience](#)

Assign Compact Layouts to Record Types

As with page layouts, there are separate compact layouts for each object. By default, each object derives its record highlight fields, preview cards, and action-related feed items from the predefined set of fields in the object's read-only, system default compact layout. You can create custom compact layouts on an object-by-object basis. After you create one or more custom compact layouts, you set one as the primary compact layout for the object. The primary compact layout is then used as the new default for that object.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Compact layouts are available in: **All** editions except **Database.com**

Record types are available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To customize compact layouts: [Customize Application](#)

To view compact layouts: [View Setup and Configuration](#)

If you have record types associated with an object, you can override the object's primary compact layout and assign different compact layouts to some or all the record types. Each record type can have only one compact layout assigned to it.

- From the management settings for the object that you want to edit, go to Compact Layouts.

 **Tip** For Salesforce Knowledge articles, from Setup, enter *Knowledge Article Types* in the Quick Find box, then select **Knowledge Article Types**, click the name of an article type, then scroll down to the Compact Layouts related list.

- Click **Compact Layout Assignment**.
- Select a compact layout to use as the primary compact layout for this object.
- In the Record Type Overrides section, select one or more record types to which you want to assign a compact layout.

If you don't have record types set for the object, you won't see this section. If you don't set any record type overrides, all record types use the object's primary compact layout by default.

Some record types in the list might be inactive. You can assign a compact layout to an inactive record type.

- Select a compact layout from the Compact Layout To Use dropdown list to assign it to the selected cells.
- Click **Save**.

See Also

- [Create Compact Layouts](#)
- [Compact Layout Limitations and Considerations](#)
- [Compact Layouts](#)
- [Find Object Management Settings](#)

Compact Layout Limitations and Considerations

Keep these limitations and considerations in mind when using compact layouts.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: all editions except **Database.com**

Considerations

- Up to ten fields on your compact layout populate the record highlights section at the top of each record view in the Salesforce mobile app. The record highlights section in Lightning Experience uses the first seven fields on the compact layout. However, the number of fields that display can vary based on the width of your screen, which record page is being viewed, and the permissions of the user.
- Changes you make to a compact layout are reflected in both Lightning Experience and the Salesforce mobile app.
- Each record type can have only one compact layout assigned to it. However, the same compact layout can be associated with multiple record types.

- Compact layouts aren't assigned to profiles or individual users. To display different sets of fields in records by use case or role, create record types for the object, then assign the appropriate custom compact layout to each record type.
- If a user doesn't have access to one of the fields that you assign to a compact layout, the next field on the layout is used.
- A compact layout must contain at least one field.
- Don't make the primary field a lookup field. Doing this could result in navigation issues in Lightning Experience and the Salesforce mobile app.
- Removing a field from a page layout doesn't remove it from the object's compact layout. The two layout types are independent.
- If you change a field on a compact layout to an unsupported type, the field is removed from the compact layout.
- Before you can delete a compact layout that's set as the primary compact layout for the object, you must choose another compact layout to replace it.
- In the Salesforce mobile app, tasks automatically show whether a task is open or closed and the due date (depending on a user's access to activity dates). When customizing a task compact layout, you don't have to add these fields to the Selected Fields list.
- Compact layouts installed from a managed package are editable. However, we recommend against editing compact layouts installed from a package, as doing so can cause destructive changes to your org. Instead, clone the installed compact layout and make your changes to the clone.
- Compact layout assignments are subscriber controlled. If a user installs a managed package that contains a compact layout and then changes the compact layout's assignment in their org, the compact layout's assignment isn't overridden when they later upgrade the package.
- These considerations apply to Chatter:
 - In the full Salesforce site, a compact layout determines which fields appear in the Chatter feed item that appears after a user creates a record with a quick action.
 - To avoid inadvertent sharing of information through the feed, the Task page layout determines the fields shown in the Chatter feed items for tasks created using a quick action.
 - Primary compact layouts determine which fields are shown in Chatter personal digest emails.
- When a field with an image larger than 50x50 pixels is placed as the first field on a compact layout, the image is cropped. In any other position, the image appears as expected. If the image is in a formula field, you can get around this issue by setting the height and width values of the image in the formula to 50 pixels each. For more information, see [IMAGE](#) in Salesforce Help.

Limitations

- A compact layout can only contain fields from its object, including a formula field that is a cross-object reference to another object.
- Fields that aren't available in SOAP API don't show up on compact layouts in the Salesforce mobile app.
- Compact layouts support all field types except:
 - text area
 - long text area
 - rich text area
 - multi-select picklist

See Also

- [Create Compact Layouts](#)
- [Assign Compact Layouts to Record Types](#)
- [Compact Layouts](#)
- [Customize Case Hovers in Lightning Experience](#)

Custom Tabs

Custom tabs let you display custom object data or other web content in Salesforce. When you add a custom tab to an app in Salesforce Classic, it appears as a tab. When you add a custom tab to an app in Lightning Experience, it appears as an item in the app's navigation bar and in the App Launcher.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Custom Object Tabs and Web Tabs available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Visualforce Tabs available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Lightning Page Tabs available in: **All Editions except Database.com**

USER PERMISSIONS NEEDED

To create and edit custom tabs:	Customize Application
---------------------------------	-----------------------

Custom tabs show custom object data or other web content embedded in the app. You can create any of these types of custom tabs.

- Custom Object Tabs: Custom object tabs (available only at an app level and not on subtab apps) show the data of your custom object. Custom object tabs look and function just like standard tabs.
- Web Tabs: Custom web tabs show any external web-based application or web page. You can design web tabs to include the sidebar or span the page without the sidebar.
- Visualforce Tabs: Visualforce tabs show data from a Visualforce page. Visualforce tabs look and function just like standard tabs.
- Lightning Component Tabs: Lightning component tabs make Lightning components available in the Salesforce mobile apps and in Lightning Experience. Lightning components aren't supported in Salesforce Classic.
- Lightning Page Tabs: Lightning page tabs let you add Lightning app pages to the Salesforce mobile app and Lightning Experience navigation bars.

In Salesforce Classic, Lightning page tabs don't appear on the All Tabs page when you click .

Lightning page tabs also don't appear in the Available Tabs list when you customize the tabs for your apps.

Subtab apps support only web tabs and Visualforce tabs.

Delegated administrators who can manage specified custom objects can also create and customize tabs for those custom objects.

In Lightning Experience, Lightning page tabs, Visualforce tabs, and Lightning component tabs have a fixed, friendly URL structure of /lightning/n/*customTabDevName*.

See Also

[Create Custom Apps for Salesforce Classic](#)

[Subtab Apps in Salesforce Classic](#)

Create Web Tabs

Build web tabs so that your users can use your web applications or other websites from within the application.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Custom Object Tabs and Web Tabs available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Visualforce Tabs available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Lightning Page Tabs available in: **All Editions except Database.com**

USER PERMISSIONS NEEDED

To create and edit custom tabs:

Customize Application

1. From Setup, enter *Tabs* in the Quick Find box, then select **Tabs**.
2. Click **New** in the Web Tabs related list.
3. Choose a layout for the new tab. The full page width spans across the entire page without the sidebar, while the column style allows users to view the sidebar.
4. Click **Next**.
5. Enter a label to display on the tab.
6. Click the Tab Style lookup icon to display the Tab Style Selector.

If a tab style is already in use, a number enclosed in brackets [] appears next to the tab style name. Hover your mouse over the style name to view the tabs that use the style. Click **Hide styles which are used on other tabs** to filter this list.

7. Click a tab style to select the color scheme and icon for the custom tab. Optionally, click **Create your**

- own style** on the Tab Style Selector dialog to create a custom tab style if your org has access to the Documents tab. To create your own tab style:
- Click the Color lookup icon to display the color selection dialog and click a color to select it.
 - Click **Insert an Image**, select the document folder, and select the image you want to use.
 - Select a file and click **OK**. The New Custom Tab wizard reappears.
- Change the content frame height if necessary.
 - Enter a description of the tab, if desired, and click **Next**.
 - Enter the URL or choose the custom s-control that you want to display in the tab. Optionally, copy and paste any merge fields for data that you want dynamically replaced in the link. Click **Preview Web Tab** to display your web tab.



Note Only User, organization, and API merge fields are supported for web tabs.

- For a URL, choose an encoding setting and click **Next**.
- Add the web tab to the appropriate profiles. Choose Default On, Default Off, or Tab Hidden to determine whether the custom tab is visible to users with that profile. You can change this setting later.
- Click **Next**.
- Specify the custom apps that should include the new tab.
- Select **Append tab to users' existing personal customizations** to apply the tab visibility settings to all users.
- Save the tab.

Create a Custom Object Tab

Give your users access to custom object data and records in an app. Custom object tabs are available only at the app level, not on subtab apps, and function like standard tabs.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Custom Object Tabs and Web Tabs available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Visualforce Tabs available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Lightning Page Tabs available in: all editions except **Database.com**

USER PERMISSIONS NEEDED

To create and edit custom tabs:

Customize Application

When you add a custom object tab to an app in Salesforce Classic, it appears as a tab. When you add a custom object tab to an app in Lightning Experience, it appears as an item in the app's navigation bar and in the App Launcher.

- From Setup, in the Quick Find box, enter *Tabs*, and then select **Tabs**.
- Click **New** in the Custom Object Tabs related list.
- Select the custom object to appear in the custom tab. If you haven't created the custom object, click **Create a new custom object now**.

The label of the new tab is the same as the plural version of the custom object label.

- To show Tab Style Selector, click the Tab Style lookup icon. If a tab style is already in use, a number enclosed in brackets [] appears next to the tab style name. To view the tabs that use that style, hover over the style name. To filter this list, click **Hide styles which are used on other tabs**.
- To select the color scheme and icon for your custom tab, select a tab style.

If your org has access to the Documents tab, click **Create your own style** on the Tab Style Selector dialog to create a custom tab style.

- To access the color selection dialog, click the Color lookup icon and select a color.
- Click **Insert an Image**, select the document folder, and select the image that you want to use. You can also click **Search in Documents**, enter a search term, and click **Go!** to find a document file name that includes your search term.

 **Note** This dialog only lists files in document folders that are under 20 KB and have the Externally Available checkbox selected in the document property settings. If the document used for the icon is later deleted, Salesforce replaces it with a default multicolor block icon ().

- Select a file, and click **OK**. The New Custom Tab wizard reappears.
- You have the option to choose a custom link to use as the introductory splash page when users initially click the tab.
- Enter a description of the tab, if desired, and click **Next**.
- Choose the user profiles that you want to have access to the new custom tab.

For Professional Edition users and Salesforce Platform One license users, tab visibility is set to Default On.

 **Note** A custom object is searchable with or without a custom tab. Go to [Search Manager](#) to make your custom object searchable in your org.

- Specify the custom apps that you want to include in the new tCRMab.
- To add the tab to your users' customized display settings if they've customized their personal display, select **Append tab to users' existing personal customizations**.

 **Note** When this option isn't selected, you can still see the tab in the users' customized display settings in Lightning Experience, but it's not visible in Salesforce Classic.

- Save the tab.
Depending on the visibility settings you selected, you see the tab right away.

The screenshot shows the Salesforce Lightning Experience interface. At the top, there's a navigation bar with tabs like Home, Opportunities, Leads, Tasks, Files, Properties (which is currently selected), Notes, and More. Below the navigation bar is a search bar labeled 'Search Properties and more...'. Under the 'Properties' tab, there's a section titled 'Recently Viewed' with a dropdown arrow. It lists four items: '9083 17th Street West', '769 Olive Court', '2345 Main Street', and '2047 Chestnut Ave.'. Each item has a small checkbox icon to its left.

See Also

[Custom Tabs](#)
[Make Search Faster](#)

Create Lightning Page Tabs

Before you can include a Lightning app page in the Salesforce mobile app or in a Lightning app, you must create a custom tab for it.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

USER PERMISSIONS NEEDED

To create and edit custom tabs:	Customize Application
---------------------------------	-----------------------

When you first activate an app page in the Lightning App Builder, a tab is created for the page as part of the activation process. You can also create a tab for the page manually in Setup before you activate it.

You can create a custom tab only for an App Page type of Lightning page.

1. From Setup, enter *Tabs* in the Quick Find box, then select **Tabs**.
2. Click **New** in the Lightning Page Tabs related list.
3. Choose a Lightning page for the tab.
4. Enter a label.
This text is used as the display name for the Lightning page.
5. Select a tab style to set a color scheme and icon for the Lightning page tab.
6. Enter a description of the tab, if desired, and click **Next**.
7. Choose the user profiles the new custom tab is available for. In Salesforce Classic, the Default On and Default Off options for Lightning page tabs don't work the same way as for other custom tabs. The Lightning page menu item appears for the selected profiles in Salesforce for Android and Salesforce

for iOS whether you choose Default On or Default Off. Select the **Tab Hidden** option to hide the Lightning page for the selected profiles.

8. Click **Save**.

When creating a custom icon for your Lightning page tab, follow these image guidelines. The icon should:

- Be less than 10k in size
- Be 120 x 120 pixels
- Be a PNG with a transparent background
- Have a resolution of 72 dpi
- Not include a color background
- Not include outer shadows on the inner icon graphic

Custom Tab Considerations

When working with custom tabs, keep these considerations in mind.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Custom Object Tabs and Web Tabs available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Visualforce Tabs available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Lightning Page Tabs available in: **All Editions except Database.com**

- The custom tabs limit is a fixed number based on edition and can't be increased. For more information, contact Salesforce.
- If you choose Default On or Default Off when setting custom object tab visibility, an option is added to the Create New dropdown list in the Salesforce Classic sidebar so that users with the Create permission can quickly create a record. For example, if the custom object displayed in the custom tab is named Expenses, an Expense option appears in this list.
- A custom object that's associated with a custom tab is searchable by default, even if users don't add the tab for display.
- A Web tab or custom link could display a blank page if the embedded site:
 - Has been set to deny the loading of its content in a frame.
 - Has been set to allow the loading of its content in a frame only if the same site is delivering the content.
 - Contains a mix of secure and unsecure content, and the user's browser has been configured to block mixed active content.

To resolve this issue, try these workarounds.

- Set your custom link to either open in a new window or display in the existing window without the sidebar or header.
- Move the URL from a Web tab into a custom link instead, and set the URL to either open in a new window or display in the existing window without the sidebar or header.
- If the site you're embedding has an HTTP prefix and mixed active content, try changing the prefix to HTTPS. If the embedded site has a valid security certificate and it hasn't blocked itself from being displayed in frames, using HTTPS as the prefix allows the site to display.
- In Salesforce Classic, Lightning page tabs don't display on the All Tabs page when you click . Lightning page tabs also don't appear in the Available Tabs list when you customize the tabs for your apps.
- In Salesforce Classic, the Default On and Default Off options for Lightning page tabs don't work the same way as for other custom tabs. The Lightning page menu item appears for the selected profiles in Salesforce for Android and Salesforce for iOS whether you choose Default On or Default Off. Select the **Tab Hidden** option to hide the Lightning page for the selected profiles.
- If the Home page tab isn't visible for a user in Salesforce Classic or Lightning Experience even after the user profile's tab setting is set to Default On, disable the **Enhanced Profile User Interface** option in the User Management Settings. If the issue persists, enable the **Overwrite users' personal tab customizations** in the tab settings for the user profile.

Custom Help Content

Tailor help so that users understand how to work within your unique implementation of Salesforce. You can add learning in the flow of work in several ways. Add custom help for a page, app, object, or org level, or provide links to help in the Utility Bar, Path, or as a text box on the page. Show users custom resources in a Lightning Experience welcome mat when they first log in. Or, to reach users with important news, training, and on-boarding information, add micro-learning prompts and walkthroughs to your app.

Help	Salesforce Classic	Lightning Experience
Field-Level Help —Detail the purpose and function of a standard or custom field.	✓	✓
Object-Level Help in Salesforce Classic —Help your users by providing object-level help for all custom objects and external objects.	✓	See Learning Paths
Learning Paths —Create a personalized and always-available learning experience for your users.		✓
Customize Content in the Guidance Center —Direct users to		✓

Help	Salesforce Classic	Lightning Experience
a globally available menu of suggested resources from Salesforce or links to custom help assigned with Learning Paths.		
Customize the Help Menu in Lightning Experience —Supplement the recommended resources by adding a section with links to your own content.		✓
In-App Guidance —Use prompts to reach users directly with news, training, and onboarding messages.		✓
Utility Bar —Use the Notes component to add links to custom help.		✓
Rich Text Component —Add simple HTML to your Lightning page.		✓
Guidance for Success on Path —Give key coaching details to support users.		✓
In-Dashboard Videos for Einstein Analytics —Provide customized instruction that helps users get the most out of dashboards.	Not applicable - cloud-based platform	Not applicable - cloud-based platform
Custom Notification from a Process —Alert an account owner if a new support case is logged while trying to close a deal, or send a notification for a workflow built entirely with custom objects.	✓	✓
Sales Enablement (myTrailhead) —Your company's unique enablement content in a branded experience powered by	Not applicable	Not applicable

Help	Salesforce Classic	Lightning Experience
the Trailhead online learning platform.		

Field-Level Help

Field-level help lets you provide help text detailing the purpose and function of any standard or custom field. Before defining field-level help, review these implementation tips and best practices.

Define Field-Level Help

Define custom help text for your organization's fields to provide users with a helpful description. Help text can be defined for standard and custom fields on all detail and edit pages where the field displays. Users can view the field-level help text by hovering over the Info icon next to the field.

Custom Help in Lightning Experience

In Lightning Experience, In-App Guidance allows you to add prompts and walkthroughs to reach users directly with feature updates, record help, and onboarding tips. Find help from anywhere in the app by opening the Guidance Center, which shows learning items assigned with Learning Paths and suggestions from Salesforce. All Trailhead modules and custom help assignments can be found on Learning Home.

Custom Help in Salesforce Classic

In Salesforce Classic, object-level help replaces the links for a custom object or external object page. Replace built-in Salesforce Help with documentation that's customized for your users.

Field-Level Help

Field-level help lets you provide help text detailing the purpose and function of any standard or custom field. Before defining field-level help, review these implementation tips and best practices.

Implementation Tips

- Field-level help is enabled by default for all editions.
- Field-level help isn't available for some standard fields, including fields on the User object, system read-only fields, auto-number fields, multi-currency fields, Ideas fields, and Experience Cloud site fields.
- The help text for a field is automatically added to a package when you add the associated field to any AppExchange package.
- In a managed package, the help text is locked to the developer, giving installers full capabilities to change it.
- Field-level help isn't available for Visualforce pages in Lightning Experience.
- Field-level help is available for Visualforce pages in Salesforce Classic only if the page's `showHeader` attribute is set to `true`.
- Field-level help on dependent fields appears in Salesforce Classic but not in Lightning Experience.

Best Practices

- Because your custom help text displays on both edit and detail pages, avoid instructions for entering data. Instead, construct help text that defines the field's purpose, such as:
The maximum discount allowed for this account.
- Provide information in your help text about the attributes of the field, such as:
A detailed description of the purpose for the expense report. Up to 32 KB of data are allowed. Only the first 255 characters display in reports.
- Provide examples in your help text that help users understand the field's meaning clearly, such as:
The four-digit promotional code used to determine the amount charged to the customer, for example, 4PLT (for level-four platinum pricing).
- If your org uses more than one language, provide translations for your Help Text using the Translation Workbench.

See Also

[Custom Help Content](#)

Define Field-Level Help

Define custom help text for your organization's fields to provide users with a helpful description. Help text can be defined for standard and custom fields on all detail and edit pages where the field displays. Users can view the field-level help text by hovering over the Info icon next to the field.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: All Editions except Database.com

USER PERMISSIONS NEEDED

To define or change field-level help: Customize Application

- From the management settings for the field's object, go to Fields.
- Click **Edit** next to the field.
- In the **Help Text** field, enter the text you want displayed.
The text displays when a user hovers the mouse over the Info icon that appears next to the field on a detail or edit page. You can enter up to 510 characters.
- Click **Save**.

 **Note** If the object you're modifying is exposed in an Experience Cloud site, field-level help is visible to site members, including unlicensed users, partners, and customers. Make sure the information you provide in field-level help accounts for all audiences and doesn't contain business-sensitive information.

See Also

[Custom Help Content](#)
[Field-Level Help](#)
[Find Object Management Settings](#)

Custom Help in Lightning Experience

In Lightning Experience, In-App Guidance allows you to add prompts and walkthroughs to reach users directly with feature updates, record help, and onboarding tips. Find help from anywhere in the app by opening the Guidance Center, which shows learning items assigned with Learning Paths and suggestions from Salesforce. All Trailhead modules and custom help assignments can be found on Learning Home.

Understand the Benefits

Easily accessible resources within the app are crucial to help users do their jobs efficiently and effectively inside their workspace. Learn more about the onboarding and troubleshooting stages of the in-app journey by taking the [User Engagement](#) and the [User Training and Enablement](#) Trailhead modules. Without training, end-user productivity declines, as can trust and feature adoption. With sufficient training and on-demand troubleshooting resources, users quickly find answers to their questions so they can get unblocked and back to work. These resources are also a time saver for admins because they can prevent them from repeatedly fielding the same questions from their users.

Because apps are so often customized, sometimes Salesforce resources don't reflect the actual user experience. As a result, many admins help their users by adding links to resources that reflect the company's unique processes and guidelines. There are several ways to integrate custom help inside the app, but the Guidance Center, Learning Paths, and In-App Guidance are great additions to your admin toolbox.

 **Important** Learning Paths is turned off by default in all environments in Winter '25 and later. To turn on Learning Paths, contact Salesforce Customer Support. Learning Paths isn't available in a sandbox or in Government Cloud Plus environments.

Decide When to Use the Guidance Center or In-App Guidance

If you...	Then consider...
<ul style="list-style-type: none">• Have important resources to share with your users.• Require users to complete training by a certain date.• Have training created for specific pages or specific users.	<p>Guidance Center and Learning Paths assignments</p> <p>Manage, assign, and track all your longform in-depth learning resources.</p> <p>Display public Trailhead and enablement site (myTrailhead) modules or URLs to</p>

If you...	Then consider...
	<p>resources in specific locations for specific users in the Guidance Center.</p> <p>Apply a due date.</p> <p>Centralize all learning resources on the Learning homepage.</p> <p>Earn Trailhead and enablement site badges inside the app.</p>
Want to nudge, prompt, or remind users of important announcements or features.	<p>In-App Guidance</p> <p>Provide targeted, micro-learning moments to boost feature adoption.</p> <p>Pick from targeted, floating, and docked prompts or walkthroughs.</p> <p>Design the guidance, select the target audience, and specify where it appears and for how long.</p> <p>Add links to internal wikis, training, or PDFs for a custom call to action.</p> <p>Use built-in engagement tracking.</p>

Decide When to Use the Guidance Center or Help Menu

Guidance Center and the Help Menu are accessible from the global header. The Help Menu offers a section for admins to add links to their own content that their users can see anywhere in the app. Guidance Center customized with Learning Paths offer admins not only a panel to add contextual help per page, but also a way to assign required learning to users. Users see all their assignments on Learning Home.

This table lists the main differences between Guidance Center customized with Learning Paths and the Help Menu.

Feature	Guidance Center and Learning Paths	Help Menu

Hide completely	No, you can't hide the Guidance Center.	No, you can't hide the Help Menu.
Hide individual sections	Yes, hide suggestions from Salesforce or Learning Home from the Guidance Center Setup page.	Yes, from the Help Menu Setup page
Add URLs as learning items	Add up to 1,000 learning items with Learning Paths, and up to four learning items appear in the Guidance Center at a time.	Add up to 30 URLs
Add help for a specific page, app, or org	Up to two resources for the object home or record appear under Help for This Page. The Selected for You section includes up to two resources assigned to all objects for all apps or for a specific app with a due date.	Resources for an org appear in the global custom resource section at the top of the Help Menu. Admins can't specify per page or per app.
Specify a required due date to complete	Yes	No
Specify user-level visibility	Yes, either specify by users or by public groups	No
View Trailhead modules inside the app	Yes, users can read and earn badges for multiple choice challenges inside the app.	No.
View suggestions from Salesforce	Salesforce suggestions appear under Selected for You and Related to This Page.	Under Resources, there's links to view keyboard shortcuts, go to Trailhead, go to Salesforce Help, and view Release Notes (admins only).

In-App Guidance in Lightning Experience

Help users discover your products and services, adopt your processes, or learn how to use a new feature. Create prompts and walkthroughs in Lightning Experience apps or pages.

Learning Paths

Create a personalized and always-available learning experience for your users. When you customize Learning Paths, you can choose whom to assign a learning item to—individuals, public groups, or all users. The Guidance Center icon in the global header opens a panel where users can see suggestions from Salesforce and your Learning Paths assignments. Associate the learning item with a particular app or page, and optionally apply a due date. Choose modules from Trailhead or an enablement site (myTrailhead), and add links to other training resources for the ultimate custom training experience. Admins can let other stakeholders, such as trainers or sales team managers, assign and manage

learning items.

Guidance Center

The Guidance Center panel in Lightning Experience provides a convenient location for showing content that's assigned to or recommended for the current user. The Guidance Center includes not only suggested content that's authored by Salesforce but also custom content that you manage. With Learning Paths, you can manage the learning items—custom links, videos, and Trailhead modules—that are available in the Guidance Center and assign those items to specific users with optional due dates.

Customize the Help Menu in Lightning Experience

The question mark icon in the global header opens a menu of Salesforce resources. Supplement these links by adding a global section with links to your own content. There's only one custom Help Menu section per org and it appears at the top of the Help Menu.

In-App Guidance in Lightning Experience

Help users discover your products and services, adopt your processes, or learn how to use a new feature. Create prompts and walkthroughs in Lightning Experience apps or pages.

See [In-App Guidance](#).

Learning Paths

Create a personalized and always-available learning experience for your users. When you customize Learning Paths, you can choose whom to assign a learning item to—individuals, public groups, or all users. The Guidance Center icon in the global header opens a panel where users can see suggestions from Salesforce and your Learning Paths assignments. Associate the learning item with a particular app or page, and optionally apply a due date. Choose modules from Trailhead or an enablement site (myTrailhead), and add links to other training resources for the ultimate custom training experience. Admins can let other stakeholders, such as trainers or sales team managers, assign and manage learning items.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Professional, Enterprise, Performance, and Unlimited** Editions

Not Available for: **Chatter External, Chatter Free, or Communities** (now known as Experience Cloud) user licenses

Enablement Sites (myTrailhead) available in: **Enterprise, Performance, and Unlimited** Editions

! **Important** Learning Paths is turned off by default in all environments in Winter '25 and later. To turn on Learning Paths, contact Salesforce Customer Support. Learning Paths isn't available in a sandbox or in Government Cloud Plus environments.

- Get going right away with [Assign a Learning Item for Learning Paths](#).

- Understand the layout of the Guidance Center with [Explore the Guidance Center](#) in Salesforce Help. Learn who sees what type of content.
- Learn more on Trailhead, and take the [User Training and Enablement](#) module.



Note As long as your Salesforce-related accounts are part of your Trailblazer.me profile, badges that you earn while inside Lightning Experience are included in your profile. See [Trailblazer.me and Trailhead](#). If your company has an enablement site, you don't necessarily have a Trailblazer.me profile, but you can create one. See [Create a Trailhead Account](#), and log in with the username and password that you use to log in to Salesforce. If your company uses Salesforce Identity for Enablement to authenticate users to your enablement site, the Salesforce Trailhead badges that you earn in-app appear on your Trailblazer.me profile. The enablement site badges that you earn don't appear on your Trailblazer.me profile. See [Considerations for Selecting an Authentication Provider](#)

[Learning Paths Best Practices](#)

Learn how to define an in-app help strategy, and get tips for customizing Learning Paths.

[Learning Paths Considerations and Limitations](#)

As you prepare to customize Learning Paths for your users, keep these details in mind.

[Assign a Learning Item for Learning Paths](#)

You can assign a Trailhead or enablement site (myTrailhead) module or a custom learning item such as a video, tutorial, or reference guide. You can assign a learning item to individuals, public groups, or all users. And you can associate a learning item with an app and a particular object and page within the app.

[Monitor Learning Paths \(Beta\)](#)

Use custom reports and dashboards to track user engagement with Learning Paths, Trailhead, and enablement sites (myTrailhead).

[Turn Off Learning Paths](#)

Optionally, you can turn off Learning Paths, which hides the Learning Home and prevents admins from assigning learning items to users. When Learning Paths is turned off, the Guidance Center remains available.

See Also

[Customize Content in the Guidance Center](#)

Learning Paths Best Practices

Learn how to define an in-app help strategy, and get tips for customizing Learning Paths.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Professional, Enterprise, Performance, and Unlimited** Editions

Not Available for: **Chatter External, Chatter Free, or Communities** (now known as Experience Cloud) user licenses

Enablement Sites (myTrailhead) available in: **Enterprise, Performance, and Unlimited Editions**

- !** **Important** Learning Paths is turned off by default in all environments in Winter '25 and later. To turn on Learning Paths, contact Salesforce Customer Support. Learning Paths isn't available in a sandbox or in Government Cloud Plus environments.

Strategy

As you plan your in-app help strategy, think of adding layers of customization to address the needs of your users and, if applicable, your existing training program. Along with creating in-app guidance, customizing Learning Paths is a great next step.

Learning Paths offers targeted assignment functionality, available now and in upcoming releases. It's as easy to assign learning items across Salesforce as it is to assign items to a specific page for specific users. Users can read assigned Trailhead modules and earn badges right inside the app.

Be sure to let users know that they can access these resources from the Trailhead icon in the global header or from Learning Home. To make Learning Home easier to find, add Learning to the navigation bar.

Tips to Get Started

Here are some ideas to help you plan which resources to add to Learning Paths.

- Replicate the modules that you added to the global custom section of the Help Menu.
- Add your company's videos, wiki pages, quick start guides, enablement site (myTrailhead) modules, or other custom resource.
- Assign shorter modules that users can complete in less than a half hour to facilitate learning as the user works.
- Assign only one or two items to a specific page in the app.
- Add due dates to give users a better sense of when to complete the learning item.
- Scope out what to add by listing the learning item, location, audience, and, optionally, due date.

Here are some resources to consider adding. All learning items are Trailhead modules, except as noted.

Onboarding

Learning Item	App
Let's Take Lightning Experience for a Spin (video)	All
Searching in Salesforce (video)	All
Working with List Views (video)	All
Trailhead: Quick Look	All

Learning Item	App
Salesforce User Basics	All
Salesforce User Tour	All
Salesforce CRM (CRM for Lightning Experience)	All

Working from Anywhere

Learning Item	App
Virtual Meeting Setup: Quick Look	All
Virtual Collaboration	All
Virtual Whiteboarding: Quick Look	All
Effective Emails: Quick Look	All
Relationship Building	All
Emotional Intelligence	All
Fearless Teaming	All

Sales

Learning Item	App
Sales Cloud: Quick Look	Sales
Prospecting to Improve Sales	Sales
Demo Delivery Essentials	Sales
Demo Storytelling	Sales
Design Thinking for Sales	Sales

Service

Learning Item	App
Service Cloud: Quick Look	Service
Service Cloud Agent Experience	Service Console
Communication Skills for Customer Service Agents	Service

Wellbeing

Learning Item	App
The Power of Movement	All
The Value of Sleep	All
Mindful Living with the Plum Village Monastics	All

See Also

[Custom Help in Lightning Experience](#)

Learning Paths Considerations and Limitations

As you prepare to customize Learning Paths for your users, keep these details in mind.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Professional, Enterprise, Performance, and Unlimited** Editions

Not Available for: **Chatter External, Chatter Free, or Communities** (now known as Experience Cloud) user licenses

Enablement Sites (myTrailhead) available in: **Enterprise, Performance, and Unlimited** Editions

 **Important** Learning Paths is turned off by default in all environments in Winter '25 and later. To turn on Learning Paths, contact Salesforce Customer Support. Learning Paths isn't available in a sandbox or in Government Cloud Plus environments.

Viewing Assignments in the Guidance Center

Not all assignments appear in the Guidance Center. Users can find all learning assignments on Learning Home.

Limitations for Assignments

There are limits to the assignments that you can make.

Assignments	Maximum
Learning items	500
Assignments per learning item	100

-  **Note** The number of assignments doesn't always correspond to the number of users assigned to the learning item. A public group can contain thousands of users, but assigning a learning item to that public group still counts as only one assignment.

When you assign a learning item to a particular app or page, the item is visible in the side panel only to users who are in that app. It's visible on Learning Home to all users, regardless of which app a user is in.

Group Assignments

To assign a learning item to a group, use a public group. An assignment to a public group counts as one assignment.

You can't assign learning items to Roles and Subordinates or Roles and Internal Subordinates public group member types. You can't assign learning items to permission set groups, personal groups, or Chatter groups.

Learning Items with Trailhead or Enablement Site Modules

You can't create more than one learning item with the same Trailhead or enablement site module.

You can assign Trailhead modules that contain quizzes, but not modules that contain hands-on challenges. You can't assign Trailhead projects.

When you assign a Trailhead or enablement site module, the module assignment appears in the Guidance Center and on Learning Home. It doesn't appear on an assignee's enablement site home page.

-  **Tip** When a user completes an assigned module, the assignment status for the learning item doesn't always update right away. Sometimes, the module can show 100% completion but remains available in the Guidance Center and doesn't move to the Completed tab in Learning Home. To see updated assignment status for a completed module, users can try refreshing Learning Home, refreshing a page where the Guidance Center is open, clearing their browser cache, or logging in with a different browser.

Modifying Assigned Learning Items

After you save a learning item, you can't change its module, custom link, or due date. Delete the learning item and create another one with the module, link, or due date that you want to assign. To add more assignees to a learning item, use the Add Assignments feature on Learning Home.

If a module is deleted from Trailhead or an enablement site, any learning item that you created with that module remains in Learning Paths. To avoid confusing users, delete the learning item.

Deleting Learning Items

If you want to delete a learning item that's assigned to more than 100,000 users, first unassign users before you delete the learning item. Otherwise, errors can occur when trying to delete a learning item that's assigned to a large number of users.

Turning Off Learning Paths and Suggested Content

See [Turn Off Learning Paths](#) and [Turn Off Suggested Content in the Guidance Center](#).

To hide assigned content from users, delete the assignments or select **Don't assign to anyone**.

See Also

- [Assign a Learning Item for Learning Paths](#)
- [Learning Paths Best Practices](#)
- [Monitor Learning Paths \(Beta\)](#)
- [Turn Off Learning Paths](#)
- [Turn Off Suggested Content in the Guidance Center](#)

Assign a Learning Item for Learning Paths

You can assign a Trailhead or enablement site (myTrailhead) module or a custom learning item such as a video, tutorial, or reference guide. You can assign a learning item to individuals, public groups, or all users. And you can associate a learning item with an app and a particular object and page within the app.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Professional, Enterprise, Performance, and Unlimited** Editions

Not Available for: **Chatter External, Chatter Free, or Communities** (now known as Experience Cloud) user licenses

Enablement Sites (myTrailhead) available in: **Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To manage as an admin	Modify All Data OR Customize Application
To manage as a designated trainer	Manage Learning

! **Important** Learning Paths is turned off by default in all environments in Winter '25 and later. To turn on Learning Paths, contact Salesforce Customer Support. Learning Paths isn't available in a sandbox or in Government Cloud Plus environments.

To see learning items in action, check out this video.

Watch the video: <https://play.vidyard.com/aRhN1xHQXC2aW9PFotU4H7>

1. To open the panel, click the icon for Guidance Center in the global header.



2. At the bottom of the panel, click **Assign Learning Content** to go to Learning Home.

You can also get to Learning Home via the App Launcher. Add Learning Home to your navigation bar for easy access.

3. On Learning Home, click **Manage Learning Assignments**, then click **New**. Only admins or users with the Manage Learning user permission have access to the Manage Learning Assignments page.
4. In the New learning item window, select a content type, then search for a Trailhead or enablement site module, or enter the URL for a custom link.

The screenshot shows the 'New learning item' window. At the top, it says 'New learning item'. Below that, there's a section titled 'Content Type' with a note: 'Your type selection determines the icon displayed next to the learning item and the open behavior when the user clicks the item.' Two radio buttons are shown: 'Trailhead Module' (selected) and 'Link'. A horizontal line separates this from the 'Module Details' section. Under 'Module Details', it says: 'Trailhead modules with hands-on challenges and modules in an existing learning item are excluded from the search results.' Below this, a search bar contains 'Campaign Basics'. A result card for 'Campaign Basics' is shown, featuring a blue circular icon with a white star, the text 'Trailhead Module', 'Campaign Basics', '+400 Points', and '~55 mins'.

When users click the link for a Trailhead or enablement site module in the side panel or on Learning Home, the item opens in the app. The progress that's shown for the learning item depends on the progress that users make in the module.

When users click a custom link, the link opens in a new browser tab and the learning item is marked as complete immediately. If you previously included custom links in the Help Menu, add those links with Learning Paths instead.

5. Under Location, specify where the learning item appears in the app by selecting an app and, if you want, an object and record.

The screenshot shows a 'Location' section with two dropdown menus. The first menu, labeled '* App', has 'All' selected. The second menu, labeled '* Objects and Records', also has 'All' selected. A note above the menus states: 'Apps and objects that you don't have access to are hidden.'

- To make the learning item appear across Salesforce, select **All** for App and Object and Record.
 - To make the learning item show up on every page in an app, select an app in the App picklist, then select **All** in the Object and Record picklist.
 - To make the learning item show up for a specific object in an app, select a value in the App and the Object and Record picklists.
 - To have the learning item show up for the same object across Salesforce, select **All** in the app picklist, then select a value in the Object and Record picklist.
 - To add a module to a specific Setup page, enter a URL. If you don't specify a URL, the module shows up on all pages in Setup.
6. Under Assignments, specify which users and public user groups see the module. If you don't assign the module to at least one user or public group, it doesn't show up in the app.

The screenshot shows an 'Assignments' section with three radio button options: 'Assign to all users', 'Don't assign to anyone', and 'Assign to specific users or groups'. The third option is selected. Below the radio buttons is a search bar with the placeholder 'Search users...' and a magnifying glass icon.

When you assign a learning item to a public group, the assignment also applies to members that you subsequently add to the group. The assignment is removed for members that you remove from the group.

7. To require users to complete the learning item, include a due date.

Due Date

To make a learning item required for assigned users, set a due date. If you don't assign the learning item to at least one user or group, you can't add a due date.

No due date
 Due in 7 days
 Due in 30 days
 Custom due date

* Date 

8. Save your changes.

- The Selected for You section in the Guidance Center displays up to two learning items with a due date and no location specified for Objects and Records. When completed, the learning items are removed from the panel.
- The Related to This Page section in the Guidance Center displays up to five learning items with a location specified for Objects and Records. An option to view more items in this section, up to 100, is also available. When completed, the learning items remain.
- A learning item that's assigned to an app or a page is visible in the side panel only when users are in the associated app.
- Users can see all assigned learning items on Learning Home.

9. On the Manage Learning Assignments page, you can expand a row in the Name column to see all the users and groups that the learning item is assigned to.

  Campaign Basics
 Eastern Sales Team
 Admin User
 Ashley James

Use the row-level actions for a learning item to edit, delete, or add assignees to the item.

- To change the page that a learning item is assigned to, select **Edit** from the actions menu for the learning item.
- To assign a learning item to another individual or group, select **Add Assignments** from the actions menu for the learning item.
- To remove all assignments, select **Unassign All** in the row-level actions menu for the learning item.
- To delete a particular assignment, select **Remove Assignment** from the row-level action menu for an individual or group.

We suggest that you let your users know about Guidance Center and Learning Home so they can watch out for required content. To hide Learning Home, in Setup, turn off **Learning Paths** on the Guidance Center page.

See Also

[Learning Paths Considerations and Limitations](#)

- [Learning Paths Best Practices](#)
- [Customize the Help Menu in Lightning Experience](#)
- [Customize Content in the Guidance Center](#)
- [Plan Your Enablement Site Solution](#)
- [Your Trailblazer Profile](#)
- [Merge Trailblazer Accounts](#)
- [Personalize the Navigation Bar in Lightning Experience](#)
- [Personalized Navigation Considerations](#)
- [Add and Customize Tabs on Lightning Pages Using the Lightning App Builder](#)
- [What Is a Group?](#)
- [Create and Edit Groups](#)

Monitor Learning Paths (Beta)

Use custom reports and dashboards to track user engagement with Learning Paths, Trailhead, and enablement sites (myTrailhead).

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Professional, Enterprise, Performance, and Unlimited** Editions

Not Available for: **Chatter External, Chatter Free, or Communities** (now known as Experience Cloud) user licenses

Enablement Sites (myTrailhead) available in: **Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To create custom reports and dashboards for Learning Paths Manage Learning Reporting

To create and update custom report types **Legacy Folder Sharing**
Create and Customize Reports

AND

Manage Custom Report Types

Enhanced Folder Sharing
Create and Customize Reports

AND

Manage Custom Report Types

USER PERMISSIONS NEEDED

To delete custom report types

Legacy Folder Sharing

Create and Customize Reports

AND

Manage Custom Report Types

AND

Modify All Data

Enhanced Folder Sharing

Create and Customize Reports

AND

Manage Custom Report Types

AND

Modify All Data

 **Important** Learning Paths is turned off by default in all environments in Winter '25 and later. To turn on Learning Paths, contact Salesforce Customer Support. Learning Paths isn't available in a sandbox or in Government Cloud Plus environments.

 **Note** As a beta service, Monitoring Learning Paths is subject to the Beta Services terms at: <https://www.salesforce.com/company/legal/agreements.jsp>. Use this feature at your sole discretion, and make your purchase decisions only on the basis of generally available products and features. Salesforce doesn't guarantee general availability of this feature within any particular time frame or at all, and we can discontinue it at any time. This feature is for evaluation purposes only, not for production use. It's offered as is and isn't supported, and Salesforce has no liability for any harm or damage arising out of or in connection with it. All restrictions, Salesforce reservation of rights, obligations concerning the Services, and terms for related Non-Salesforce Applications and Content apply equally to your use of this feature.

Use Cases for Reporting on Learning Paths

Learning Paths comes with four objects. Two of them give you information about Trailhead and an enablement site, and two give you information about your assignments in Learning Paths. You can:

- Track the number of Trailhead and enablement site modules completed, including total points earned and how long it took each user to complete each module.
- Identify the modules that are the most popular with your users by tracking how many people

complete each one.

- Track the number of assigned modules that your users started but haven't completed to get an idea of how helpful that learning item is.
- Track how many learning items you assign in a given month and how many are in progress, overdue, or completed.
- Filter your reports by user fields such as username, profile, role, manager, or locale.

Here are examples of how you can use the four objects to track your users' progress on learning items. Set up a custom report type using one of the following objects to monitor Learning Paths.

Object	Pulls Data from...	Create a Report Showing...
Learning Content	Trailhead and an enablement site	How many Trailhead or enablement site modules exist, by module title, and how many points a user can earn for each
Learning Content Progress	Trailhead and an enablement site	How many Trailhead and enablement site modules your users completed Which users completed a particular Trailhead or enablement site module and how long it took them to finish How many enablement site modules are in progress How many total points earned company-wide
Learning	Learning Paths	How many individual users you assigned a learning item to and how many of those users completed the item
Learning Assignment Progress	Learning Paths	Which users you assigned a learning item to, which users have it in progress, and which users completed it Which users have an overdue learning assignment The percentage of your users that completed an assigned

Object	Pulls Data from...	Create a Report Showing...
		learning item

For example, create a report that associates the Learning Content object with the Learning object and Learning Assignments Progress object. Then you can review data on the modules that you assign to specific users and those users' progress on the assigned modules.

Custom Report Type
In-App Learning Items
[« Back to List: Users](#)

Below is the information for this custom report type. You can click the buttons on this page to preview or update information for the custom report type.

Custom Report Type Definition	
Report Type Label	In-App Learning Items
Report Type Name	In_App_Learning_Items
Description	How many users you've assigned a learning item to, and how many of those users have completed the item.
Created By	Admin User, 3/26/2021, 1:52 PM

Object Relationships [Edit](#)

LearningContents (A)

- with or without related records from **Learning (B)**
- with or without related records from **Learning Assignments Progresses (C)**

After you create reports, you can create a dashboard that shows key data.

Dashboard
Learning In-App with Trailhead Overview
As of Mar 26, 2021, 2:36 PM Viewing as Admin User

<p>Badges Completion Trends</p>  <p>Record Count</p> <p>Completed Date</p> <p>View Report (Badges Completion Trends)</p>	<p>Most Popular Badges</p> <table border="1"> <thead> <tr> <th>Content Title</th> <th>Is Comp...</th> <th>Content Duration ...</th> <th>Earned Point ...</th> </tr> </thead> <tbody> <tr> <td>CRM for Lightning Experience</td> <td>✓</td> <td>40</td> <td>75</td> </tr> <tr> <td>Trailhead Basics</td> <td>✓</td> <td>15</td> <td>200</td> </tr> <tr> <td>Action Plans in Financial Services Cloud</td> <td>✓</td> <td>65</td> <td>400</td> </tr> <tr> <td>Strategies for Big Data Architecture</td> <td>✓</td> <td>30</td> <td>300</td> </tr> <tr> <td>Customer 360 Guide: Quick Look</td> <td>✓</td> <td>5</td> <td>25</td> </tr> <tr> <td>Brighten Your Day: Trailhead Wk1*</td> <td>✗</td> <td>n/a</td> <td>n/a</td> </tr> </tbody> </table> <p>View Report (Most Popular Badges)</p>	Content Title	Is Comp...	Content Duration ...	Earned Point ...	CRM for Lightning Experience	✓	40	75	Trailhead Basics	✓	15	200	Action Plans in Financial Services Cloud	✓	65	400	Strategies for Big Data Architecture	✓	30	300	Customer 360 Guide: Quick Look	✓	5	25	Brighten Your Day: Trailhead Wk1*	✗	n/a	n/a	<p>Total Badges Completed</p> <p>Nailed it.</p> <p>50</p> <p>Want to see the report? Check it.</p> <p>View Report (Total Badges Completed)</p>
Content Title	Is Comp...	Content Duration ...	Earned Point ...																											
CRM for Lightning Experience	✓	40	75																											
Trailhead Basics	✓	15	200																											
Action Plans in Financial Services Cloud	✓	65	400																											
Strategies for Big Data Architecture	✓	30	300																											
Customer 360 Guide: Quick Look	✓	5	25																											
Brighten Your Day: Trailhead Wk1*	✗	n/a	n/a																											
<p>Total Time Learning Entire Org</p> <p>125K</p> <p>Time = \$2E 4f</p> <p>View Report (Most Popular Badges)</p>	<p>Total Points Earned Entire Org</p> <p>881K</p> <p>That's the point!</p> <p>View Report (Most Popular Badges)</p>	<p>Total Badges In Progress (Not Fi...</p> <p>Behold, Trailblazers are on the trail.</p> <p>50</p> <p>Want to see the report? It's here.</p> <p>View Report (Total Badges In Progress)</p>																												

- Number of Trailhead and enablement site modules completed company-wide, by week (1)
- Most popular Trailhead and enablement site modules (2)
- Number of Trailhead and enablement site modules completed company-wide (3)
- Number of hours spent reading Trailhead and enablement site modules company-wide (4)
- Number of points earned company-wide (5)
- Number of Trailhead and enablement site modules that your users have started but not yet finished (6)

Considerations for Monitoring Learning Paths

You can track user engagement with Trailhead and enablement site modules, but not with custom links.

Reports using the Learning Content and Learning Content Progress objects show a maximum of 1,000 rows.

You can filter a report on the Learning Content Progress object by the date when a user began a module, but not when a user completed the module.

You can't filter a report on the Learning Assignment Progress object by the date when a user completed a module.

To report on users' learning assignment progress, filter your report on the Title field from the Learning Content object. You can't create a report that shows users' progress on all assigned Trailhead modules.

You can't track how many public groups or public group members you assigned a learning item to.

For more details on limitations for reports on Learning Paths, see Limits on Report Types and Report Considerations for Salesforce Connect—All Adaptors in Salesforce Help.

See Also

- [Assign a Learning Item for Learning Paths](#)
- [Create a Custom Report Type](#)
- [Limits on Report Types](#)
- [Manage Custom Report Types](#)
- [Build a Lightning Experience Dashboard](#)
- [Salesforce Connect Support for Reports](#)

Turn Off Learning Paths

Optionally, you can turn off Learning Paths, which hides the Learning Home and prevents admins from assigning learning items to users. When Learning Paths is turned off, the Guidance Center remains available.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Professional, Enterprise, Performance, and Unlimited** Editions

Not Available for: **Chatter External, Chatter Free, or Communities** (now known as Experience Cloud) user licenses

Enablement Sites (myTrailhead) available in: **Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

Change Guidance Center Settings

Modify All Data

OR

Customize Application

 **Important** Learning Paths is turned off by default in all environments in Winter '25 and later. To turn on Learning Paths, contact Salesforce Customer Support. Learning Paths isn't available in a sandbox or in Government Cloud Plus environments.

1. From Setup, in the Quick Find box, enter *Guidance Center*, and then select **Guidance Center**.
2. Turn off **Learning Paths**.

Guidance Center

The Guidance Center panel in Lightning Experience provides a convenient location for showing content that's assigned to or recommended for the current user. The Guidance Center includes not only suggested content that's authored by Salesforce but also custom content that you manage. With Learning Paths, you can manage the learning items—custom links, videos, and Trailhead modules—that are available in the Guidance Center and assign those items to specific users with optional due dates.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Starter, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

 **Note** The Guidance Center isn't supported in the Salesforce mobile app. Trailhead modules aren't available in the Guidance Center in sandbox environments.

[Explore the Guidance Center](#)

Open the Guidance Center to easily access help while in the flow of work.

[Explore Guidance Sets](#)

Salesforce admins, Enablement admins, and all Starter Edition users have access to guidance sets in the Guidance Center. Guidance sets offer personalized setup and learning resources that align with your experience level and the products you're using. Each guidance set includes suggested resources to help you successfully complete your job.

Customize Content in the Guidance Center

To add your own learning items in the Guidance Center, assign content with Learning Paths. Or, optionally hide Learning Home or Salesforce suggestions.

Configuring the Size and Position of the Guidance Center

Personalize the in-app content experience with options for pinning and unpinning the Guidance Center panel. The Guidance Center panel also automatically expands for videos and Trailhead modules, offering more room on the screen for your content.

Turn Off Suggested Content in the Guidance Center

Optionally, you can turn off Salesforce suggestions for content in the Guidance Center for end users. Suggested content still appears for admins. The Guidance Center remains available so you can assign specific learning items with Learning Paths.

Turn Off Salesblazer Content in the Guidance Center

Optionally, turn off Salesblazer content in the Guidance Center for end users. The Guidance Center remains available for other content.

Explore the Guidance Center

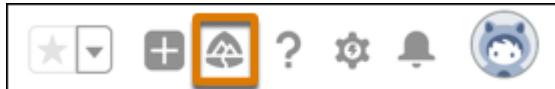
Open the Guidance Center to easily access help while in the flow of work.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Starter, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

To open the Guidance Center panel, in the global header, click the Guidance Center icon.



The Guidance Center panel includes these sections.

Guidance Center   

Selected for You  

 Guidance Set
Get to Know Your Customers
Understanding how accounts and contacts work together in Salesforce is essential to know who your customers are, where to find them, how to contact them, and how you can make them happy.
3 steps

 Trailhead
Salesforce User Basics
Get started with Salesforce and learn how to make it work for your bottom line.
+300 Points  ~1 hr 5 mins

Salesblazer 
 Article
Why the Best Account Managers Are Both Hunters and Farmers
To keep customers for as long as possible, account managers must be laser-focused and nurturing. Sales and...
~7 mins

 Article
11 Tactics to Level up Your Sales Game
When you're trying to close deals, it's easy to let ambition take the wheel. The best sellers, however, focus on using th...
~9 mins

Related to This Page 
 Help
Find Your Way Around Lightning Experience

 Help
Set Up Lightning Experience Home

Selected for You (1)

Includes suggested content that's specific to your role. Learning items assigned through Learning Paths also appear here when they have a due date but no specific object or record location. Content is removed from this section when you complete it.

Salesblazer (2)

Includes insights, tips, and strategies curated for and written by sales leaders and professionals, originally published on the Salesforce website. Content in the Salesblazer section opens in a new browser tab. Clicking an item in the Salesblazer section opens the content in a new browser tab. Items remain in the Salesblazer section until the source content is refreshed, which occurs weekly on average.

 **Important** The Salesblazer section isn't available in Government Cloud Plus environments.

Related to This Page (3)

Includes contextual content that's specific to the page that you're currently working on. Learning items assigned through Learning Paths also appear here when they have a specific object or record location. Content remains visible in this section when complete it so you can refer to it again. If there's no contextual content or assignments for the page, the section is removed.

If your company uses Enablement for sales reps, the Guidance Center also shows a user's assigned Enablement programs.

See Also

[Take Enablement Programs](#)

Explore Guidance Sets

Salesforce admins, Enablement admins, and all Starter Edition users have access to guidance sets in the Guidance Center. Guidance sets offer personalized setup and learning resources that align with your experience level and the products you're using. Each guidance set includes suggested resources to help you successfully complete your job.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Starter, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

 **Important** Guidance sets aren't available in a sandbox.

1. Open the Guidance Center.

The Selected for You section shows up to two guidance sets.

Guidance Center

Selected for You

View More

Guidance Set
Use Multi-Factor Authentication for Salesforce Logins

Got multi-factor authentication (MFA) yet? To protect against security threats like phishing attacks, everyone is required to use MFA when logging in to Salesforce. Learn how to enable MFA for your users now — and understand what to expect when Salesforce automatically enables MFA for your org.

6 steps

Guidance Set
Use Email and Salesforce Together

Enable your email channels and understand the considerations.

3 steps

Trailhead
Salesforce User Basics

Get started with Salesforce and learn how to make it work for your bottom line.

+300 Points ~1 hr 5 mins

2. To find all available guidance sets, click **View More**.

As you open resources and complete steps in a guidance set, they're marked as completed.

- Note** You can't turn off guidance sets from Setup. Instead, submit a Salesforce Customer Support case.

Customize Content in the Guidance Center

To add your own learning items in the Guidance Center, assign content with Learning Paths. Or, optionally hide Learning Home or Salesforce suggestions.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Starter, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

Change Guidance Center Settings

Modify All Data OR Customize Application

For information about assigning learning items with Learning Paths, see [Assign a Learning Item for Learning Paths](#).

You can also turn off certain content in the Guidance Center.

- [Turn Off Learning Paths](#)
- [Turn Off Suggested Content in the Guidance Center](#)
- [Turn Off Salesblazer Content in the Guidance Center](#)

 **Note** To turn off guidance sets, submit a Salesforce Customer Support case.

See Also

[Learning Paths](#)

Configuring the Size and Position of the Guidance Center

Personalize the in-app content experience with options for pinning and unpinning the Guidance Center panel. The Guidance Center panel also automatically expands for videos and Trailhead modules, offering more room on the screen for your content.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Starter, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Pinning and Unpinning the Guidance Center

The Guidance Center panel can overlap some elements of the page in Lightning Experience. To split the screen between the Guidance Center and the Lightning Experience page, side by side with no overlap, click the pin icon  in the Guidance Center header.

Salesforce remembers a user's pin selection for the browser and device they're on.

Tip Pinning works best on wide screens. If the screen is too narrow, the Guidance Center can still overlap content on the page even when it's pinned.

To exit the split view and have the Guidance Center overlap the Lightning Experience page again, click the unpin icon  in the Guidance Center header.

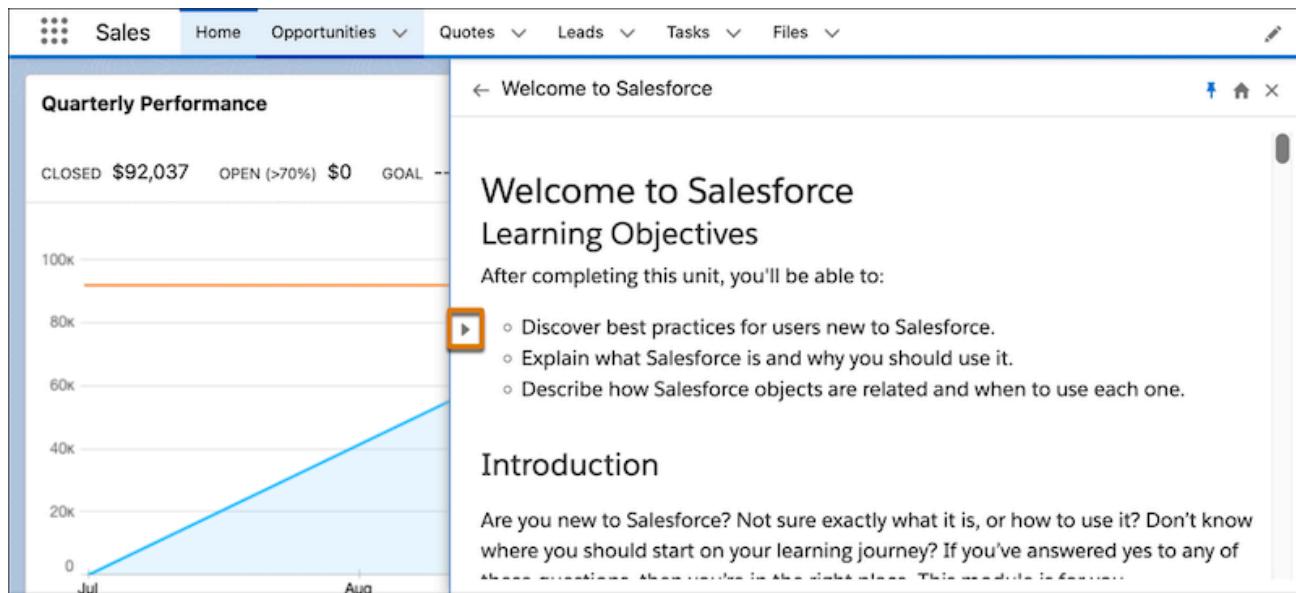
Note If you're using Enablement programs, the Guidance Center is pinned by default.

To turn off pinning and unpinning, contact Salesforce Customer Support.

Expanded View for Videos and Trailhead Content

Some content types in the Guidance Center, such as videos or Trailhead units, are easier to experience when you have more room on the screen. For these content types, the Guidance Center automatically expands to a larger width.

To resize the Guidance Center after it expands, toggle the arrow icon.



Turn Off Suggested Content in the Guidance Center

Optionally, you can turn off Salesforce suggestions for content in the Guidance Center for end users. Suggested content still appears for admins. The Guidance Center remains available so you can assign specific learning items with Learning Paths.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Starter, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

Change Guidance Center Settings

Modify All Data

OR

Customize Application

1. From Setup, in the Quick Find box, enter *Guidance Center*, and then select **Guidance Center**.
2. Turn off **Salesforce Suggestions for End Users**.

Turning off Salesforce suggestions also turns off Salesblazer content in the Guidance Center. When Salesforce suggestions are on, you can optionally [turn off Salesblazer content separately](#).

 **Note** The Salesforce Suggestions for End Users option doesn't apply to guidance sets. To turn off guidance sets, submit a Salesforce Customer Support case.

Turn Off Salesblazer Content in the Guidance Center

Optionally, turn off Salesblazer content in the Guidance Center for end users. The Guidance Center remains available for other content.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Starter, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

Change Guidance Center Settings:

Modify All Data

OR

Customize Application

1. From Setup, in the Quick Find box, enter *Guidance Center*, and then select **Guidance Center**.
2. Turn off **Salesblazer Content**.

-  **Note** If you turn off Salesforce suggestions for content in the Guidance Center, Salesblazer content also turns off.

See Also

Turn Off Suggested Content in the Guidance Center

Salesforce: Salesblazer Website

Customize the Help Menu in Lightning Experience

The question mark icon in the global header opens a menu of Salesforce resources. Supplement these links by adding a global section with links to your own content. There's only one custom Help Menu section per org and it appears at the top of the Help Menu.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS NEEDED

To create custom help:

Customize Application OR Modify All Data

-  **Note** To create an improved and more robust learning experience for your users, check out [Guidance Center](#).

 1. From Setup in Lightning Experience, in the Quick Find box, enter *Help Menu*, and then select **Help Menu**.
 2. Enter a title for the custom section. Salesforce recommends naming the section so that users understand that the resources are custom help for your org or company. For example, *Acme Company Help*. Labels aren't translated and appear as you entered them.
 3.  **Tip** List the most important resources first. Only the first two resources are shown in each section. Users can view all resources by clicking the section name or the arrow.

Add labels and URLs for the resources. You can add up to 30 resources. Items are listed in the Help Menu in the order that they appear on the setup page.
 4. Save your changes.
 5. Turn on **Customize the Help Menu**.

If you find that some sections of the Help Menu aren't working for your users, simply hide them. Under Salesforce Help Content, turn off the links that you want to hide. As an admin with the Customize Application or Modify All Data user permission, you always see all resources, including a link to the release notes.

- ! Important** If you install a package with custom Help Menu resources, they don't appear in your Help Menu Setup page or in the Help Menu user interface. To add resources per the package

suggestions, use the [CustomHelpMenuItem](#) and [CustomHelpMenuSection](#) SOAP API objects to view the information contained in the package. Then manually add any resources to the Help Menu Setup page.

The Help Menu isn't supported in the Salesforce mobile app.

Custom Help in Salesforce Classic

In Salesforce Classic, object-level help replaces the links for a custom object or external object page. Replace built-in Salesforce Help with documentation that's customized for your users.

[Object-Level Help in Salesforce Classic](#)

Help your users by providing object-level help for all custom objects and external objects. This way, when your users click the **Help for this Page** link on your custom object, they'll find useful information that's relevant to your custom object. When you add custom help to a custom or external object, the **Help for this Page** link on those object pages displays your custom help instead of generic help. Your users can access the custom help content from the object home (overview), detail, and edit pages, list views, and related lists.

[Define Object-Level Help in Salesforce Classic](#)

Object-level help overrides the **Help for this Page** links for a custom object or external object with your own help content contained in a Visualforce page. To make object-level help available to all your users, create a Visualforce page that contains your help content. Then edit the custom or external object definition to reference the page. Object-level help becomes available to all your users instantly.

[Create a Custom Object Help Page with Static Content in Salesforce Classic](#)

If you know HTML, it's easy to add help to your custom objects by writing the content in HTML and saving it in a Visualforce page. No need to learn Visualforce. Just use the template that we provide.

[Create Custom Object Help with a PDF File in Salesforce Classic](#)

Add help to your custom objects by creating Visualforce pages that redirect to PDF help files or a URL. No need to learn Visualforce. Just use the template that we provide.

[Object-Level Help Considerations in Salesforce Classic](#)

Before defining object-level help text for your custom or external objects, review these best practices and implementation considerations.

[Define Org-Level Help in Salesforce Classic](#)

If you rename standard tabs, objects, fields, and other related user interface labels, you can also replace the built-in Salesforce Help with documentation that's customized specifically for your users. To replace the built-in help, simply provide a URL to your custom help. This feature is available for Salesforce Classic only.

Object-Level Help in Salesforce Classic

Help your users by providing object-level help for all custom objects and external objects. This way, when your users click the **Help for this Page** link on your custom object, they'll find useful information that's relevant to your custom object. When you add custom help to a custom or external object, the **Help for**

this Page link on those object pages displays your custom help instead of generic help. Your users can access the custom help content from the object home (overview), detail, and edit pages, list views, and related lists.

 **Note** If you don't create object-level help, the **Help for this Page** link provides information about standard objects that won't be relevant to your custom object. You can override the **Help for this Page** links for a custom object or external object with help content contained in a Visualforce page. But don't worry! You don't have to learn Visualforce to add help content to your custom objects.

See Also

- [Custom Help Content](#)
- [Define Object-Level Help in Salesforce Classic](#)
- [Object-Level Help Considerations in Salesforce Classic](#)

Define Object-Level Help in Salesforce Classic

Object-level help overrides the **Help for this Page** links for a custom object or external object with your own help content contained in a Visualforce page. To make object-level help available to all your users, create a Visualforce page that contains your help content. Then edit the custom or external object definition to reference the page. Object-level help becomes available to all your users instantly.

REQUIRED EDITIONS

Available in: Salesforce Classic

Custom objects are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Salesforce Connect external objects are available in Developer Edition and, for an extra cost, in Enterprise, Performance, and Unlimited Editions.

USER PERMISSIONS NEEDED

To define or change object-level help: [Customize Application](#)

1. Create a Visualforce page that contains your help content.
2. Edit the [custom object definition](#) or [external object definition](#) that uses the custom help when users click the **Help for this Page** link.
3. For Context-Sensitive Help Setting, select **Open a window using a Visualforce page**.
4. Select the Visualforce page that contains your help content.
5. Save your work.

See Also

- [Object-Level Help in Salesforce Classic](#)
- [Custom Help Content](#)

Create a Custom Object Help Page with Static Content in Salesforce Classic

If you know HTML, it's easy to add help to your custom objects by writing the content in HTML and saving it in a Visualforce page. No need to learn Visualforce. Just use the template that we provide.

REQUIRED EDITIONS

Available in: Salesforce Classic

Custom objects are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Salesforce Connect external objects are available in: **Developer** Edition and for an extra cost in: **Enterprise, Performance, and Unlimited** Editions

Visualforce is available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To define or change object-level help: Customize Application

To create or edit Visualforce pages: Customize Application

1. From Setup, enter *Visualforce Pages* in the Quick Find box, then select **Visualforce Pages**.
2. Click **New**.
The Visualforce page editor opens with a new page.
3. Complete the following fields.

Field	Description
Label	The human-friendly name of the page used to identify the page in Setup tools. It's a great idea to have a naming convention for your custom help pages. For example, start all custom help pages with "Help_" and then the object name.
Name	The API name for the page. You can use the auto-filled value.
Description	An optional description of the page.
Available for Lightning Experience, Experience Builder sites, and the mobile app	Select this option if your custom object is available in the Salesforce mobile app.

4. Click **Quick Save**.
5. In the **Visualforce Markup** tab code editor, select the default code and delete it.
6. Paste the following help template code into the code editor.

```
<apex:page showHeader="false">  
  
    <!-- Add your help styles -->  
  
    <h1>Help for {YourObjectName} Object</h1>  
  
    <p>Your custom help message here.</p>  
  
</apex:page>
```

7. Click **Quick Save**.
8. Edit the template to add your help content.
To add formatting to your page, use HTML markup. You can also use Visualforce markup if you know it.
9. Click **Save**.

You can now add this page as custom help. When users click **Help for this Page**, they see this page in the Help & Training window.

See Also

- [Define Object-Level Help in Salesforce Classic](#)
- [Create Custom Object Help with a PDF File in Salesforce Classic](#)
- [Object-Level Help Considerations in Salesforce Classic](#)

Create Custom Object Help with a PDF File in Salesforce Classic

Add help to your custom objects by creating Visualforce pages that redirect to PDF help files or a URL. No need to learn Visualforce. Just use the template that we provide.

REQUIRED EDITIONS

Available in: Salesforce Classic

Custom objects are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Salesforce Connect external objects are available in: **Developer Edition** and for an extra cost in: **Enterprise, Performance, and Unlimited Editions**

Visualforce is available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To define or change object-level help:

Customize Application

USER PERMISSIONS NEEDED

To create or edit Visualforce pages: **Customize Application**

You can write your help content in an authoring tool such as Microsoft Word and convert it to a PDF file.

1. Upload your PDF file as a static resource.
Users see this PDF file when they request help with the custom object.
2. From Setup, enter *Visualforce Pages* in the Quick Find box, then select **Visualforce Pages**.
3. Click **New**.
The Visualforce page editor opens with a new page.
4. Complete the following fields.

Field	Description
Label	The human-friendly name of the page used to identify the page in Setup tools. It's a great idea to have a naming convention for your custom help pages. For example, start all custom help pages with "Help_" and then the object name.
Name	The API name for the page. You can use the auto-filled value.
Description	An optional description of the page.
Available for Lightning Experience, Experience Builder sites, and the mobile app	Select this option if your custom object is available in the Salesforce mobile app.

5. Click **Quick Save**.
6. In the **Visualforce Markup** tab code editor, select the default code and delete it.
7. Paste the following help template code into the code editor.

```
<apex:page showHeader="false" action="{! URLFOR ($Resource.YourCustomHelpReso
urce) }">

    <!-- This page redirects to the URL in the action attribute above -->

    <p>Redirecting to help content...</p>

</apex:page>
```

8. Replace `YourCustomHelpResource` in the action attribute with the name of the static resource that you uploaded.
9. Click **Save**.

You can now add this page as help. When users click **Help for this Page**, they're redirected to the resource you set in the action attribute.

-  **Note** The user's browser controls the behavior of a PDF link, not your Visualforce page. The PDF content might display in the browser or be downloaded as a PDF file.

See Also

- [Define Object-Level Help in Salesforce Classic](#)
- [Create a Custom Object Help Page with Static Content in Salesforce Classic](#)
- [Object-Level Help Considerations in Salesforce Classic](#)

Object-Level Help Considerations in Salesforce Classic

Before defining object-level help text for your custom or external objects, review these best practices and implementation considerations.

REQUIRED EDITIONS

Available in: Salesforce Classic

Custom objects are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Salesforce Connect external objects are available in: **Developer Edition** and for an extra cost in: **Enterprise, Performance, and Unlimited Editions**

Visualforce is available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Best Practices

- The window that displays your object-level help has the same height and width dimensions as the standard Salesforce Help & Training window. To increase the usability of your help content, size and style your content appropriately.
- Your Visualforce help content pages can use merge fields or other functions to make the experience more personalized. For example, you can design the help to add the user's name when the help page is displayed.

Advanced Implementation Considerations

- Create custom help Visualforce pages without a controller, or use a custom controller. You can't use a standard controller.
- If you have defined object-level help for an object that you add to a Salesforce AppExchange package, Salesforce adds the Visualforce page or static resource referenced in your **Context-Sensitive Help Settings** for that object.
- In managed packages, object-level help is locked to the developer, giving installers the ability to change it if needed.

See Also

- [Define Object-Level Help in Salesforce Classic](#)
- [Object-Level Help in Salesforce Classic](#)

Define Org-Level Help in Salesforce Classic

If you rename standard tabs, objects, fields, and other related user interface labels, you can also replace the built-in Salesforce Help with documentation that's customized specifically for your users. To replace the built-in help, simply provide a URL to your custom help. This feature is available for Salesforce Classic only.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **All** Editions except **Database.com**

Users can view this URL whenever they click on any context-sensitive help link on an end-user page or within their personal settings. After you replace the help, the Help & Training link at the very top of every page and all Setup pages will continue to display the Salesforce Help.

1. From Setup, enter *Help Settings* in the Quick Find box, then select **Help Settings**.
 2. Enter the complete URL for your help file that you would like to replace the Salesforce Help.
 3. Click **Save**.
- When you replace the Salesforce Help with your own help file, the Help & Training link still displays Salesforce Help. However, other than within Setup, the Help for this Page links on all pages are no longer context-sensitive. That is, your help file will open at the same place regardless of which page the user is viewing when they click the link.
 - You can make your help context-sensitive by taking advantage of the context-specific parameters that are passed with each help link. For example, the help link from the Opportunities tab home page is constructed as follows (without any line breaks):

`http://your_help_file.com?loc=help&body=%2Fhelp%2Fdoc%2Fen%2Fhelp2.jsp&target=opp_overview`

The values of the `target` and `section` parameters are unique for every page within the application. You can parse these parameters to display context-sensitive help content for your users.

Tailor Business Processes to Different Record Types Users

Record types let you offer different business processes, picklist values, and page layouts to different users. You can create record types to differentiate your regular sales deals from your professional services engagements, offering different picklist values for each. Or you can display different page layouts for your customer support cases versus your billing cases.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Considerations for Creating and Updating Record Types and Picklists

Keep these considerations in mind when working with record types and business process picklists.

Create Record Types

Before creating record types, include all of the possible record type values in your master list of picklists. The master picklist is a complete list of picklist values that can be used in any record type.

Use Record Types

Here's an example of how record types can work in your org. Let's say you have two sales divisions, Hardware and Consulting, and only your Consulting division receives leads through seminars. You can choose to display the Seminar contact lead source for the Consulting division only.

Edit Picklists for Record Types and Business Processes

Customize the values in record type or business process picklists based on your organization's unique needs.

Limitations for Creating and Updating Record Types and Picklists

Keep these limitations in mind when working with record types and business process picklists.

Managing Multiple Business Processes

Use multiple business processes to display different picklist values according to each user's profile. Use multiple business processes to track separate sales, support, and lead lifecycles.

Create Multiple Business Processes

Follow these steps to create sales processes, support processes, lead processes, and solution processes.

See Also

[How Is Record Type Access Specified?](#)

Considerations for Creating and Updating Record Types and Picklists

Keep these considerations in mind when working with record types and business process picklists.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

General

- If each profile is associated with a single record type, users will never be prompted to select a record

- type when creating records.
- Don't name your record type Master because the name is reserved.
 - Don't use record types as an access control mechanism. Profile assignment governs create and edit access for an object but doesn't govern read access. For example, a user assigned to a profile that isn't enabled for a particular record type can't create records with that record type, but can access records associated with that record type. Users with access to an object can read all record type information for that object.
 - We strongly recommend against storing sensitive information in the record type description, name, or label. Instead, store sensitive information in a separate object or fields to which you applied appropriate access controls.
 - A user can be associated with several record types. For example, a user who creates marketing campaigns for both US and European divisions can have both US and European campaign record types available when creating campaigns.
 - When creating and editing record types for accounts, opportunities, cases, contacts, or custom objects, check for criteria-based sharing rules that use existing record types as criteria. A record type change can affect the number of records that the rule shares. For example, let's say you have a record type named Service, and you created a criteria-based sharing rule that shares all Service record types with your service team. If you create another record type named Support, and you want these records shared with your service team, update the sharing rule to include Support record types in the criteria.
 - Deleting a record type also deletes the related path.
 - Business and person accounts require at least one active record type.
 - Deleting campaign member record types updates the Campaign Member Type field on campaign and campaign member records.
 - Person accounts are account records to which a special record type has been assigned. These record types are called *person account record types*. Person account record types allow contact fields to be available on the account and allow the account to be used as if it were a contact. A default person account record type named Person Account is automatically created when person accounts are enabled for your org. You can change the name of this record type, and you can create more person account record types.
 - From the UI, you can change an account's record type from a business account to a business account or from a person account to a person account. However, to change an account's record type from a business account to a person account, or vice versa, you must use the API.
 - When overriding a standard action with an Aura component on an object with more than one active record type, the record type selection screen uses Salesforce Classic styling on mobile devices.
 - When users convert, clone, or create records, these special considerations apply.
 - When a user converts a lead, the new account, contact, and opportunity records use the default record type for the owner of the new records. The user can choose a different record type during conversion.
 - When a user clones a record, the new record has the record type of the cloned record. If the user's profile doesn't have access to the record type of the cloned record, the new record adopts the user's default record type.
 - When a user creates a case or lead and applies assignment rules, the new record can keep the creator's default record type or take the record type of the assignee, depending on the case and lead settings specified by the administrator.
 - Changing a record type causes Lightning pages to refresh.

- Field accessibility settings for Record Type fields are ignored in Lightning Experience. For example, if you set the Opportunity Record Type field access settings to **Read-Only**, you can't edit that field from a record page in Salesforce Classic, but you can from a record page in Lightning Experience. To get around this issue, use a validation rule to prevent changes to the field in Lightning Experience.
- Before creating a record using Apex, ensure that the object's default record type is active.

Deactivating Record Types

Consider these guidelines if you're deactivating a record type.

- Deactivating a record type doesn't remove it from any user profiles or permission sets.
- Deactivating a record type means that no new records can be created with the record type. However, any records that were previously created with the record type are still associated with it and with its associated page layout.
- To deactivate all record types from an object, remove all record types from all the profiles and deactivate the record types. Then create one new record type and activate it, but don't add it to any profiles. One record type must exist to enable existing records that used the deactivated record types to display properly.

If you encounter any issues inline editing a record in Lightning Experience after deactivating a record type, edit the Page Layout Assignment so that another layout on the object, such as the default layout, is used for the custom record type. Otherwise, consider reactivating the disabled record type.

Record Types and Picklists

- Before creating record types, include all the possible record type values in your master list of picklists. The master picklist is a complete list of picklist values that can be used in any record type.
- Changing the default value of the master picklist doesn't affect the default value of the picklist for a record type.
- The master picklist is independent of all record types and business processes. If you add a picklist value to the master picklist, you must manually include the new value in the appropriate record types. If you remove a picklist value from the master, it's no longer available when creating records, but records assigned to that value are unchanged.
- When you create a record type without cloning an existing one, the new record type automatically includes the master picklist values for both standard and custom picklists. You can then customize the picklist values for the record type.
- To retrieve custom picklist values with record types from an org that has source tracking disabled, add the custom picklist fields to the org's `package.xml` file.

See Also

[Tailor Business Processes to Different Record Types](#) [Users](#)

[Create Record Types](#)

[Limitations for Creating and Updating Record Types and Picklists](#)

Create Record Types

Before creating record types, include all of the possible record type values in your master list of picklists. The master picklist is a complete list of picklist values that can be used in any record type.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create or change record types: Customize Application

1. From the management settings for the appropriate object, go to Record Types.
2. Click **New**.
3. Select **Master** from the Existing Record Type dropdown list to copy all available picklist values, or choose an existing record type to clone its picklist values.
 **Note** When you create a record type without cloning an existing one, the new record type automatically includes the master picklist values for both standard and custom picklists. You can then customize the picklist values for the record type.
4. Enter a Record Type Label that's unique within the object.
-  **Important** Don't name your record type Master because it's reserved for record types.
5. Enter a Record Type Name. The Record Type Name refers to the component when using Metadata API and prevents naming conflicts on package installation in managed packages.
6. For opportunity, case, lead, and solution record types, select a business process to associate with the record type.
7. Enter a description.
8. Select **Active** to activate the record type.
9. Select **Make Available** next to a profile to make the record type available to users with that profile. Select the checkbox in the header row to make it available for all profiles.
 **Tip** If each profile is associated with a single record type, users will never be prompted to select a record type when creating records. Users assigned to a record type can still view and edit records associated with record types not enabled for their profiles.
10. For selected profiles, select **Make Default** next to a profile to make it the default record type for users of that profile. Select the checkbox in the header row to make it the default for all profiles.
11. Click **Next**.
12. Choose a page layout option to determine what page layout displays for records with this record type:
 - To apply a single page layout for all profiles, select **Apply one layout to all profiles** and choose the page layout from the dropdown list.
 - To apply different page layouts based on user profiles, select **Apply a different layout for each profile** and choose a page layout for each profile.
13. Click **Save** to edit the values of the standard and custom picklists available for the record type, or click

Save and New to create another record type.

See Also

- [Tailor Business Processes to Different Record Types Users](#)
- [Considerations for Creating and Updating Record Types and Picklists](#)
- [Limitations for Creating and Updating Record Types and Picklists](#)

Use Record Types

Here's an example of how record types can work in your org. Let's say you have two sales divisions, Hardware and Consulting, and only your Consulting division receives leads through seminars. You can choose to display the Seminar contact lead source for the Consulting division only.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create or change record types:	Customize Application
-----------------------------------	-----------------------

1. Manage your master picklists.
Define a list of contact Lead Source picklist values that contains all the values used by both the Hardware and Consulting divisions, including Seminar.
2. Create record types.
Create two contact record types: one called Hardware and another called Consulting. This step includes adding master picklist values to the record types.
3. Add record types to profiles.
Add the Hardware record type to the profiles for all users in the Hardware sales division. Add the Consulting record type to the profiles of all users in the Consulting sales division.
4. Set personal options for the record types.
Let users of both the Hardware and Consulting sales divisions bypass the prompt that asks them to select a record type when creating a contact. If you have users that create contact records for both sales divisions, they can customize their personal settings to always prompt them to select a record type.

See Also

- [Considerations for Creating and Updating Record Types and Picklists](#)
- [Tailor Business Processes to Different Record Types Users](#)

Edit Picklists for Record Types and Business Processes

Customize the values in record type or business process picklists based on your organization's unique needs.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Record types are available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Business processes are available in: **Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create or change record types: Customize Application

To create or change business processes: Customize Application

1. Select a record type or business process and click **Edit** next to the picklist field to change its values.
2. Add or remove values as needed.
Users can choose from these values when creating or editing records.
3. Optionally, choose a default picklist value.
Some picklists require a default value. The default value in a dependent field is ignored.
4. Click **Save**.

See Also

[Salesforce Help: Create Custom Fields](#)

Limitations for Creating and Updating Record Types and Picklists

Keep these limitations in mind when working with record types and business process picklists.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

- These special picklist fields aren't available for record types because they're used exclusively for sales processes, lead processes, support processes, and solution processes.
 - Stage field on Opportunity
 - Status field on Case and Solution
 - Lead StatusYou can use these fields to provide different picklist values for different record types by assigning a different process to each record type.
- These campaign member picklists aren't available for these record types.
 - Status
 - Salutation
 - Lead Source
- If the object is referenced in Apex, you can't edit or delete a record type for the object.

- If a record type is in use by an email routing address for Email-to-Case or On-Demand Email-to-Case, you can't deactivate the record type.
- Record types can only be assigned to campaign members using the Campaign Member Type field on new or existing campaigns. To assign record types to campaign members, add the Campaign Member Type field to the campaign page layout. You must have the Marketing User user permission to change the campaign member type. You can also add a read-only Campaign Member Type field to the campaign members page layout.
- We recommend creating no more than 200 record types. Orgs can have difficulty managing their record types if they exceed 200.
- When the org source tracking permission is enabled and you try retrieving information about a specific custom field associated with a record type, you retrieve information for all the fields associated with that record type. However, when the org source tracking permission is disabled, you retrieve information only for the standard fields associated with the record type and the field mentioned in the package.xml file.

See Also

[Create Record Types](#)

[Considerations for Creating and Updating Record Types and Picklists](#)

Managing Multiple Business Processes

Use multiple business processes to display different picklist values according to each user's profile. Use multiple business processes to track separate sales, support, and lead lifecycles.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create or change business processes: Customize Application

Sales Processes

Create different sales processes that include some or all of the picklist values available for the opportunity Stage field.

Lead Processes

Create different lead processes that include some or all of the picklist values available for the Lead Status field.

Support Processes

Create different support processes that include some or all of the picklist values available for the case Status field.

Solution Processes

Create different solution processes that include some or all of the picklist values available for the Status field.

After creating a sales, support, lead, or solution process, assign the process to a record type. The record type determines the user profiles that are associated with the business process.

To view a list of business processes, from Setup, enter *Processes* in the Quick Find box, then select the appropriate link.

See Also

[Edit Picklists for Record Types and Business Processes](#)

Create Multiple Business Processes

Follow these steps to create sales processes, support processes, lead processes, and solution processes.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create or change business processes: Customize Application

1. From Setup, enter *Processes* in the Quick Find box, then select the appropriate link.
2. Click **New**.
3. Choose an existing process to copy its picklist values into the new process. Select **Master** to copy all available picklist values.
4. Enter a name and description for the new process. The name must be unique within the tab.
5. Click **Save**.

All of the available values in the picklist are displayed. Choose the values that you would like included in the new business process.

Next, add the new business process to a record type, and then make the record type available to users based on profile.

Manage Your Translations

If your Salesforce org has multiple languages enabled, manage translations so that your global users can use Salesforce in their language.

REQUIRED EDITIONS

Metadata translation available in: Salesforce Classic and Lightning Experience

Data translation available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Data translation applies to: B2B Commerce

-  **Note** Standard objects aren't available in Translation Workbench. Use the rename tabs and labels interface for standard object translation.

Metadata Translation

When you enable multiple languages in your Salesforce org, Salesforce translates some labels for you, based on the language type. For labels without a default translation, you can localize your apps and custom functionality for any Salesforce supported language through metadata translation.

Data Translation

When data translation is enabled, the data stored in certain objects and fields are available for translation. You can also enable data translation for custom text and URL fields on objects.

Translation Workbench

Use Translation Workbench to maintain translated values for metadata and data labels in your Salesforce org. Specify languages for translation and assign translators for each language. Manage translated values for any Salesforce supported language. Translators can maintain translations directly through the workbench, or you can export translation files for bulk translation imports.

Translation Considerations

Review considerations for managing your translations and translating flows.

Metadata Translation

When you enable multiple languages in your Salesforce org, Salesforce translates some labels for you, based on the language type. For labels without a default translation, you can localize your apps and custom functionality for any Salesforce supported language through metadata translation.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Salesforce-provided translations vary by language type.

Language Type	Translations Provided by Salesforce
Fully supported languages	Metadata labels for all standard features, plus Help.

Language Type	Translations Provided by Salesforce
End-user languages	Metadata labels for all standard objects and pages, except admin pages and Setup. No translations for Help.
Platform-only languages	None

In situations where Salesforce doesn't provide default translations, metadata translation allows you to localize apps and custom functionality that you build in Salesforce. You can translate items such as custom labels, custom objects, and field names.

Metadata Available for Translation

You can translate metadata labels only for certain Salesforce Setup components.

Flow Components for Metadata Translation

Flow components are the parts of a flow you can translate.

See Also

[Supported Languages](#)

[Translation Workbench](#)

Metadata Available for Translation

You can translate metadata labels only for certain Salesforce Setup components.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

 **Note** Standard objects aren't available in Translation Workbench. Use the rename tabs and labels interface for standard object translation.

To view the translatable metadata labels in your Salesforce organization, first enable Translation Workbench. Then, from the Translate Setup page, select a Setup component. If needed, select **Object**, **Custom Report Type Entity**, **Flow**, **Flow Type**, **Flow Component**, or **Aspect**.

You can translate the following components.

- Action
- Address Country
- Address State
- Apex Sharing Reason
- App
- Button and Link Label
- Chatter Extension

- Custom Field
- Custom Labels
- Custom Report Type
- Dashboard*
- Dashboard Component*
- Data Category
- Data Category Group
- Division
- Feed Filter
- Field Set
- Flow
- Global Value Set
- Layout Section
- Lookup Filter
- Managed Content Node Type
- Managed Content Type
- Navigation Menu Item (for Experience Cloud sites)
- Path Step Rich Text
- Picklist Value (For standard picklist values provided by Salesforce, only editable picklist values are available in the Translation Workbench.)
- Prompt
- Prompt Version
- Record Type
- Report*
- Reputation Level (for Experience Cloud sites)
- S-Control
- Solution Category
- Stamp
- Standard Field Help
- Timeline Object Definition
- Validation Error Message
- Web Tab (also includes Lightning component and Visualforce tabs)
- Workflow Task

 **Important** Visualforce pages supersede s-controls. Organizations that haven't previously used s-controls can't create them. Existing s-controls are unaffected and can still be edited.

 **Note** The components marked with an asterisk (*) cannot be exported.

See Also

[Supported Languages](#)
[Translation Workbench](#)

Flow Components for Metadata Translation

Flow components are the parts of a flow you can translate.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Flow Component	Description	Where Translation Appears
Definition	The flow name	The flow interview's title bar.
Version	The version label of a flow	The active flow version label is the flow definition label by default, so it appears in the flow interview's title bar. When you enter the flow definition and its translation manually, the translated definition label overrides the active flow version label. When the flow definition label doesn't have a translation, the translated version label appears as the flow interview's title.
Screen Info	Aspect includes help text and paused messages	Help text and paused messages for the overall screen.
Screen Field	Aspect includes labels, description, help text, and error messages for screen components	Field-level text on a screen. The description aspect is the text for screen output components.
Choice	Aspect includes field labels, help text, text input labels	Field-level text for choice components.
Stage	Stage labels	Screen components that can reference stage labels, including Display Text components and attributes for screen components that require Lightning runtime.
Text Template	Aspect includes text in a text template	All the pages of a survey. Available only in Salesforce Surveys.
Custom Error	Aspect includes error messages in a Custom Error element	A window on a record page, inline with a field on a record page, or as part of an API response if the flow is triggered by a record modification made by an API call. Available only in record-triggered flows.

See Also

[Metadata Translation](#)

- [Translation Workbench](#)
- [Considerations for Translating Flows](#)

Data Translation

When data translation is enabled, the data stored in certain objects and fields are available for translation. You can also enable data translation for custom text and URL fields on objects.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, and Developer** Editions

-  **Note** Data translation requires API version 48.0 or later.

[Enable Data Translation](#)

Allow translation of data stored in certain objects and fields.

[Enable Data Translation for Custom Fields](#)

Allow translation of data stored in custom fields on certain objects and fields. You can enable data translation on custom fields with a type of Text, Text Area, Text Area (Long), Text Area (Rich), and URL.

Enable Data Translation

Allow translation of data stored in certain objects and fields.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, and Developer** Editions

USER PERMISSIONS NEEDED

To view company information: View Setup and Configuration

To change company information: Modify All Data

-  **Note** Before enabling data translation, note these important considerations:

- Data translation requires API version 48.0 or later.
- Data translation counts against your Salesforce org's storage limits.
- Data translation is available on some objects and fields. For example, data translation is available on the Industries Record Alert, Product, and Product Category objects.

1. From Setup, in the Quick Find box, enter *Company Information*, and then select **Company Information**.
2. In the Organization Detail section, click **Edit**.
3. Select **Enable Data Translation**.
4. Click **Save**.

Data translation is available for the Name and Description fields through the Translation tab within Product. You can also manage your data translations through the Export and Import options within Translation Workbench.

Optionally, enable data translation for custom fields.

See Also

[Translation Workbench](#)

[Salesforce B2B Commerce and D2C Commerce](#)

[Salesforce B2B Commerce and D2C Commerce Enable Data Translation](#)

[Data and File Storage Allocations](#)

[Localize Record Alerts](#)

[Manage Objects' Data Translations](#)

Enable Data Translation for Custom Fields

Allow translation of data stored in custom fields on certain objects and fields. You can enable data translation on custom fields with a type of Text, Text Area, Text Area (Long), Text Area (Rich), and URL.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, and Developer** Editions

USER PERMISSIONS NEEDED

To change data translation settings:

Customize Application

AND

Modify All Data

Before enabling data translation for custom fields, you must enable data translation.

 **Note** Data translation requires API version 48.0 or later.

1. From Setup, in the Quick Find box, enter *Data Translation Settings*, and then select **Data Translation Settings**.
2. Select an object to enable data translation for its custom fields.
Only objects that support data translation are listed.

3. Select the custom fields that you want to make available for data translation.
Data translation can only be enabled on custom fields with a type of Text, Text Area, Text Area (Long), Text Area (Rich), and URL.
4. Click **Save**.

If your B2B Commerce Store supports multiple languages, data translation is available for the selected custom fields through the Translation tab within Product and Product Category. You can also manage your data translations through the Export and Import options within Translation Workbench.

See Also

- [Salesforce B2B Commerce and D2C Commerce](#)
- [Salesforce B2B Commerce and D2C Commerce Enable Data Translation](#)
- [Translation Workbench](#)
- [Localize Record Alerts](#)
- [Manage Objects' Data Translations](#)

Translation Workbench

Use Translation Workbench to maintain translated values for metadata and data labels in your Salesforce org. Specify languages for translation and assign translators for each language. Manage translated values for any Salesforce supported language. Translators can maintain translations directly through the workbench, or you can export translation files for bulk translation imports.

REQUIRED EDITIONS

Metadata translation available in: Salesforce Classic and Lightning Experience

Data translation available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Data translation applies to: B2B Commerce

-  **Note** Translation Workbench is only available for multi-language orgs. If you aren't sure whether you have a single-language or multi-language organization, contact Salesforce Customer Support.

[Enable or Disable Translation Workbench](#)

Translation Workbench allows you to specify languages for translation, assign translators, and manage your translations through the workbench or bulk translation.

[Add Translated Languages and Translators](#)

Add languages for translation, assign translators for each language, and activate or deactivate a language's translations.

[Translate Metadata Labels](#)

Create and update metadata translations for customizations you make to your Salesforce organization, such as custom picklist values and custom field labels.

[Override Translations in Second-Generation Managed Packages and Unlocked Packages](#)

You can override metadata translations for custom objects in namespaced unlocked packages and second-generation managed packages. For example, override the label on a custom field or workflow task.

[Export Metadata Translation Files](#)

To easily translate your metadata, create files for your translators that contain your Salesforce org's translatable metadata. Examples of translatable metadata include custom field labels, report type names, and picklist values.

[Export Data Translation Files](#)

If data translation is enabled in your Salesforce org, you can create files for your translators that contain your org's data translations. Examples of translatable data include B2B Commerce Product names and data stored in custom fields.

[Work with Translation Files](#)

Translate metadata labels or data translation text, or review existing translations, with XML Localization Interchange File Format (.xlf) or Salesforce Translation Format (.stf) files.

[Import Translated Files](#)

Import and update the translations for your Salesforce org's metadata, such as custom fields, report types, and picklist values. Or import and update data translations, such as Product names. Typically, you export translation files from Salesforce, then send them to outside translators or a translation agency for bulk translation activities. You then import the translated files.

[Common Errors with Exporting and Importing Translation Files](#)

Troubleshoot issues that you can encounter when exporting and importing files in Translation Workbench.

See Also

[Supported Languages](#)

[Rename Object, Tab, and Field Labels](#)

Enable or Disable Translation Workbench

Translation Workbench allows you to specify languages for translation, assign translators, and manage your translations through the workbench or bulk translation.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

User Permissions Needed

To enable and disable Translation Workbench:	Customize Application
--	-----------------------

- From Setup, in the Quick Find box, enter *Translation Language Settings*, and then select **Translation Language Settings**.

2. On the welcome page, click **Enable**.

Enabling Translation Workbench makes these changes to your Salesforce org. If a customized component doesn't have a translated value, the component uses the org's default language. When you deactivate a language, all translations for that language are still available in Translation Workbench. However, users with that language selected see the org's default language values.

- The Manage Translation systems permission is available in permission sets.
- You must edit picklist values individually. You can't mass-edit existing picklist values, but you can still mass-add new values.
- When picklist values are sorted alphabetically, the values are alphabetical by the user's locale.
- Reports have a Filter Language dropdown list in the Filters pane of the report builder. Selecting a language filters on translated strings for any filter criteria that use the starts with, contains, or doesn't contain operator.
- Import files have a Language dropdown list, and all records and values within the import file must be in the selected language.
- Web-to-Lead and Web-to-Case have a Language dropdown list before you generate the HTML.

To disable Translation Workbench, from Setup, in the Quick Find box, enter *Translation Language Settings*, and then select **Translation Language Settings**. Click **Disable**.



Note In a Developer org with a managed package containing translations, Translation Workbench can't be disabled after it's enabled.

See Also

- [Supported Languages](#)
- [Language, Locale, and Currency Settings](#)
- [Rename Object, Tab, and Field Labels](#)

Add Translated Languages and Translators

Add languages for translation, assign translators for each language, and activate or deactivate a language's translations.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To add or edit languages: Manage Translation

To assign translators: Manage Translation



Note The Manage Translation permission is enabled by default in the System Administrator profile.

Before adding a language for translation, you must select languages for your org and enable Translation Workbench.

1. From Setup, in the Quick Find box, enter *Translation Language Settings*, and then select **Translation Language Settings**.
2. To activate a new language, click **Add**. Or to change an existing supported language, click **Edit**.
3. If adding a language, choose a language.
4. To make the entered translations available to your users, select **Active**. Users can change their personal language anytime, regardless of whether it's active in the Translation Workbench. Selecting **Active** makes the translations available to the users in that language.

We recommend that you don't make a language active until the translators have translated all values.



Note If you installed a managed package that includes translations, those translated values appear to users regardless of whether the language is active on the Translation Language Settings Setup page. To override metadata translations delivered by a managed package for custom objects, see [Override Translations in Second-Generation Managed Packages and Unlocked Packages](#).

5. To assign translators for this language, select them from the **Available List**, and click **Add**. If you don't see the member that you want to add, enter keywords in the search box, and click **Find**.
 6. Save your changes.
- Important** Ensure that all translators have the View Setup and Configuration permission so that they can begin translating. Users can only translate languages that they're assigned to.

See Also

[Select Languages for Your Org](#)

[Enable or Disable Translation Workbench](#)

[Override Translations in Second-Generation Managed Packages and Unlocked Packages](#)

Translate Metadata Labels

Create and update metadata translations for customizations you make to your Salesforce organization, such as custom picklist values and custom field labels.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To translate terms:

[View Setup and Configuration](#)

USER PERMISSIONS NEEDED

AND

Be designated as a translator

-  **Note** Entering translations through Translation Workbench has limitations, so note the following.

- Use the rename tabs and labels interface for standard object translation. Standard objects and custom object names aren't available in Translation Workbench.
- Manage data translations through the Translation tab within Product or through the Export and Import options in Translation Workbench. Only metadata translations are available for translation via the Translate Setup page.

Before translating metadata labels, you must select languages for your org, enable Translation Workbench, and add translated languages and translators.

1. From Setup, in the Quick Find box, enter *Translate*, and then select **Translate**.
2. Select the **Language** you're translating into.
3. Select a **Setup Component**. See Metadata Available for Translation for a list of translatable components.
4. Depending on the setup component, select the next options.

The aspect is a part of the setup component that you can translate. For example:For global value sets and picklist values, you can translate inactive values by selecting **Show Inactive Values**.

- Workflow tasks have an object (for example, Account or Contact) and aspect (Subject or Comment).
 - Custom Report Types have a custom report type entity (Custom Report Type, Custom Report Type Column, or Custom Report Type Layout Section) and aspect (field label or description).
 - Flows have a flow type (Flow and Autolaunched Flow), a flow name, and a flow component (Definition, Version, Screen Info, Screen Field, and Choice). Flow components can have a flow version, screen, or aspect.
5. To enter new values, double-click in the translation column. You can press Tab to advance to the next editable field or Shift+Tab to go to the previous editable field.

-  **Note** The **Out of Date** column indicates the possibility that the label needs translating because the primary label has been updated. When editing a button or link label, you see the **Button or Link Name** column, which is used to refer to the component when using SOAP API.

6. Click **Save**.

If a customized component doesn't have a translated value, the component uses the org's default language. When you deactivate a language, all translations for that language are still available in Translation Workbench. However, users with that language selected see the org's default language values.

See Also

[Select Languages for Your Org](#)[Translation Workbench](#)[Rename Object, Tab, and Field Labels](#)[Metadata Available for Translation](#)

Override Translations in Second-Generation Managed Packages and Unlocked Packages

You can override metadata translations for custom objects in namespaced unlocked packages and second-generation managed packages. For example, override the label on a custom field or workflow task.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To override metadata translations:

View Setup and Configuration

AND

Customize Application

 **Note** Overriding translations in second-generation managed packages and unlocked packages has limitations:

- You can't override translations for standard objects in packages.
- You can't override translations for global picklist value sets.
- You can't override data translations.

If you installed a managed package that includes translations, those translated values appear to users regardless of whether the language is active on the Translation Language Settings Setup page. Before you can override those translations, you must select languages for your org and enable Translation Workbench.

1. From Setup, in the Quick Find box, enter *Override*, and then select **Override**.
2. Select the **Package** that you're overriding.
3. Select the **Language** that you're entering your overrides in.

 **Note** The Language list shows the languages that meet these criteria:

- The language is in the package that's associated with this namespace.
- There is at least one translation for the language, or it's the package default language.

4. Select a **Setup Component**. See Metadata Available for Translation for a list of translatable

components.

5. Depending on the setup component, select the next options.

The aspect is a part of the setup component that you can translate. For example: For global value sets and picklist values, you can translate inactive values by selecting **Show Inactive Values**.

- Workflow tasks have an object (for example, Account or Contact) and aspect (Subject or Comment).
- Custom Report Types have a custom report type entity (Custom Report Type, Custom Report Type Column, or Custom Report Type Layout Section) and aspect (field label or description).
- Flows have a flow type (Flow and Autolaunched Flow), a flow name, and a flow component (Definition, Version, Screen Info, Screen Field, and Choice). Flow components can have a flow version, screen, or aspect.

6. To enter new values, double-click in the translation column. You can press TAB to advance to the next editable field or SHIFT-TAB to go to the previous editable field.

 **Note** The **Out of Date** column indicates the possibility that the term needs translation because the primary label has been updated. When editing a button or link label, you see the **Button or Link Name** column, which is used to refer to the component when using SOAP API.

7. Click **Save**.

See Also

- [Select Languages for Your Org](#)
- [Enable or Disable Translation Workbench](#)
- [Metadata Available for Translation](#)

Export Metadata Translation Files

To easily translate your metadata, create files for your translators that contain your Salesforce org's translatable metadata. Examples of translatable metadata include custom field labels, report type names, and picklist values.

REQUIRED EDITIONS

Before you can export metadata translation files, you must enable Translation Workbench.

Available in: Salesforce Classic and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To export or import translation files:

Manage Translation

AND

Create Documents

-  **Note** You need the Manage Translation AND Create Documents user permissions to import or export translation files. If you attempt either operation without both user permissions, it's possible to navigate to the import or export page, but the operation itself fails.

Exported files count toward your file storage limit. Before exporting, review the file storage limits for your org.

1. From Setup, in the Quick Find box, enter *Export*, and then select **Export**.
2. If data translation is enabled in your org, select the **Metadata** translation type.
3. Select the Salesforce Setup components and date filters that you want to export.

All Setup Components	Used as the default filter for creating translations. Includes all Salesforce Setup components in the translation files.
Selected Setup Components	Used to filter Salesforce Setup components. Add and remove available Salesforce Setup components to the translation files.
All Time	Used as the default date filter for creating translations. Includes text from all time.
Source Last Modified Date	Used as a date filter for creating translations. Includes text from a date range of when the source text was last modified. To include all records that were modified after the start date, omit the end date. To include all records that were modified before the end date, omit the start date.

4. Select the export type.

Source	Used as the initial source for creating translations. Creates a single file that contains a list of all your translatable customizations.
Outdated and untranslated	Used to make updates.

	<p>Creates a set of files that contain only metadata labels that haven't been translated, including new and modified labels.</p> <p>One file is created for each language. These files are then compressed into .zip files.</p>
Bilingual	<p>Used for reference and reviewing all your untranslated and translated customizations.</p> <p>Creates a list of all the translatable metadata labels in their current translated or untranslated state.</p> <p>One file is created for each language. These files are then compressed into .zip files.</p> <p>The content in each file is divided into Untranslated and Translated sections. Each translatable label is in the Untranslated or the Translated section, according to its translation state. The Translated section includes the out-of-date status for each label.</p>



Note Exported translation file content is in your org's default language.

5. If you selected **Outdated and untranslated** or **Bilingual**, select the languages to include in the output text file.
6. Select a format.
Salesforce recommends the XML Localization Interchange File Format (XLIFF) because it contains more information than the .stf format, such as the field width.
7. Click **Export**.
A status message tells you that the export is being processed. Wait for it to finish before you submit another export request. When the export is complete, an email is sent to the address specified in your profile.
8. Locate the exported .stf, .xlf, or .zip file. You can retrieve exported metadata translation files from either the My Personal Documents folder in Salesforce Classic or a link that's emailed to you.
 - In Salesforce Classic, go to **Your name | Documents | Document Folders | My Personal Documents | Go!**.
 - In the email sent to you, click the link.

The names of the exported files indicate the export option and include a timestamp. Individual files end with a .stf or .xlf extension. Multiple files are grouped into .zip files.

Find the exported files under the sort letter.

B	Bilingual export option, for example: Bilingual_2020-01-23_11:20.zip.
S	Source export option, for example: Source_en_US_2020-01-23_11:20.stf.
O	Outdated and untranslated export option, for example: Outdated_and_untranslated_2020-01-23_11:20.zip.

When you retrieve the files from My Personal Documents, each uncompressed file is limited to 5 MB. If your org's metadata translations exceed 5 MB, then the metadata is exported into multiple files, and each file name is timestamped and numbered. For example, Outdated_and_untranslated_2020-09-20_05:13_part_1_of_2.zip.

When you retrieve the files from the emailed link, each export request is limited to 1 GB of data and 100,000 records. Individual uncompressed files are limited to 25 MB each. If your org's metadata translations exceed 25 MB, the metadata is exported into multiple files, and each file name is date-stamped and numbered. For example, Bilingual_zh_CN_2025-05-08_1036_1.xlf, Bilingual_zh_CN_2025-05-08_1036_2.xlf.

9. Save the files for translation by your translators or translation agency. Click **View | Save File | OK**. The file is saved to the location specified by your browser. For example, C:/Users/username/Downloads.
10. Send the files to your translators or translation agency for bulk translation.

Use **Import** to update your labels.

See Also

- [Translation Workbench](#)
- [Gaps Between Salesforce Classic and Lightning Experience](#)
- [Salesforce Developer Doc: Translations](#)

Export Data Translation Files

If data translation is enabled in your Salesforce org, you can create files for your translators that contain your org's data translations. Examples of translatable data include B2B Commerce Product names and data stored in custom fields.

REQUIRED EDITIONS

Available in: Lightning Experience
Available in: Enterprise, Performance, and Developer Editions

USER PERMISSIONS NEEDED

To export or import translation files

Manage Translation

AND

Create Documents

-  **Note** You need the Manage Translation AND Create Documents user permissions to import or export translation files. If you attempt either operation without both user permissions, it's possible to navigate to the import or export page, but the operation itself fails.

Before you can export data translation files, you must enable Translation Workbench and data translation.

-  **Important** Each data translation export request is limited to 1 GB of data and 100,000 records. If your requested export exceeds either of those limits, only a partial file is exported. To reduce the amount of data exported, use the language filter, available for the Outdated and translated and Bilingual export types. If your org's data translations for a single object and language exceed either of those limits, use BULK API or [Data Loader](#).

1. From Setup, in the Quick Find box, enter *Export*, and then select **Export**.
2. Select **Data** as the Translation Type.
3. Select the objects for data translation.
4. Select which labels you want to export.
 - **Source**—Used as the initial source for creating translations.
Creates a set of files by object with all translatable text.
 - **Outdated and untranslated**—Used to make updates.
Creates a set of files, by language and then by object, including text changed after the last translation and text that isn't yet translated. These files are then compressed into .zip files.
 - **Bilingual**—Used for reference and reviewing all your untranslated and translated customizations.
Creates a set of files, by language and then by object, including all the translatable text in its current translated or untranslated state. These files are then compressed into .zip files.
The content in each file is divided into Untranslated and Translated sections. Each translatable text element is in the Untranslated or the Translated section, according to its translation state. The Translated section includes the out-of-date status for each text element.
5. If you selected the Outdated and untranslated or Bilingual export type, select at least one language.
6. Select a file format. Salesforce recommends the XML Localization Interchange File Format (XLIFF) because it contains more information than the .stf format, like the field width.
7. Click **Export**.

A status message tells you that the export is being processed. Wait for it to finish before submitting another export request. When the export is complete, an email is sent to the email address specified in your profile. The email includes a link to a .zip file with your exported translation files.



Note When exporting data translation files, individual uncompressed files are limited to 25 MB. If multiple files are required, each file name is date stamped and incremented. For example, `Bilingual_de_Product2_2020-10-20_0836_1.xlf` and `Bilingual_de_Product2_2020-10-20_0836_2.xlf`.

8. Send the files to your outside translators or translation agency for bulk translation activities, then use **Import** to update your data translations.

See Also

[Enable Data Translation](#)

[Translation Workbench](#)

[Salesforce Developer Doc: Translations](#)

Work with Translation Files

Translate metadata labels or data translation text, or review existing translations, with XML Localization Interchange File Format (.xlf) or Salesforce Translation Format (.stf) files.

REQUIRED EDITIONS

Metadata translation available in: Salesforce Classic and Lightning Experience

Data translation available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Data translation applies to: B2B Commerce

Translation files can contain metadata translations or data translations. Metadata translations are for customizations that you made to your Salesforce organization, such as custom picklist values and custom field labels. Data translations are for the data stored within fields, such as the text in the Product object's Name and Description fields.

For metadata and data translation files, there are three file types.

- Source: Use to translate labels for the first time.
- Outdated and untranslated: Use to translate labels after the first translation pass.
- Bilingual: Use to review and edit translations.

Translation files are identified with either the .xlf or .stf extension. Salesforce recommends the XML Localization Interchange File Format (.xlf) for translation files.

[Considerations for Working with Translation Files](#)

Review some important considerations to ensure that your edited translation file can be successfully imported.

[Source Translation Files](#)

Use the Source file to translate an organization's labels or data for the first time. The Source file

contains labels for all of a Salesforce org's translatable metadata or data in the org's default language.

Outdated and Untranslated Translation Files

Use the Outdated and untranslated file to translate labels or data that need translation. The file includes labels or data changed since the last translation and labels that haven't been translated. One Outdated and untranslated file is generated for each language. When multiple files are generated, they're exported to a .zip file containing a file for each translation language.

Bilingual Translation Files

Use the Bilingual file to review translations, edit existing translations, and add translations for labels or data that haven't been translated. One Bilingual file is generated for each translation language.

Translation File IDs and Keys

Each translatable item has a unique identifier in the translation file. In .xlf files, it's the id within a trans-unit tag. In .stf files, it's the key. The structure of these identifiers differs for metadata and data translation files.

Flow Identifiers in Translation Files

In a translation file exported from Translation Workbench, a unique key or trans-unit ID attribute identifies a flow metadata label.

See Also

[Export Metadata Translation Files](#)

[Export Data Translation Files](#)

[Import Translated Files](#)

Considerations for Working with Translation Files

Review some important considerations to ensure that your edited translation file can be successfully imported.

REQUIRED EDITIONS

Metadata translation available in: Salesforce Classic and Lightning Experience

Data translation available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Data translation applies to: B2B Commerce

General Considerations

- Translation files are identified with the .stf or .xlf extension. Don't change the file extension.
- Salesforce recommends the XML Localization Interchange File Format (.xlf) for translation files.
- All translation files are language-specific.
- A translation file name includes the name of the export option used to create it, the language code for the file's content, and a date stamp. Data translation file names also include the name of the object.
- Deleting a translation value, row, or `trans-unit` tag in the file doesn't remove the translation after

the file is imported. To delete a translation, replace the translated text with `<>` in an .stf file or `< >` in an .xlf file. When the file is imported, the label reverts to its primary label value. See Outdated and Untranslated Translation Files and Bilingual Translation Files in Salesforce Help for examples.



Note See the following Data Translation File Considerations for important notes about deleting the data translation for a record's Name field.

Exported Translation File Limits

- When exporting metadata translation files, individual uncompressed files are limited to 5 MB each. If your org's metadata translations exceed 5 MB, the system exports multiple files. If multiple .zip files are required, each file name is date-stamped and incremented. For example, `Outdated_and_untranslated_2020-09-20_05:13_part_1_of_2.zip`.
- Each data translation export request is limited to 1 GB of data and 100,000 records. If your requested export exceeds either of those limits, only a partial file is exported. To reduce the amount of data exported, use the language filter, available for the Outdated and translated and Bilingual export types. If your org's data translations for a single object and language exceed either of those limits, use BULK API or [Data Loader](#).
- When exporting data translation files, individual uncompressed files are limited to 25 MB. If multiple files are required, each file name is date stamped and incremented. For example, `Bilingual_de_Product2_2020-10-20_0836_1.xlf` and `Bilingual_de_Product2_2020-10-20_0836_2.xlf`.

Translation File Import Limits

- When importing metadata translation files, individual uncompressed files are limited to 10 MB each.
- If data translation isn't enabled in your org, each imported .zip file is limited to 10 MB. If data translation is enabled, each imported .zip file is limited to 1 GB.
- When importing data translation files, individual uncompressed files are limited to 50 MB each.
- When zipping data translation files for import, each .zip file can contain up to 100,000 total translation records within up to 2 GB of uncompressed files.

Data Translation File Considerations

A translated value for the data in the record's Name field is required to translate data in other fields for that record. For example, you must provide a German translation for the name of a Product before you can translate its description into German.

When translating text in a rich text area field, don't delete HTML tags such as `<p></p>` and ``. If you remove those tags, the returned value of `toLabel(rich_text_area_field)` in SOQL queries can be truncated to 255 characters.

There are two ways to delete the translated values for all fields related to a record. Because translation

files are language-specific, this action is per language.

 **Note** You can restore deleted data translation values through the recycle bin.

- Delete translated text for all of the record's fields. For example, a Product has German translated values for its name and description. To remove all German data translations for that Product, replace the translated text for that Product record's Name and Description fields with `<>` in an .stf file or `<,>` in an .xlf file.
- Delete the translated text for the record's Name field, and remove the rows or `trans-unit` tags for the record's other fields. For example, a Product has German translated values for its name and description. To remove all German data translations for that Product, replace the translated text for that Product record's Name fields with `<>` in an .stf file or `<,>` in an .xlf file. Also delete the row or `trans-unit` tag for that Product's Description field.

If an imported translation file deletes the translated value for a record's Name and includes a translated value for another field, no action is taken. For instance, you delete the translated value for a Product's name but leave the translation key for that Product's description unchanged in a translation file for German. When you import that translation file, no changes are made to that Product's translated values for German.

XML Localization Interchange File Format (.xlf) File Considerations

The .xlf format is the recommended export type because it contains more information than the .stf format, like the field width.

- .xlf translation file content is organized into translation units. Translation units for translated labels contain a `target` tag with the translated value. Untranslated labels have a `source` tag, but no `target` tag.
- Translators must create a `target` tag for each untranslated label to store the translated value. Otherwise, don't add tags to or remove tags from the .xlf file.
- The `file` tag's attributes define details about the translation file, such as the source language, target language, and the translation type.
- The `trans-unit` tags' attributes define details about the translation unit, such as the id and the field width.

Salesforce Translation Format (.stf) File Considerations

The .stf format behaves like a .csv file, with content organized into tab-delimited columns.

- If you don't use a standard translation tool such as Trados, edit the file using an application that supports tabs and word wrap, such as WordPad.
- Don't add columns to or remove columns from the translation file.
- If you use tabs, new lines, or carriage returns in your text for translation, they're represented with special characters in the .stf file format. Tabs are `\t`, new lines are `\n`, and carriage returns are `\r`.

To ensure consistency between your language versions, ensure that these characters are maintained in your translations.

- If you use Microsoft Excel to enter translations in an .stf file, your file format can be corrupted. MS Excel automatically adds quotation marks around entries that have commas. We recommend that you open your files in a text editor before importing them and remove these quotation marks. The import fails if these quotation marks aren't removed.

Translating Rich Text Fields

Fields with a type of Text Area (Rich) allow special formatting, such as bolding text or adding an image. When rich text fields are exported for translation, the translatable text for those fields includes encoded HTML tags.

If your translators edit raw translation files, make sure that they understand HTML tags and their encoding. When translating the text, HTML tags aren't required. However, if HTML tags are included in a translated value, they must be valid and in the same format as the exported tags. If the tags are incorrect, the translation import fails.

 **Note** The HTML tags are visible in the raw files. Many translation tools handle the conversion of markup languages like HTML for you.

All rich text field translations must be contained within an HTML paragraph (`<p>`) tag with the appropriate HTML encoding for that file type. If the translation value contains only plain text, the required paragraph tag is added upon import.

This table provides examples of exported rich text field content.

Rich Text Field Content	Format	Exported Text
Available in women's shoe sizes 5–13.	.stf	<code><p>Available in women's shoe sizes 5-13.</p></code>
	.xlf	<code>&lt;p&gt;Available in women's shoe sizes 5-13.&lt;/p&gt;</code>
15% discount available for veterans. Verification required.	.stf	<code><p>15% discount available for veterans.</p><p>Verification required.</p></code>
	.xlf	<code>&lt;p&gt;15% discount available for veterans.&lt;/p&gt;&lt;p&gt;&lt;b&gt;Verification required.&lt;/b&gt;&lt;/p&gt;</code>
Features: <ul style="list-style-type: none">LeatherImported	.stf	<code><p>Features:LeatherImportedRubber soleLightweightFlexible sole</p></code>
	.xlf	<code>&lt;p&gt;Features: &lt;ul&gt;&lt;li&gt;Leather</code>

Rich Text Field Content	Format	Exported Text
<ul style="list-style-type: none"> • Rubber sole • Lightweight • Flexible sole 		Imported Rubber sole Lightweight Flexible sole
See more information at this reference site .	.stf	<p>See more information at this reference site.</p>
	.xlf	<p><p>this reference site</p>*</p>

See Also

[Export Metadata Translation Files](#)

[Export Data Translation Files](#)

[Import Translated Files](#)

[View, Restore, and Manage the Recycle Bin in Salesforce Classic](#)

Source Translation Files

Use the Source file to translate an organization's labels or data for the first time. The Source file contains labels for all of a Salesforce org's translatable metadata or data in the org's default language.

REQUIRED EDITIONS

Metadata translation available in: Salesforce Classic and Lightning Experience

Data translation available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Data translation applies to: B2B Commerce

 **Note** Salesforce recommends the XML Localization Interchange File Format (.xlf) for translation files. See Considerations for Working with Translation Files for tips on editing translation files and how to translate rich text field content.

To prepare the translation file for your translators:

- Create one copy of the Source file for each language you're translating into.
- In the header of each Source file, change the language code from the organization's default language to the translation language. For example, replace `en_US` for English (US) with `es` for Spanish.
- Ensure that the headers are in the organization's default language. To confirm, export a translation file

and copy the headers.

XML Localization Interchange File Format (.xlf) Source Translation Files

Source .xlf translation file content is organized into translation units. Translation units for translated labels contain a `target` tag with the translated value. Untranslated labels have a `source` tag, but no `target` tag.

Tell your translators:

- After each `source` tag, add a `target` tag that contains the translated value.
- If a `target` tag exists and the translation is out of date, replace the text in the target tag. Outdated labels have a value of `outOfDate="true"` within the trans-unit tag.
- When translating text in a rich text area field, include all HTML tags such as `<p></p>` and ``.

! **Important** A translated value for the data in the record's Name field is required to translate data in other fields for that record. For example, you must provide a German translation for the name of a Product before you can translate its description into German.

Tag	Description	Edit Options
<code>trans-unit</code>	Translation unit. Contains unique identifiers for the label, including the label's id, maximum width, and out-of-date indicator.	Do not edit.
<code>source</code>	Label or text in the org's default language.	Do not edit.
<code>target</code>	The current translation that's visible to end users selecting the target language as their personal language.	Enter the translated value. Add a target tag if needed.
<code>note</code>	Description of the metadata label, if defined in the source language.	Do not edit. Translatable field descriptions each have a separate <code>trans-unit</code> tag.

For example, if you build a custom Nickname field on the Account object, the original file contains this `trans-unit` tag in the exported Source .xlf file.

```
<trans-unit id="CustomField.Account.Nickname.FieldLabel" maxwidth="40" size-unit="char">
    <source>Nickname</source>
    <note>The person's nickname, or what they prefer to be called.</note>
</trans-unit>
```

To translate this label, add a `target` tag containing the translated value to the corresponding `trans-unit` tag after the `source` tag.

```
<trans-unit id="CustomField.Account.Nickname.FieldLabel" maxwidth="40" size-unit="char">
    <source>Nickname</source>
    <target>Apodo</target>
    <note>The person's nickname, or what they prefer to be called.</note>
</trans-unit>
```

Salesforce Translation Format (.stf) Source Translation Files

 **Note** Salesforce doesn't recommend the STF format for translation files. If you choose to use this format, we don't recommend editing the file with Microsoft Excel. For more information and restrictions, see Considerations for Working with Translation Files.

Tell your translators to replace the untranslated values in the LABEL column with translated values.

 **Important** A translated value for the data in the record's Name field is required to translate data in other fields for that record. For example, you must provide a German translation for the name of a Product before you can translate its description into German.

Column	Description	Edit Options
KEY	Unique identifier for the label.	Do not edit.
LABEL	Label or text in the org's default language.	Replace untranslated values with translated values.

For example, if you build a custom Nickname field on the Account object, the original file contains this row in the exported Source .stf file.

# KEY	LABEL
CustomField.Account.Nickname.FieldLabel	Nickname
1	

To translate this label, replace the LABEL text in that row with the translated value.

# KEY	LABEL
CustomField.Account.Nickname.FieldLabel	Apodo
1	

 **Important** Don't add columns to or remove columns from the .stf translation file.

See Also

- [Considerations for Working with Translation Files](#)
- [Supported Languages](#)

Outdated and Untranslated Translation Files

Use the Outdated and untranslated file to translate labels or data that need translation. The file includes labels or data changed since the last translation and labels that haven't been translated. One Outdated and untranslated file is generated for each language. When multiple files are generated, they're exported to a .zip file containing a file for each translation language.

REQUIRED EDITIONS

Metadata translation available in: Salesforce Classic and Lightning Experience

Data translation available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Data translation applies to: B2B Commerce

 **Note** Salesforce recommends the XML Localization Interchange File Format (.xlf) for translation files. See Considerations for Working with Translation Files for tips on editing translation files and how to translate rich text field content.

When working with the Outdated and untranslated translation file, each entry needs a new translated value. Special values are used to delete an existing translation and revert the label to its original value.

XML Localization Interchange File Format (.xlf) Outdated and Untranslated Translation Files

Outdated and untranslated .xlf translation file content is organized into translation units. Translation units for translated labels contain a `target` tag with the translated value. Untranslated labels have a `source` tag, but no `target` tag.

Tell your translators:

- For untranslated labels, add a `target` tag containing the translated value after each `source` tag.
- If the label's translation is out of date, replace the text in the target tag. Outdated labels have a value of `outOfDate="true"` within the trans-unit tag. Don't update the `outOfDate` value.
- To delete a translation, replace the value in the label's target tag with `<>`. When the Bilingual file is imported, the label reverts to its original value.
- A translated value for the data in the record's Name field is required to translate data in other fields for that record. For example, you must provide a German translation for the name of a Product before you

- can translate its description into German.
- Deleting the data translated value for a record's Name field can delete all of that record's other translated values for that language. See Considerations for Working with Translation Files for more information.
 - When translating text in a rich text area field, don't delete HTML tags such as `<p></p>` and ``. If you remove those tags, the translated text can be truncated.

Tag	Description	Edit Options
<code>trans-unit</code>	Translation unit. Contains unique identifiers for the label, including the label's id, maximum width, and out-of-date indicator.	Do not edit.
<code>source</code>	Label or text in the org's default language.	Do not edit.
<code>target</code>	The current translation that is visible to end users selecting the target language as their personal language.	Enter the translated value. Add a target tag if needed.
<code>note</code>	Description of the metadata label, if defined, in the source language.	Do not edit. Translatable field descriptions each have a separate <code>trans-unit</code> tag.

For example, an existing Nickname field on the Account object has a Spanish translated value of "Apodo." You change the primary label on the Nickname field from "Nickname" to "Preferred Name." That label is now outdated. You also build a new custom Business Hours field on the Account object. The exported Outdated and untranslated .xlf translation file contains the following `trans-unit` tags.

```

<trans-unit id="CustomField.Account.Nickname.FieldLabel" maxwidth="40" size-unit="char" outOfDate="true">
    <source>Preferred Name</source>
    <target>Apodo</target>
    <note>The name preferred by this person.</note>
</trans-unit>
<trans-unit id="CustomField.Account.Business_Hours.FieldLabel" maxwidth="40" size-unit="char" outOfDate="false">
    <source>Business Hours</source>
</trans-unit>

```

To update the outdated label, update the text in the corresponding `target` tag. To translate the new field's label, add a `target` tag containing the translated value to the corresponding `trans-unit` tag after the `source` tag. Don't change the `outOfDate` tag values. When you import the translated file, labels with updated translations are marked as up to date.

```

<trans-unit id="CustomLabel.Account.Nickname.FieldLabel" maxwidth="40" size-unit="char" outOfDate="true">
    <source>Preferred Name</source>
    <target>Nombre preferido</target>
    <note>The name preferred by this person.</note>
</trans-unit>
<trans-unit id="CustomLabel.Account.Business_Hours.FieldLabel" maxwidth="40" size-unit="char" outOfDate="false">
    <source>Business Hours</source>
    <target>Horario de oficina</target>
</trans-unit>

```

To delete the Preferred Name label's translation, update the translation value in the target tag with `<>`. When the file is imported, the label reverts to its original value.

```

<trans-unit id="CustomLabel.Account.Nickname.FieldLabel" maxwidth="40" size-unit="char" outOfDate="true">
    <source>Preferred Name</source>
    <target>&lt;&gt;</target>
    <note>The name preferred by this person.</note>
</trans-unit>

```

Salesforce Translation Format (.stf) Outdated and Untranslated Translation Files

-  **Note** Salesforce doesn't recommend the STF format for translation files. If you choose to use this format, we don't recommend editing the file with Microsoft Excel. For more information and restrictions, see Considerations for Working with Translation Files.

Tell your translators to replace the values in the LABEL column with updated translated values.

- To delete a translation, replace the desired value in the TRANSLATION column with a left and right angle bracket pair (<>). When the file is imported, the label reverts to its original value.
- A translated value for the data in the record's Name field is required to translate data in other fields for that record. For example, you must provide a German translation for the name of a Product before you can translate its description into German.
- Deleting the data translated value for a record's Name field can delete all of that record's other translated values for that language. See Considerations for Working with Translation Files for more information.

Column	Description	Edit Options
KEY	Unique identifier for the label.	Do not edit.

Column	Description	Edit Options
LABEL	Label or text in the org's default language.	Replace untranslated values with translated values.

For example, an existing custom Nickname field on the Account object has a Spanish translated value of “Apodo.” You change the primary label on the Nickname field from “Nickname” to “Preferred Name.” That label is now outdated. You also build a new custom Business Hours field on the Account object. The exported Outdated and untranslated .stf translation file contains these rows.

# KEY	LABEL
CustomField.Account.Nickname.FieldLabel	Preferred Name
CustomField.Account.Business_Hours.FieldLabel	Business Hours

To translate the new field’s label and update the existing label, replace the LABEL text in each row.

# KEY	LABEL
CustomField.Account.Nickname.FieldLabel	Nombre preferido
CustomField.Account.Business_Hours.FieldLabel	Horario de oficina

To delete the Nickname field’s outdated translation of “Apodo,” replace the translation value in the LABEL column with `<>`. When the file is imported, the label reverts to its primary label value of “Preferred Name.”

# KEY	LABEL
CustomField.Account.Nickname.FieldLabel	<>

See Also

[Considerations for Working with Translation Files](#)

Bilingual Translation Files

Use the Bilingual file to review translations, edit existing translations, and add translations for labels or data that haven’t been translated. One Bilingual file is generated for each translation language.

REQUIRED EDITIONS

Metadata translation available in: Salesforce Classic and Lightning Experience

Data translation available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Data translation applies to: B2B Commerce

-  **Note** Salesforce recommends the XML Localization Interchange File Format (.xlf) for translation files. See Considerations for Working with Translation Files for tips on editing translation files and how to translate rich text field content.

XML Localization Interchange File Format (.xlf) Bilingual Translation Files

Bilingual .xlf translation file content is organized into translation units. Translation units for translated labels contain a `target` tag with the translated value. Untranslated labels have a `source` tag, but no `target` tag.

Tell your translators:

- For untranslated labels, add a `target` tag containing the translated value after the `source` tag.
- If the label's translation is out of date, replace the text in the `target` tag. Outdated labels have a value of `outOfDate="true"` within the `trans-unit` tag.
- To delete a translation, replace the value in the `trans-unit`'s `target` tag with `<>`. When the Bilingual file is imported, the label reverts to its primary label value.
- A translated value for the data in the record's Name field is required to translate data in other fields for that record. For example, you must provide a German translation for the name of a Product before you can translate its description into German.
- Deleting the data translated value for a record's Name field can delete all of that record's other translated values for that language. See Considerations for Working with Translation Files for more information.
- When translating text in a rich text area field, don't delete HTML tags such as `<p></p>` and ``. If you remove those tags, the translated text can be truncated.

Tag	Description	Edit Options
<code>trans-unit</code>	Translation unit. Contains unique identifiers for the label, including the label's id, maximum width, and out-of-date indicator.	Do not edit.
<code>source</code>	Label or text in the org's default language.	Do not edit.
<code>target</code>	The current translation that is	Enter the translated value. Add a

Tag	Description	Edit Options
	visible to end users selecting the target language as their personal language.	target tag if needed. Replace a value with <code>&lt;&gt;</code> to delete the translation.
<code>note</code>	Description of the metadata label, if defined, in the source language.	Do not edit. Translatable field descriptions each have a separate <code>trans-unit</code> tag.

For example, in an org with English as its default language, you build a new custom Business Hours field on the Account object. This label is untranslated.

Nickname, an existing custom field on the Account object, has a Spanish translated value of “Apodo.” You change the primary label on the Nickname field from “Nickname” to “Preferred Name.” This label is outdated.

Another existing custom Prior Reference Number field on the Account object has an incorrect Spanish translated value of “Número de consulta previa.” Although the translation isn’t out of date, it must be updated to “Número de referencia precedente.”

The Name field on a custom Widget object had a primary label of “SLK.” A translator misinterpreted this acronym and entered a Spanish translation of “Flojo.” Although the translation isn’t out of date, you want to revert the translation to the primary label.

Finally, a custom Number field on a custom Widget object had a primary label of “Number” and a Spanish translated value of “Número.” You update the primary label to “#” and want to remove the translated value.

The exported Bilingual .xlf translation file contains the following row in the OUTDATED AND UNTRANSLATED section.

```

<trans-unit id="CustomField.Account.Business_Hours.FieldLabel" maxwidth="40" size-unit="char" outOfDate="false">
    <source>Business Hours</source>
</trans-unit>
<trans-unit id="CustomField.Account.Nickname.FieldLabel" maxwidth="40" size-unit="char" outOfDate="true">
    <source>Preferred Name</source>
    <target>Apodo</target>
    <note>The name preferred by this person.</note>
</trans-unit>
<trans-unit id="CustomField.Account.Prior_Ref_No.FieldLabel" maxwidth="20" size-unit="char" outOfDate="false">
    <source>Prior Reference Number</source>
    <target>Número de consulta previa</target>

```

```

</trans-unit>
<trans-unit id="CustomField.Widget__c.Name.FieldLabel" maxwidth="20" size-unit="char" outOfDate="false">
    <source>SLK</source>
    <target>Seda</target>
</trans-unit>
<trans-unit id="CustomField.Widget__c.Number.FieldLabel" maxwidth="20" size-unit="char" outOfDate="true">
    <source>#</source>
    <target>Número</target>
</trans-unit>

```

To make the requested changes:

- Add a `target` tag with the translated value to the new Business Hours field.
- Update the translation values in the target tags for the Preferred Name and Prior Reference Number fields.
- To delete the translation of the Widget object's Name and Number fields, update the translation values in those target tags with `< >`. When this file is imported, those labels revert to the primary label values.
- Don't change the `outOfDate` tag values. When you import the translated file, labels with updated translations are marked as up to date.

```

<trans-unit id="CustomField.Account.Business_Hours.FieldLabel" maxwidth="40" size-unit="char" outOfDate="false">
    <source>Business Hours</source>
    <target>Horario de oficina</target>
</trans-unit>
<trans-unit id="CustomField.Account.Nickname.FieldLabel" maxwidth="40" size-unit="char" outOfDate="true">
    <source>Preferred Name</source>
    <target>Nombre preferido</target>
    <note>The name preferred by this person.</note>
</trans-unit>
<trans-unit id="CustomField.Account.Prior_Ref_No.FieldLabel" maxwidth="20" size-unit="char" outOfDate="false">
    <source>Prior Reference Number</source>
    <target>Número de referencia precedente</target>
</trans-unit>
<trans-unit id="CustomField.Widget__c.Name.FieldLabel" maxwidth="20" size-unit="char" outOfDate="false">
    <source>SLK</source>
    <target>&lt; &gt;</target>
</trans-unit>

```

```
<trans-unit id="CustomField.Widget__c.Number.FieldLabel" maxwidth="20" size-unit="char" outOfDate="true">
    <source>#</source>
    <target>&lt; &gt;</target>
</trans-unit>
```

Salesforce Translation Format (.stf) Bilingual Translation Files

-  **Note** Salesforce doesn't recommend the STF format for translation files. If you choose to use this format, we don't recommend editing the file with Microsoft Excel. For more information and restrictions, see Considerations for Working with Translation Files.

Bilingual .stf translation file content is separated into translated labels and outdated or untranslated labels.

The TRANSLATED section of the .stf file contains text that has been translated and is up to date. When importing, four columns are expected for each label in this section: KEY, LABEL, TRANSLATION, and OUT OF DATE.

For this section, tell your translators:

- To update a current translation, replace the value in the TRANSLATION column.
- To delete a translation, replace the value in the TRANSLATION column with a left and right angle bracket pair (<>). When the file is imported, the label reverts to its primary label's value.

Column	Description	Edit Options
KEY	Unique identifier for the label.	Do not edit.
LABEL	Label or text in the org's default language.	Do not edit.
TRANSLATION	The current translation that is visible to end users selecting the target language as their personal language.	<ul style="list-style-type: none"> • To edit a translation, replace the translated value. • To delete a translation, replace the translated value with <> .
OUT OF DATE	Indicates whether the source text has changed since the previous translation. A dash (-) indicates that the translation is current.	Do not edit.

The OUTDATED AND UNTRANSLATED section of the file contains labels changed after the label's translation value was last updated and text that hasn't been translated. When importing, two columns are expected for each label in this section: KEY and LABEL.

For this section, tell your translators:

- Replace the text in the LABEL column with new or updated translation values.
- Delete any values in the TRANSLATED and OUT OF DATE columns.
- Delete the corresponding columns in the OUTDATED AND UNTRANSLATED section.
- To delete an outdated translation, replace the value in the LABEL column with a left and right angle bracket pair (<>). When the file is imported, the label reverts to its primary label's value.
- A translated value for the data in the record's Name field is required to translate data in other fields for that record. For example, you must provide a German translation for the name of a Product before you can translate its description into German.
- Deleting the data translated value for a record's Name field can delete all of that record's other translated values for that language. See Considerations for Working with Translation Files for more information.

Column	Description	Edit Options
KEY	Unique identifier for the label.	Do not edit.
LABEL	Label or text in the org's default language.	Replace label text with new or updated translated values.
TRANSLATION	<p>The current translation that is visible to end users selecting the target language as their personal language.</p> <p>Untranslated labels don't have a value in this column.</p>	Delete this column and its contents when updating an out-of-date translation.
OUT OF DATE	<p>Indicates whether the source text has changed since the previous translation.</p> <p>An asterisk (*) indicates that the label is out of date. A change was made to the primary label and the translation hasn't been updated.</p> <p>Untranslated labels don't have a value in this column.</p>	Delete this column and its contents when updating an out-of-date translation.

For example, in an org with English as its default language, an existing custom Prior Reference Number field on the Account object has an incorrect Spanish translated value of “Número de consulta previa.” Although the translation isn’t out of date, it must be updated to “Número de referencia precedente.”

The Name field on a custom Widget object had a primary label of “SLK.” A translator misinterpreted this acronym and entered a Spanish translation of “Flojo.” Although the translation isn’t out of date, you want to revert the translation to the primary label.

You also build a new custom Business Hours field on the Account object. This label is untranslated.

Nickname, another existing custom field on the Account object, has a Spanish translated value of “Apodo.” You change the primary label on the Nickname field from “Nickname” to “Preferred Name.” This label is outdated.

Finally, a custom Number field on a custom Widget object had a primary label of “Number” and a Spanish translated value of “Número.” You update the primary label to “#” and want to remove the translated value.

The exported Bilingual .stf translation file contains these rows in the OUTDATED AND UNTRANSLATED section.

-----TRANSLATED-----			
# KEY	LABEL	TRANSLATION	OUT OF DATE
CustomField.Account.Prior_Ref_No.FieldLabel	Prior Reference Number	Número de consulta previa	-
CustomField.Widget__c.Name.FieldLabel	SLK	Seda	-
-----OUTDATED AND UNTRANSLATED-----			
# KEY	LABEL	TRANSLATION	OUT OF DATE
CustomField.Account.Business_Hours.FieldLabel	Business Hours		
CustomField.Account.Nickname.FieldLabel	Preferred Name	Apodo	*
CustomField.Widget__c.Number.FieldLabel	#	Número	*

To make the requested changes:

- In the TRANSLATED section, update translations by replacing the value in the TRANSLATION column.

- To delete the translation of the Widget object's Name field, replace the TRANSLATION value with <> . When this file is imported, the label for that Number field reverts to its primary label value of SLK.
- In the OUTDATED AND UNTRANSLATED section, replace the value in the LABEL column. Then remove the TRANSLATION and OUT OF DATE columns and their content for the Nickname field. To delete the translation of the Widget object's Number field, replace the TRANSLATION value with <> . When this file is imported, the label for that Number field reverts to its primary label value of #.

----- TRANSLATED -----			
# KEY	LABEL	TRANSLATION	OUT OF DATE
CustomField.Account.Prior_Ref_No.FieldLabel	Prior Reference Number	Número de referencia precedente	-
CustomField.Widget__c.Name.FieldLabel	SLK	<>	-
----- OUTDATED AND UNTRANSLATED -----			
# KEY	LABEL		
CustomField.Account.Business_Hours.FieldLabel	Horario de oficina		
CustomField.Account.Nickname.FieldLabel	Nombre preferido		
CustomField.Widget__c.Number.FieldLabel	<>		

! **Important** Delete the TRANSLATION and OUT OF DATE columns only in the OUTDATED AND UNTRANSLATED section. Rows in that section must have exactly two columns of data to be imported. Rows in the TRANSLATED section must have exactly four columns of data to be imported.

See Also

[Considerations for Working with Translation Files](#)

Translation File IDs and Keys

Each translatable item has a unique identifier in the translation file. In .xlf files, it's the id within a trans-unit tag. In .stf files, it's the key. The structure of these identifiers differs for metadata and data translation files.

REQUIRED EDITIONS

Metadata translation available in: Salesforce Classic and Lightning Experience

Data translation available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Data translation applies to: B2B Commerce

-  **Note** Translation file keys and ids aren't edited during translation. The text to be translated is in the translation file in the `source` tag for .xlf files or the label column for .stf files. For admins, knowing how the id or key work can help them understand how these identifiers are structured.

Metadata Translation File IDs and Keys

All translatable metadata has a setup component. Depending on the setup component, the translatable metadata can have an object, aspect, custom report type entity, flow type, flow name, and flow component.

- Metadata label translation file IDs and keys follow the format:
`SetupComponentName.ObjectName.AspectName`.
- Metadata label translation file IDs and keys for custom report types contain the report type entity in place of the object name. The format is:
`SetupComponentName.CustomReportEntityName.AspectName`.
- Metadata translation file IDs and keys for flows can also contain the flow component, flow type, flow name, flow version, flow screen, or flow aspect. See [Flow Identifiers in Translation Files](#) for detailed examples.

This table provides examples of metadata translation IDs and keys.

Translation Label or Text	id (in .xlf Files) or Key (in .stf Files)
Country “Spain” in address	<code>AddressCountry.ES</code>
“Billing” button or link on the Account object	<code>ButtonOrLink.Account.Billing</code>
Label of the “Active” field on the Account object	<code>CustomField.Account.Active.FieldLabel</code>
Description of the “Active” field on the Account object	<code>CustomField.Account.Active.Description.FieldLabel</code>
Help text of the “Active” field on the Account object	<code>CustomField.Account.Active.Description.HelpText</code>
“Low” picklist entry in the Customer Priority picklist on the Account object	<code>PicklistValue.Account.CustomerPriority.Low</code>
Description of the “East Accounts” Custom Report Type	<code>CustomReportType.East_Accounts.Description</code>
“Accounts” Custom Report Type Layout Section on	<code>CrtLayoutSection.Custom_Accounts_Report</code>

Translation Label or Text	id (in .xlf Files) or Key (in .stf Files)
the “Accounts” Custom Report Type	ts_1
Description of the “Goal Layout” Custom Report Type Layout on the “Goals” Custom Report Type	LayoutSection.Goal.Goal Layout.Description_1
“Survey Customers” flow name	Flow.Flow.Survey_customers.FieldLabel
“Customer Name” screen input field on version 2 of the “Survey Customer” flow	Flow.Flow.Survey_customers.2.Survey_Customer.Field.Customer_Name. FieldLabel

Data Translation File IDs and keys

If data translation is enabled in your Salesforce org, the record ID is used to identify the translatable text in an exported data translation file. Data translation IDs and keys follow the format ObjectName.recordUniqueIdentifier.FieldName.

This table provides examples of data translation IDs and keys.

Translation Label or Text	id (in .xlf Files) or Key (in .stf Files)
Name of the product with record ID 01txx0000006yvEAAQ	Product2.01txx0000006yvEAAQ.Name
Description of the product with record ID 01txx0000006yvEAAQ	Product2.01txx0000006yvEAAQ.Description
Text stored in a custom “Discount Notes” field for the product record ID 01txx0000006yvEAAQ	Product2.01txx0000006yvEAAQ.Discount_Notes__c

See Also

[Flow Components for Metadata Translation](#)

Flow Identifiers in Translation Files

In a translation file exported from Translation Workbench, a unique key or trans-unit ID attribute identifies a flow metadata label.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Flows follow a convention based on the flow label.

Flow Component	Key or Trans-Unit ID	Example
Flow Definition Name	Flow.flowType.flowUniqueName.FieldLabel	Flow.Flow.Survey_customers.FieldLabel
Flow Version Name	Flow.flowType.flowUniqueName.versionNumber.Name	Flow.Flow.Survey_customers.1.Name
Screen		
Paused Message	Flow.flowType.flowUniqueName.versionNumber.Screen.screenUniqueName.PausedText	Flow.Flow.Survey_customers.1.Screen.Greet_Customer.PausedText
Help Text	Flow.flowType.flowUniqueName.versionNumber.Screen.screenUniqueName.HelpText	Flow.Flow.Survey_customers.1.Screen.Greet_Customer.HelpText
Screen Input Field		
Label	Flow.flowType.flowUniqueName.versionNumber.screenUniqueName.Field.fieldUniqueName.FieldLabel	Flow.Flow.Survey_customers.2.Survey_Customer.Field.Customer_Name.FieldLabel
Help Text	Flow.flowType.flowUniqueName.versionNumber.screenUniqueName.Field.fieldUniqueName.HelpText	Flow.Flow.Survey_customers.2.Survey_Customer.Field.Customer_Name.HelpText
Error Message	Flow.flowType.flowUniqueName.versionNumber.screenUniqueName.Field.fieldUniqueName.ErrorMessage	Flow.Flow.Survey_customers.2.Survey_Customer.Field.Customer_Name.ErrorMessage
Screen Output Field		
Display Text	Flow.flowType.flowUniqueName.versionNumber.screenUniqueName.Field.fieldUniqueName.Description	Flow.Flow.Survey_customers.2.Greet_Customer.Field.WelcomeMessage.Description
Choice		
Label	Flow.flowType.flowUniqueName.versionNumber.Choice.choiceUniqueName.FieldLabel	Flow.Flow.Survey_customers.2.Choice.Participate_No.FieldLabel
Error Message	Flow.flowType.flowUniqueName.versionNumber.Choice.choiceUniqueName.ErrorMessage	Flow.Flow.Survey_customers.2.Choice.Participate_No.ErrorMessage
Input Label	Flow.flowType.flowUniqueName.versionNumber.Choice.	Flow.Flow.Survey_customers.2.Choice.Participate_No.InputLabel

Flow Component	Key or Trans-Unit ID	Example
	<code>choiceUniqueName.InputLabel</code>	
Stages		
Stage Label	<code>Flow.flowType.flowUniqueName.versionNumber.Stage.stageUniqueName.FieldLabel</code>	<code>Flow.Flow.Stages_Online_Purchase_Breadcrumbs.1.Stage.Billing_Details.FieldLabel</code>
Text Template		
Label	<code>Flow.flowType.flowUniqueName.versionNumber.TextTemplate.texttemplateUniqueName.FieldLabel</code>	<code>Flow.Flow.Survey_customers.2.TextTemplate.ParticipantCity_FieldLabel</code>

See Also

[Work with Translation Files](#)

[Considerations for Translating Flows](#)

Import Translated Files

Import and update the translations for your Salesforce org's metadata, such as custom fields, report types, and picklist values. Or import and update data translations, such as Product names. Typically, you export translation files from Salesforce, then send them to outside translators or a translation agency for bulk translation activities. You then import the translated files.

REQUIRED EDITIONS

Metadata translation available in: Salesforce Classic and Lightning Experience

Data translation available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Data translation applies to: B2B Commerce

USER PERMISSIONS NEEDED

To export or import translation files

Manage Translation

AND

Create Documents



Note You need the Manage Translation AND Create Documents user permissions to import or export translation files. If you attempt either operation without both user permissions, it's possible

to navigate to the import or export page, but the operation itself fails.

Prepare Your Translated Files:

1. Create a separate file for each language.

Salesforce recommends the XML Localization Interchange File Format (.xlf) for translation files.

2. Specify the language for this translation import.

When you export a translation file with the Bilingual or Outdated and untranslated export type, the language is already specified. If you're importing a Source file or importing translations for a different language than the original file, update the language code.

- For .xlf files, use the `target-language` attribute on the file tag. For example, `<file original="Salesforce" source-language="en_US" target-language="es" datatype="xml">`.
- For .stf files, use the language code attribute at the top of the file. For example:

```
# Language: Spanish
Language code: es
Type: Source
```

Translation for the specified language must be supported for your org. For a full list of Salesforce supported languages and their language codes, see [Supported Languages in Salesforce Help](#).

3. If data translation is enabled in your Salesforce org, include a translation type attribute. For .xlf files, include the `translation-type` attribute on the file tag. For .stf files, include the translation type on its own line after the language code and type.

 **Note** If you export a translation file after data translation is enabled in your org, the resulting file includes this attribute.

File Type	Metadata Translation Attribute	Data Translation Attribute
XML Localization Interchange File Format (.xlf)	<code>translation-type="metadata"</code>	<code>translation-type="data"</code>
Salesforce Translation Format (.stf)	<code>Translation Type: Metadata</code>	<code>Translation Type: Data</code>

4. Save import files in UTF-8 encoding.
5. Size your files.

When importing metadata translation files, individual uncompressed files are limited to 10 MB each.
When importing data translation files, individual uncompressed files are limited to 50 MB each.

6. Bundle multiple files into .zip files.

If data translation isn't enabled in your org, each imported .zip file is limited to 10 MB. If data translation is enabled, each imported .zip file is limited to 1 GB. When zipping data translation files for import, each .zip file can contain up to 100,000 total translation records within up to 2 GB of uncompressed files. Create multiple .zip files as needed.

 **Important** Each .zip file can only contain metadata or data translation. If data translation is enabled in your org, import metadata and data translations separately.

The zipped files don't have to be in the same order or grouping as the exported .zip files.

For example, you start with two exported .zip files. The first file includes French, Italian, and Japanese. The second file includes Russian, Simplified Chinese, and Greek. You can create:

- One .zip file with French, Greek, and Italian.
- One .zip file with Russian and Greek.
- One .zip file with Simplified Chinese.

Import Your Prepared Translated Files:

1. From Setup, in the Quick Find box, enter *Import*, and then select **Import** under Translation Workbench.
2. Click **Choose File**, and select the file you want to import.
3. Click **Import**.

After the import is complete, a confirmation email is sent to the email address specified in your profile. Wait for each import to finish before submitting another translation file for import.

If any portion of the import fails, the email includes details about what went wrong. If the imported zip file exceeds translation import limits, the email lists the files that were imported before the limit was reached. If records within the imported translation files weren't processed, the email lists those files and includes a link to an error log.

Verify Your Translations:

There are multiple ways to view the imported translations:

 **Note** Labels that are exported and left unchanged in the translation file aren't saved as translations on import.

- Check metadata labels and data translations in your Salesforce org.
- Check metadata labels through Translation Workbench.
- Check data translations through the Translation tab within Product.

- Export data translations and verify your updated text. Check the labels in your Salesforce org.

See Also

[Supported Languages](#)
[Work with Translation Files](#)
[Common Errors with Exporting and Importing Translation Files](#)
[Documents Home](#)

Common Errors with Exporting and Importing Translation Files

Troubleshoot issues that you can encounter when exporting and importing files in Translation Workbench.

REQUIRED EDITIONS

Metadata translation available in: Salesforce Classic and Lightning Experience

Data translation available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Data translation applies to: B2B Commerce

Error Message	What It Means	Troubleshooting Steps
Bilingual file section starts with non-header row: <line number>	The header rows of the file that you're trying to import are missing. Or there's extraneous text in those rows, such as notes that aren't commented out.	Export your file again. Make sure that there are header rows for all sections and that all extraneous text has been commented out or removed from the header rows.
Data translation import for key <key> failed. A translated value for the <object name> <unique record ID> record ID's Name field is required to translate data for the <field name> field.	This error message only applies to data translation files. A translated value for a record ID's Name field is required to translate data in other fields for that record ID.	Provide a translated value for the data stored in the Name field for that record ID, and then import again.
Data translation isn't enabled for this org.	This error message only applies to data translation files. Data translation isn't enabled or was disabled after the data translation file was exported.	Enable data translation, and then import the file again.
Duplicate key: <key> exists in import file, re-export.	The specified key appears in your imported file more than one time. Each translated item must have its own unique key,	Export your file again, and make sure that each key is unique. Then import the file again.

Error Message	What It Means	Troubleshooting Steps
	and each key can only appear in the file one time.	
File contains translation keys that don't match the translation type specified in the file header. Create separate import files for metadata and data translation.	The file contains at least one key with a translation type that doesn't match the file type in the header.	Create separate import files for metadata and data translation.
Invalid Key	During translation, Salesforce generates unique keys, or identifiers, for each object, picklist value, or page element that you're translating. If these names or keys are changed after you export your file, Salesforce can't match the correct key with the correct name.	Export your file again, and make sure that the keys in it match the keys in the file that you're trying to import.
Invalid key: <key>. Data translation isn't enabled for the <field name> field on the <object name> object.	This error message only applies to data translation files. Data translation was disabled on the field after the translation file was exported.	Enable data translation on the field through the Data Translation Settings Setup page, or delete references to the key from the translation file.
Invalid key: <key>. The <field name> field on the <object name> object doesn't support data translation.	This error message only applies to data translation files. The key includes a field that doesn't support data translation. Either an exported key was changed after the export or the key was manually added.	Export your file again, and compare it with the identified key. If the identified key was modified, use the newly exported key. If it doesn't exist in the exported translation file, delete the identified key from the translation file.
Invalid key: <key>. The key's translation type must match the file's translation type.	Imported translation files must contain either metadata keys or data translation keys. This key doesn't match the file's translation type.	Update the file's translation type in the file header. Create separate import files for metadata and data translation as needed.
Invalid key: <key>. The <object name> object doesn't support data translation.	This error message only applies to data translation files. The key includes an object that doesn't support data translation. Either	Export your file again, and compare it with the identified key. If the identified key was modified, use the newly

Error Message	What It Means	Troubleshooting Steps
	an exported key was changed after the export or the key was manually added.	exported key. If it doesn't exist in the exported translation file, delete the identified key from the translation file.
Invalid key: <key>. You can't delete the data translation value for a record's name and update the value for another of that record's fields at the same time.	This error message only applies to data translation files. The file includes a key to delete the data translation for this record's Name. The file also includes a data translation value for another of the record's fields. Because deleting the translated value of a record's Name deletes the translated values for all fields related to that record, the translation keys conflict.	Edit your file. To delete all data translations associated with this record, remove all keys related to this record except the key deleting the translated value for the record's Name. To update other fields, remove the key deleting the translated value for the record's Name. See Considerations for Working with Translation Files for more information.
Key: <key> couldn't be uniquely resolved. Caused by a change to our Custom Report Type Column key format. Re-export and use the new key format for those keys.	One of the keys in your Custom Report Type (CRT) column is in the wrong format.	Export your file again, and make sure that you're using the correct CRT key format.
Maximum character limit <x> for <field type> translation exceeded in line:	Each type of field, such as a picklist value, can only have a certain number of characters. Your translated labels for the type of field at the line specified in the error message are too long.	Edit your translated labels so that they're within the character limit listed for the field type, and then import your file again.
Missing translation for: <key>	The missing translation for error simply means there is no target node in the segment of the source file.	Remove the segment from the source file or add target node to the segment.
No data to import	The file that you're trying to import is empty or doesn't contain any translation changes.	Make sure that you're importing the correct file and that it contains translated data.
No language code specified in file header	The file that you're trying to import doesn't have a valid	Make sure that your language

Error Message	What It Means	Troubleshooting Steps
	language code, or the language code is in the wrong place.	code is valid and isn't missing or commented out. Note: Make sure that the headers are in the organization's default language. To confirm, export a translation file and copy the headers.
No translated or untranslated section header found in the bilingual file	The file that you're trying to import is missing section headers.	Make sure that your file has section headers, and import it again.
No valid file type specified in file header	The file that you're trying to import doesn't have a valid import/export type (Source, Outdated and untranslated, or Bilingual) specified in the file header. The file type attribute must be in the default language for your org.	Make sure that your file's header contains a valid import/export type and isn't translated. Note: Make sure that the headers are in the organization's default language. To confirm, export a translation file and copy the headers.
No valid translation type specified in file header	<p>The file that you're trying to import doesn't have a valid translation type specified in the file header. The translation type is only required if data translation is enabled.</p> <p>The attributes for metadata translation are:</p> <ul style="list-style-type: none"> • <code>Translation Type: Metadata</code> for .stf files and • <code>translation-type="metadata"</code> for .xlf files <p>The attributes for data translation are:</p> <ul style="list-style-type: none"> • <code>Translation Type: Data</code> 	Make sure that your file has a valid translation type in the file header and that the header isn't translated. Note: Make sure that the headers are in the organization's default language. To confirm, export a translation file and copy the headers.

Error Message	What It Means	Troubleshooting Steps
	<p>for .stf files and</p> <ul style="list-style-type: none"> • <code>translation-type="data"</code> for .xlf files <p>The translation type attribute must be in the default language for your organization.</p>	
Not a valid file to import. Select a .stf, .xlf, or a .zip file for import.	You can import files in .stf or .xlf formats, or .zip files that contain .stf or .xlf files.	Make sure that your file is a .stf, .xlf, or .zip file, and try importing it again.
Salesforce cannot make this key for your Custom Report Type column unique: <Key>. Please delete this key from your file and use the Translation Workbench to translate this item. After you have deleted the key, import your file again.	The key appears more than once in the import file.	Delete the key from your file. Use the Translation Workbench to translate the item, then import your file.
Some keys are appended with their sort order for uniqueness. Re-export your file and ensure that the keys in both files match.	The order of the picklist values in your source file doesn't match your setup.	Export your source file, match the order of the picklist values to your import file, and then import again.
There were issues with import of file: importFileName.xlf [Your import request failed. Retry or contact support.]	The file that you're importing has invalid or missing HTML tags. HTML tags are used in translations for rich text area fields.	<p>Export your file again, and identify the fields with HTML tags in the exported source text.</p> <p>Edit your translation values for these rich text area fields with the correct HTML tags. For more information on translating rich text fields, see Considerations for Working with Translation Files.</p>
Wrong number of columns in line: <line number>. Check that you have escaped tabs (\t), new lines (\n), and carriage returns (\r) in your files.	The file that you're importing has extra tabs, new lines, or carriage returns in the line specified in the error message.	To remove or escape any extra tabs, newlines, or carriage returns be sure to edit your data. Make sure that the translated file has the same number of columns as the file you exported.

Error Message	What It Means	Troubleshooting Steps
Your export request failed. Retry or contact support.	Salesforce had an unexpected problem while exporting your file.	Contact Salesforce Customer Support.
Your import request failed. Retry or contact support.	Salesforce had an unexpected problem while importing your file.	Contact Salesforce Customer Support.
Your organization doesn't have language permissions for <language>.	The file that you're trying to import is in a language that you haven't yet added to Translation Workbench.	Add the language that you want to use to Translation Workbench through the Translation Language Settings Setup page. Then import your file again.

See Also

- [Export Metadata Translation Files](#)
- [Export Data Translation Files](#)
- [Import Translated Files](#)
- [Considerations for Working with Translation Files](#)

Translation Considerations

Review considerations for managing your translations and translating flows.

REQUIRED EDITIONS

Metadata translation available in: Salesforce Classic and Lightning Experience

Data translation available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Data translation applies to: B2B Commerce

[Considerations for Managing Translations](#)

Keep these tips in mind when creating custom labels, working with translators, and maintaining translations.

[Considerations for Translating Flows](#)

When you use Translation Workbench to translate flows, note these considerations.

Considerations for Managing Translations

Keep these tips in mind when creating custom labels, working with translators, and maintaining translations.

REQUIRED EDITIONS

Metadata translation available in: Salesforce Classic and Lightning Experience

Data translation available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Data translation applies to: B2B Commerce

- Salesforce assumes that all customizations are entered in the Salesforce org's default language. We recommend that global administrators work together in the org's default language.
- Salesforce recommends the XML Localization Interchange File Format (.xlf) for translation files.
- When creating a custom report type for translation into multiple languages via Translation Workbench, set your personal language to match your org's default language. Setting a language ensures that translated words display in the correct language for translators.
- Advise users customizing reports or list views to use filter criteria values in their personal language. However, if they use the `starts with` or `contains` operators, advise them to choose the language of the filter criteria values they entered.
- If you installed a managed package that includes translations, those translated values appear to users regardless of whether the language is active on the Translation Language Settings Setup page. To override metadata translations delivered by a managed package for custom objects, see Override Translations in Second-Generation Managed Packages and Unlocked Packages.
- Let translators know which languages they're responsible for translating.
- Notify all translators when you add new translated components to your org. For best results, have your translators check their translations frequently, and be sure to notify them when changes occur.
- Periodically review outdated translations by exporting your translations. To generate a list of all the translatable customizations and the associated Out of Date states, use the Outdated and Untranslated export type or Bilingual export type.
- If you have more than 25 navigation menu items open, use the scroll bar to find the navigation tools at the bottom of the window to navigate between pages.
- You can create up to 1,000 custom objects that include the picklist field. If you receive an error that your collection exceeded this limit, reduce the number of custom objects with a picklist field to 999 or fewer or export the translation files.
- You can view up to 10,000 records in Translation Workbench. To view more than 10,000 records, use translation import and export.

See Also

[Work with Translation Files](#)

[Override Translations in Second-Generation Managed Packages and Unlocked Packages](#)

Considerations for Translating Flows

When you use Translation Workbench to translate flows, note these considerations.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Translating Flows

- To check which flow types are translatable, see Flow Types in Salesforce Help.
- A translation can only reference a merge field (like {!myVar}) if the field is referenced in the first 1,000 characters of the primary label.
- Translations for flow definition name and version name each have a maximum limit of 255 characters. Other translations for flow labels have a maximum of 1,000 characters.
- Each use of a merge field in a label reduces the 1,000-character limit by 66 characters. For example, even though `{ !Account.Name }` is only 15 characters long, it counts as 66 characters toward the limit.
- Text templates and merge field values aren't supported for translation.
- Right-to-left languages aren't supported.
- When a flow label isn't translated for a language, Salesforce uses the translation for the appropriate fallback language. If the fallback language has no translated label, the primary label is used.

Updating Flow Translations

When you change a flow that you translated, Salesforce copies as much information as it can when you create a version or save changes to the translated version.

For example, version 1 of the Survey Customers flow has a translation for the WelcomeMessage field. When you save another version of the flow, Salesforce copies all the latest translations from version 1 to version 2. The same happens if you save it as a new flow.

When you remove a label from a flow, translations aren't copied. Salesforce uses the label's Unique Name to copy translations to another version. When you change the label's unique name, Salesforce treats it as a new label.

When you delete a flow, its translations are also deleted.

When you translate a flow from a managed package, the flow's Master Definition Name doesn't appear on the Translate page or the Override page. To update the translation for the Master Definition Name, edit the flow label and then update the translation from the Translate page.

Exporting and Importing Flow Translations

- You can export and import translation files to send to translators or to help you translate lengthy text.
- When you export translations, the primary labels are truncated after 1,000 characters.

See Also

[Salesforce Help: Translate Flow Screen Components](#)

Set Up Your Data Your Way

Optimize your Salesforce data to fit the unique needs of your users. You can create your own objects with data that fits together in the ways that make the most sense for you.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

The available customization options vary according to which Salesforce Edition you have.

USER PERMISSIONS NEEDED

To view setup options: [View Setup and Configuration](#)

To customize your org: [Customize Application](#)

[Store Information That's Unique to Your Organization](#)

Create custom objects to store information that's unique to your organization. Choose whether your custom objects are searchable, support sharing, or include access to the Bulk API and Streaming API.

[Store Customers' Data Privacy Preferences](#)

Store certain data privacy preferences for your customers.

[Classify Sensitive Data to Support Data Management Policies](#)

Record data sensitivity and compliance categorization at the field level. Data classification can be used to guide decisions around access, reporting, and data compliance.

[Design Your Own Data Model With Schema Builder](#)

Schema Builder provides a dynamic environment for viewing and modifying all the objects and relationships in your app. This greatly simplifies the task of designing, implementing, and modifying your data model, or schema. Schema Builder is enabled by default.

[Create Objects with Schema Builder](#)

Custom objects are objects that you create to store information that's specific to your company or industry. You can create them via Setup or in the Schema Builder.

[Create Fields with Schema Builder](#)

Add new custom fields to an object right inside Schema Builder.

[Delete Custom Objects with Schema Builder](#)

You can delete the custom objects that you no longer need by using Schema Builder.

[Delete Custom Fields with Schema Builder](#)

Avoid custom field clutter by using Schema Builder to delete custom fields that you no longer need.

[Schema Builder Custom Object Definition](#)

These fields are key to defining your custom object.

Schema Builder Considerations

Keep these items in mind when working with Schema Builder.

Create Custom Settings

Use custom settings to create custom sets of data, or to create and associate custom data for an org, profile, or user.

Customize Fields

Customize standard and custom fields to tailor your org to your own unique requirements.

Calculate Field Values with Formulas

A formula is an algorithm that derives its value from other fields, expressions, or values. Formulas can help you automatically calculate the value of a field based on other fields.

Generate Emails From Records

A merge field is a field you can put in an email template, mail merge template, custom link, or formula to incorporate values from a record. For example, you can place a merge field in an email template so that the greeting includes the recipient's name rather than a generic "Hello!".

Store Information That's Unique to Your Organization

Create custom objects to store information that's unique to your organization. Choose whether your custom objects are searchable, support sharing, or include access to the Bulk API and Streaming API.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Every custom object is classified as either an *Enterprise Application object* or a *Light Application object*. The difference between these two categories is that Light Application objects don't support sharing, access to the Bulk API, or access to the Streaming API.

By default, all custom objects are Enterprise Application objects. To make your custom object a Light Application object, disable **Allow Sharing, Allow Bulk API Access, and Allow Streaming API Access** on the object's detail page.

Manage Custom Objects

Create, customize, edit, delete, or truncate custom objects to extend the functionality that standard objects, like accounts and contacts, provide.

Manage Big Objects

A big object stores and manages massive amounts of data on the Salesforce platform. You can archive data from other objects or bring datasets from outside systems into a big object to get a full view of your customers. From Setup, you can create a custom big object and define its fields and index.

Object Relationships Overview

Create relationships to link objects with each other, so that when your users view records, they can

also see related data. For example, link a custom object called Bugs to cases to track product defects that are associated with customer cases.

Custom Object Security

Learn how security settings work together so you can control access to your custom objects with great flexibility.

Notes on Enabling Activities for Custom Objects

Learn about things to consider when enabling activities for custom objects.

Manage Custom Objects

Create, customize, edit, delete, or truncate custom objects to extend the functionality that standard objects, like accounts and contacts, provide.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com editions**

Managed Packages aren't available in **Database.com**.

USER PERMISSIONS NEEDED

To create and edit custom objects: Customize Application

 **Note** Your administrator may have created a tab without any help. If you need help to understand how a tab for a custom object works, contact your administrator.

Your object management settings list the custom objects that are defined for your organization. From this list, you can:

- Define a custom object.
- Display detailed information about a custom object.
Optional features you can customize include enabling search and reports, tracking activities, tracking field history, and making the object available for the Salesforce Customer Portal.
- To update the custom object definition, click **Edit** and update the desired fields.

 **Note** The Allow Reports, Allow Activities, and Allow Search fields aren't locked in Managed - Released and can be changed by the developer in future releases of a managed package.

- To delete a custom object, click **Del**.
- To truncate a custom object, click **Truncate**.
- To view deleted custom objects, click the **Deleted Objects** link. The total number of deleted custom objects for your organization is listed in parentheses.

The detail page of the custom object provides information about various characteristics of the object, including standard fields, custom fields, field history tracking, relationships, custom links, search layouts, page layouts, and object limits. You can:

- Click individual items to display additional detail.
- To delete a custom field, click **Del** next to its name in the Custom Fields & Relationships section.
- Click **More** at the bottom of the page or **View More** below a related list to display more items.
- Click **New** to directly add new items.

 **Note** The object limit percentages are truncated, not rounded. For example, if your org uses 95.55% of the limit for a particular customization, the object limit displays 95%.

Deployment Status for Custom Objects and External Objects

Use the Deployment Status setting in the object definition to control when users can see and use the object and its associated custom tab, related lists, and reports.

Create a Custom Object

Track and store data that's unique to your organization. Follow different steps, depending on which Salesforce experience you're using.

Modify Custom Objects

Customize the user interface for your custom objects.

Custom Object Standard Fields

When you create a custom object, these default fields are automatically assigned to the object.

Delete Custom Objects

When you delete a custom object, Salesforce doesn't add it to the Recycle Bin. Instead, deleted objects appear in the Deleted Objects list for 15 days. During this time, the object and its data are soft deleted, meaning you can restore or permanently erase (hard delete) the object and its data. After 15 days, the object and its data are automatically hard deleted.

Manage Deleted Custom Objects

Deleted custom objects appear in the Deleted Objects list for 15 days. During this time, you can choose to permanently delete the object and its data, or you can undelete them. If you undelete a custom object, some manual cleanup can be required to restore list views and other customizations that use the object.

Considerations for Truncating Custom Objects

It's important to understand what truncating an object does before you use it to remove records.

Truncate Custom Objects

Truncating custom objects allows you to delete all of the object's records permanently, but preserve the empty object and its metadata.

See Also

[Make Search Faster](#)

[Store Information That's Unique to Your Organization](#)

[Find Object Management Settings](#)

Deployment Status for Custom Objects and External Objects

Use the Deployment Status setting in the object definition to control when users can see and use the object and its associated custom tab, related lists, and reports.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Salesforce Connect external objects are available in Developer Edition and, for an extra cost, in Enterprise, Performance, and Unlimited Editions.

USER PERMISSIONS NEEDED

To deploy custom objects and external objects: Customize Application

While developing a custom object or external object, you might not want users to see and interact with it. Because users can get frustrated with changes in layout or lose data when you delete custom fields, control visibility of the new object until you're finished.

- Set the deployment status to In Development when first creating your custom object or external object. Doing so hides it from users while you're designing and testing it. Only users with the Customize Application permission can see the object tab, search results, related lists, and report data types.
- Change the deployment status to Deployed when you want to allow all users to use the object and the associated custom tab, related lists, and reports.
- If you make more enhancements after deploying a custom object or external object, you can change the deployment status back to In Development.

 **Note** A custom report type's deployment status changes from Deployed to In Development if its primary object is a custom or external object whose deployment status similarly changes.

When you create a big object, the status is set to In Development. You can't deploy a big object until it includes an index that contains at least one custom field. Only required custom fields are allowed in an index. After you create an index, you see a second status of Deployed. When you're ready to grant users access, change the status to Deployed.

See Also

[Make Search Faster](#)

Create a Custom Object

Track and store data that's unique to your organization. Follow different steps, depending on which Salesforce experience you're using.

[Create a Custom Object in Lightning Experience](#)

Track and store data that's unique to your organization.

[Create a Custom Object from a Spreadsheet in Lightning Experience](#)

Use custom objects to track and store data that's unique to your organization. If you prefer not to create a custom object and its fields manually, you can use a spreadsheet to add the object and its fields and populate all its record data.

[Create a Custom Object in Salesforce Classic](#)

Track and store data that's unique to your org.

[Fields Required for Creating Custom Objects](#)

When you create a custom object, several fields are required to define how you can access the object.

[Considerations for Creating Custom Objects](#)

Before you create a custom object, make sure that you review these considerations.

See Also

[Make Search Faster](#)

[Object Relationships Overview](#)

[Define Object-Level Help in Salesforce Classic](#)

Create a Custom Object in Lightning Experience

Track and store data that's unique to your organization.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create and edit custom objects: [Customize Application](#)

1. From the Object Manager tab in Setup, click **Create | Custom Object**.
2. Complete the fields for your custom object and configure its features.
3. If you want to create a custom tab for the object immediately after you save it, select **Launch New Custom Tab Wizard after saving this custom object**.
To create the custom object tab later, from Setup in the Quick Find box, enter *Tabs*, and then click **Tabs**.
4. Save the new object.

5. In the Object Manager, click **Fields & Relationships**, and create the custom fields that your object needs.

 **Tip** If you don't want your users to see the new custom object while you design and test it, to hide it, set the deployment status to In Development.

See Also

- [Create a Custom Object from a Spreadsheet in Lightning Experience](#)
- [Fields Required for Creating Custom Objects](#)
- [Considerations for Creating Custom Objects](#)
- [Create a Custom Object](#)

Create a Custom Object from a Spreadsheet in Lightning Experience

Use custom objects to track and store data that's unique to your organization. If you prefer not to create a custom object and its fields manually, you can use a spreadsheet to add the object and its fields and populate all its record data.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create and edit custom objects: Customize Application

 **Note** Although it's available in Professional Edition, this feature uses API. To use this feature, Professional Edition orgs must have the API add on enabled. To enable the API add on, contact your Account Executive.

You can create a custom object from a spreadsheet in two places: in Setup, or from the Navigation Items tab inside a Lightning app's settings.

1. In Setup, go to Object Manager, then click **Create | Custom Object from Spreadsheet**.
 2. Alternatively, navigate to the Navigation Items tab inside a Lightning app.
 - a. In Setup, enter *App* in the Quick Find box, then select **App Manager**.
 - b. Next to the app you want to add the custom object to, click  and select **Edit**.
 - c. Click **Navigation Items**.
 - d. From the Available Items list, click **Create | Custom Object from Spreadsheet**.
 3. Follow the wizard steps to import your custom object data from an .xlsx file, a .csv file, or a Google sheet.
- Salesforce detects object field labels from the spreadsheet row that you specify. All fields must be mapped to create the custom object. Skipping the import step creates an empty custom object that

uses the fields in the spreadsheet as a template. When you finish creating the object, a custom tab is created for it. The object appears in the list of available items for your app. If you imported field data, your object is ready to go with fully populated records.

4. If you don't want your users to see the new custom object right away, in the Object Manager in Setup, set its deployment status to In Development. This setting hides the object from users while you're designing and testing it.
5. When you're ready for your users to see the new custom object, add it to your app's Selected Items list.

See Also

[Fields Required for Creating Custom Objects](#)
[Create a Custom Object in Lightning Experience](#)
[Considerations for Creating Custom Objects](#)
[Create a Custom Object](#)

Create a Custom Object in Salesforce Classic

Track and store data that's unique to your org.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create and edit custom objects:	Customize Application
------------------------------------	-----------------------

1. From Setup, enter *Objects* in the Quick Find box, then select **Objects**.
2. Click **New Custom Object**.
3. Follow the wizard to complete the fields for your custom object.
4. Save the new object.

See Also

[Fields Required for Creating Custom Objects](#)
[Considerations for Creating Custom Objects](#)
[Create a Custom Object](#)

Fields Required for Creating Custom Objects

When you create a custom object, several fields are required to define how you can access the object.

REQUIRED EDITIONS

! **Important** Where possible, we changed noninclusive terms to align with our company value of

Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: Lightning Experience and Salesforce Classic

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

-  **Note** If an administrator created a tab without including help, contact your administrator if you need help with how a custom object works.

Field	Description
Label	This name is used to refer to the object in a user interface page.
Plural Label	The plural name of the object. If you create a tab for this object, this name is used for the tab.
Gender	If it's appropriate for your org's default language, specify the gender of the label. This field appears if the org-wide default language expects gender. Your personal language preference setting doesn't affect whether the field appears. For example, if the org's default language is English and your personal language is French, you aren't prompted for gender when creating a custom object.
Starts with a vowel sound	If it's appropriate for your org's default language, indicate whether "an" or "a" precedes the label.
Object Name	A unique name used to refer to the object when using the API. In managed packages, this name prevents naming conflicts with package installations. Use only alphanumeric characters and underscores. The name must begin with a letter and have no spaces. It can't end with an underscore nor have two consecutive underscores.
Description	An optional description of the object. A meaningful description helps you remember the differences between objects when you're viewing them in a list.
Context-Sensitive Help Setting	Defines the URL that displays when a user clicks Help for this Page from the object record's home (overview), edit, and detail pages, list views, and

Field	Description
	<p>related lists. This setting doesn't affect the Help link at the top of a page. That link always opens the Help window.</p> <ul style="list-style-type: none"> • To display the standard Salesforce Help available for any custom object record, select Open the standard Salesforce Help & Training window. • To display custom object-level help for your custom object, select Open a window using a Visualforce page and then select the Visualforce page to use as the target of the context-sensitive help link from that custom object's pages.
Record Name	<p>The name used in page layouts, list views, related lists, and search results.</p> <p>If you select the Auto Number data type, there could be issues when inserting a high volume of records, for example, via the API. If you anticipate a high volume of record inserts, use the Text data type.</p>
Data Type	<p>The type of field (text or auto-number) for the record name. Records that have unique IDs instead of names are auto-numbered and are always a read-only field.</p>
Display Format	<p>For an auto-numbered record name, enter the display format. You can have up to two sets of curly braces.</p>
Starting Number	<p>For an auto-numbered record name, enter the number to use when creating your first record for this custom object.</p>
Allow Reports	<p>Makes the data in the custom object records available for reporting purposes.</p> <p>To create reports on custom objects, choose the Other Reports report type category, unless the</p>

Field	Description
	<p>custom object has a relationship with a standard object. When the custom object has a master-detail relationship with a standard object or is a lookup object on a standard object, select the standard object for the report type category instead.</p> <p>You can still create and run reports without selecting Allow Reports; however, the custom report type isn't visible.</p>
Allow Activities	<p>Allows users to associate tasks and scheduled calendar events related to the custom object records.</p>
Allow in Chatter Groups	<p>Allows users to add records of this custom object type to Chatter groups.</p> <p>When <code>true</code>, users with permissions can create records of this object type using the group publisher. The created record is associated with the group and appears in the group record list.</p> <p>When <code>false</code>, users with permissions can use the group publisher to create records of this object type, but the record isn't associated with the group.</p>
Enable Divisions	<p>If your org has divisions enabled, select this option to enable the custom object for divisions. A division groups records for simplified search results, list views, reports, and other areas within Salesforce. Salesforce adds a Division field to the custom object. If the custom object is the master in a master-detail relationship, custom objects on the detail side also get the Division field and inherit their division from the master record.</p>
Available for Customer Portal	<p>Makes the custom object available to all portal users.</p> <p>This option is available only if your org has a customer portal.</p>

Field	Description
	If you enable Digital Experiences in your org, this option no longer appears, and all custom objects are available in your Experience Cloud sites. If before enabling, you had a Customer Portal and custom objects without this option selected, those objects become available in your Customer Portal.
Track Field History	Enables your org to track changes to fields on the custom object records. For example, it tracks who changed the field value and when, what the value was before the edit, and what it was changed to. History data is available for reporting, so users can easily create audit trail reports when this feature is enabled.
Allow Sharing	<p>When this setting is enabled, the custom object is an Enterprise Application object. When this setting isn't enabled, the custom object is a Light Application object.</p> <p>When this setting is enabled, you must also enable Allow Bulk API Access and Allow Streaming API Access.</p>
Allow Bulk API Access	<p>When this setting is enabled, the custom object is an Enterprise Application object. When this setting isn't enabled, the custom object is a Light Application object.</p> <p>When this setting is enabled, you must also enable Allow Sharing and Allow Streaming API Access.</p>
Allow Streaming API Access	<p>When this setting is enabled, the custom object is an Enterprise Application object. When this setting isn't enabled, the custom object is a Light Application object.</p> <p>When this setting is enabled, you must also enable Allow Bulk API Access and Allow Sharing.</p>
Deployment Status	Indicates whether the custom object is visible to other users.

Field	Description
Allow Search	<p>To allow your users to find a custom object's records when they search, create a custom tab set to Default On or Default Off. Creating a custom tab enables the custom object's Allow Search setting.</p> <p>Searchability is disabled for custom objects that haven't been searched for more than 120 days. To enable object and field searchability, contact your admin.</p>
Add Notes & Attachments...	<p>Allows users to attach notes and attachments to custom object records. You can attach external documents to any object record in much the same way that you can add a PDF file or photo as an attachment to an email.</p> <p>This option is available only when you're creating an object.</p>
Launch the New Custom Tab Wizard	Starts the custom tab wizard after you save the custom object.

See Also

[Create a Custom Object](#)

Considerations for Creating Custom Objects

Before you create a custom object, make sure that you review these considerations.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Object Creation

- Establish object relationships first, before adding all custom fields, page layouts, and related lists.
- The standard Name field is required on custom object related lists and page layouts.
- Provide meaningful names for your custom objects. The plural label of the custom object is used as the label of the custom tab based on that object.

- When you create a custom object, you specify the data type of the Record Name field: Auto Number or Text. If you select the Auto Number data type, there could be issues when inserting a high volume of records, for example, via the API. If you anticipate a high volume of record inserts, use the Text data type.
- When creating an object from a spreadsheet:
 - Skipping the import step creates an empty custom object that uses the fields in the spreadsheet as a template.
 - You can click  to preview the object data. Only the first 24 rows of data are displayed in preview.
 - These field types aren't supported:
 - Auto Number
 - Formula
 - Roll-Up Summary
 - Lookup Relationship
 - Master-Detail Relationship
 - External Lookup Relationship
 - Text Area (Rich)
 - Text (Encrypted)
 - Time

Object Permissions

In Enterprise, Unlimited, Performance, Professional, and Developer editions, when you create a custom object, the Read, Create, Edit, and Delete permissions for that object are disabled for profiles that have View All Data or Modify All Data disabled. Enable access to custom objects in permission sets or custom profiles, and assign them to the users who need access.

In Contact Manager and Group editions, when you create a custom object, the Read, Create, Edit, and Delete permissions for that object are enabled for all profiles.

Sharing Model

An org-wide default setting controls the data sharing model for custom objects. For more information, see [Custom Object Security](#).

Delegating Custom Object Administration

After you create a custom object, you can delegate its administration to non-admin users.

Queues

After you create a custom object, you can define queues to distribute ownership of custom object records to your users.

See Also

[Create a Custom Object](#)
[Salesforce Features and Edition Allocations](#)

Modify Custom Objects

Customize the user interface for your custom objects.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To customize custom objects:	Customize Application
------------------------------	-----------------------

- [Create a custom tab](#).
- [Create custom fields and relationships](#).
- Add [customized buttons and links](#) to perform actions or link to other pages or websites.
- Define which [fields](#) display for users on record detail and edit pages.
- Specify which [fields display for users](#) in search results, lookup dialogs, and in the key lists on custom object tabs.
- [Create record types](#) to display different picklist values and page layouts to different users based on their profiles.

Custom Object Standard Fields

When you create a custom object, these default fields are automatically assigned to the object.

REQUIRED EDITIONS

! **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Divisions aren't available in **Database.com**.

USER PERMISSIONS NEEDED

To view and edit standard fields:	Customize Application
-----------------------------------	-----------------------

USER PERMISSIONS NEEDED

To create custom fields: **Customize Application**

Custom object fields store the data for your custom object records.

Custom Fields for Custom Objects

You can create custom fields to store information unique to your org. You can also create custom relationship fields to associate your custom object with another object in Salesforce.

Standard Fields for Custom Objects

Custom objects automatically include these standard fields. Click **Edit** to modify any of the editable fields.

Field Name	Description
CreatedById	ID of the user who created the record.
Currency	Currency of the record if multicurrency is enabled.
Division	Division to which the custom object record belongs. Custom objects that are “detail” objects in a master-detail relationship inherit their division from the master object. Custom objects that aren’t related to other records are automatically in the global division. Available only in orgs that use divisions to segment their data.
LastModifiedById	ID of the user who most recently changed the record.
Name	Identifier for the custom object record. This name appears in page layouts, related lists, lookup dialogs, search results, and key lists on tab home pages. By default, this field is added to the custom object page layout as a required field.
OwnerId	ID of the assigned owner of the custom object record. If the custom object becomes the detail side of a master-detail relationship, this field is removed, as ownership of the data is controlled by the master object, or by the primary master object for a custom object with two master-detail relationships.

Field Name	Description
	Custom objects on the detail side of a master-detail relationship can't have sharing rules, manual sharing, or queues, as these require the Owner field.

Delete Custom Objects

When you delete a custom object, Salesforce doesn't add it to the Recycle Bin. Instead, deleted objects appear in the Deleted Objects list for 15 days. During this time, the object and its data are soft deleted, meaning you can restore or permanently erase (hard delete) the object and its data. After 15 days, the object and its data are automatically hard deleted.

REQUIRED EDITIONS

! **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: Salesforce Classic

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

USER PERMISSIONS NEEDED

To delete custom objects:

Customize Application

AND

View All Data

Soft-deleted custom objects and their data count against your org's limits but hard-deleted items don't.

To delete a custom object:

- From the object management settings for custom objects, click **Del** next to the object that you want to delete.
- When prompted, select **Yes, I want to delete the custom object** to confirm, and click **Delete**.

You can't delete a custom object if it:

- Is on the master side of a master-detail relationship
- Contains custom fields that are used in a roll-up summary field on another object
- Is referenced in Apex, a Visualforce page, or a reporting snapshot
- Is referenced by a duplicate rule or a matching rule

- Contains more than 100,000 records. If the object that you want to delete has more than 100,000 records, delete an appropriate number of records first, and then delete the object.

When you delete a custom object, Salesforce:

- Displays an Insufficient Privileges message if someone clicks a bookmark to a deleted custom object record's URL
- Removes the object from AppExchange packages
- Changes the master-detail relationship to a lookup relationship if the deleted object is on the detail side of a master-detail relationship
- Removes or erases:
 - The object's custom tab
 - List views and workflow rules for the object
 - Mobile configuration settings including datasets, mobile views, and excluded fields
 - Standard report types associated with the object, and reports based on standard report types if the deleted object is on the detail side of a master-detail relationship
- Hides, inactivates, or disables:
 - The custom object definition and related definitions
 - The object's records and related records, including any records in the Recycle Bin
 - Custom report types where the deleted object is the main object
 - Custom reports where the deleted object is the main object
 - Custom formula fields on the object
 - Custom validation rules and approval processes on the object

To restore removed, hidden, inactive, or disabled items, you can undelete the custom object. See [Manage Deleted Custom Objects](#) for information about restoring deleted custom objects.

See Also

- [Manage Deleted Custom Objects](#)
- [Find Object Management Settings](#)
- [Delete a Big Object](#)

Manage Deleted Custom Objects

Deleted custom objects appear in the Deleted Objects list for 15 days. During this time, you can choose to permanently delete the object and its data, or you can undelete them. If you undelete a custom object, some manual cleanup can be required to restore list views and other customizations that use the object.

 **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

View Deleted Custom Objects

To view a list of deleted custom objects, go to object management settings for custom objects. The

Deleted Objects link appears only when you have at least one deleted custom object in your org. The number in parentheses indicates the total number of deleted custom objects. In the Deleted Objects list, you can:

- Click the object's label to view details about it.
- Click **Erase** to permanently remove the object and its data.
- Click **Undelete** to restore the object and its data.

What Happens When You Hard Delete a Custom Object

A custom object can be hard deleted either manually or automatically after 15 days.

- The custom object's definition and data are permanently deleted and can't be restored.
- The custom object and its data no longer count against your org's limits.
- If the deleted object is on the detail side of a master-detail relationship, master records currently in the Recycle Bin aren't restorable if one or more detail records were automatically deleted as a result of the master record being deleted. Attempting to undelete the master record results in an error.

 **Note** This scenario only happens when the deleted detail records have their custom object definition hard deleted while the master record is in the Recycle Bin.

Limitations for Restoring Truncated Custom Objects

Copies of truncated custom objects also appear in the list of deleted objects. Truncated custom objects can't be restored to their original state. Undeleted copies of truncated objects have a new name and new URL, and some fields and data can't be manually restored.

Restoring a Custom Object to Its Predeleted State

When you restore a deleted custom object, its records are also undeleted, including any that were in the Recycle Bin. It can take several hours before you can search an undeleted object's records.

To ensure that you return the object to its fully functional, predeleted state, check all affected conditions and customizations, and manually fix them if necessary.

AppExchange packages

Add the custom object to any appropriate AppExchange packages.

Custom tabs

Re-create a custom tab for the object and add it to any custom apps that use it.

List views, reports, and workflow rules

Re-create them.

Validation rules and approval processes

Reactivate them.

Formula fields

Open and save any custom formula fields on the object to re-enable them.

Page layouts

Page layouts are restored automatically on the undeleted object. Page layouts are also restored automatically on other objects that use the page layout in a related list—as long as the page layout wasn't edited while the object was deleted. Otherwise, you must add the related list back to the other object.

Custom report types

For custom report types where the deleted object wasn't the main object, add the reference back to the restored object. Reports based on the custom report type are automatically restored if they weren't edited while the object was deleted. Re-create any reports that were edited.

Relationships

If the deleted custom object was on the detail side of a master-detail relationship, Salesforce converted it to a lookup relationship. Change the relationship back to master-detail.

Developer name

The developer name for the object was changed to *objectname_del*. Change it back to the original name, *objectname_c*, so that customizations using the name work properly.

Deployment status

When the custom object was deleted, its Deployment Status field was set to In Development. After you restore all affected customizations to the undeleted object, change its status back to Deployed.

See Also

[Delete Custom Objects](#)

[Considerations for Truncating Custom Objects](#)

[Find Object Management Settings](#)

Considerations for Truncating Custom Objects

It's important to understand what truncating an object does before you use it to remove records.

REQUIRED EDITIONS

 **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Truncating a custom object lets you remove all of the object's records, while keeping the object and its metadata. Truncating custom objects is similar to the mass delete option available for standard objects.

Truncating a custom object permanently removes all of its records. You can't recover the records from the Recycle Bin. A copy of the truncated object appears in the Deleted Objects list for 15 days—during this period the object and its records continue to count toward your org's limits—and then the copied object and its records are permanently deleted.

In contrast, if you delete a custom object, the object moves to the Deleted Objects list for 15 days. After that the object and its records are permanently deleted.

 **Important** Truncated custom objects can't be restored to their original state.

You can't truncate standard objects or custom objects that:

- Are referenced by another object through a lookup field or that are on the master side of a master-detail relationship
- Are referenced in a reporting snapshot
- Have a custom index or an external ID
- Have activated skinny tables

In addition, you can't truncate custom objects when your org has reached its limit on allowed custom objects.

Truncating a custom object erases:

- All records currently sitting in the custom object's Recycle Bin
- The custom object's history
- Related events, tasks, notes, and attachments for each deleted record

Truncating a custom object breaks:

- Bookmarks to the truncated object and its records. If someone clicks a bookmark to the truncated custom object or to a deleted record's URL, Salesforce displays an `Insufficient Privileges` message.
- Apex scripts and Visualforce pages with references to a truncated object or record.

After truncating a custom object, you can continue to use the custom object and add new records. Salesforce preserves:

- The custom object definition and all related definitions
- Workflow rules, actions, and triggers
- Sharing rules associated with the custom object

- Validation rules and approval processes
- Master-detail relationships and formula fields
- Translations
- Mobile configuration settings

When working with truncated objects, keep in mind:

- The truncated object tab has a new URL, so new bookmarks need to be created.
- List views and reports must be refreshed after truncation.
- Roll-up summary fields must be recalculated after truncation.
- There's no support for truncation in the API.
- To truncate objects that contain master-detail relationships, first truncate the detail (child) objects and then the (master) parent objects, working your way up the relationship tree.

See Also

[Truncate Custom Objects](#)

[Manage Deleted Custom Objects](#)

Truncate Custom Objects

Truncating custom objects allows you to delete all of the object's records permanently, but preserve the empty object and its metadata.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

USER PERMISSIONS NEEDED

To truncate custom objects: Customize Application

! **Important** Truncating custom objects causes some irreversible changes to the truncated object and its records. Before truncating, see [Truncating Custom Objects](#). Then, enable it for your organization by entering *User Interface* in the Quick Find box, selecting **User Interface**, and then selecting the permission.

Truncating custom objects is a way to permanently remove all of the records from a custom object, while keeping the object and its metadata intact for future use. Truncating is useful, for example, if you've created a custom object and filled it with test records. When you're done with the test data, you can truncate the object to purge the test records, but keep the object and put it into production. This is much faster than batch-deleting records and possibly recreating the object.

1. Go to the object management settings for custom objects.

2. Click an object name to go to the object's detail page, and then click **Truncate**.
3. In the Confirm Custom Object Truncate window, review the warning, then enter the name of the object to truncate in the empty field.
4. Click **Truncate**.

See Also

[Manage Deleted Custom Objects](#)

[Find Object Management Settings](#)

Manage Big Objects

A big object stores and manages massive amounts of data on the Salesforce platform. You can archive data from other objects or bring datasets from outside systems into a big object to get a full view of your customers. From Setup, you can create a custom big object and define its fields and index.

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions for up to 1 million records

Additional record capacity and Async SOQL query available as an add-on license.

From Setup, enter *Big Objects* in the Quick Find box, then select **Big Objects**. From this page you can:

- Create a big object.
- View and update details about a big object.
- Delete a big object or view any big objects that were deleted in the last 15 days.

[Create or Edit a Big Object](#)

Track and store big data that's unique to your org. For existing big objects, a details page shows information about the big object's fields and index.

See Also

[Salesforce Developers: Big Objects Implementation Guide](#)

Create or Edit a Big Object

Track and store big data that's unique to your org. For existing big objects, a details page shows information about the big object's fields and index.

Before you create a big object, make sure that you understand the restrictions on updating and deleting fields for big objects described in [Changing Big Object Fields](#).

1. From Setup, in the Quick Find box, enter *Big Objects*, and then select **Big Objects**.
2. Click **New** or click an existing big object.
3. Add or edit details about the big object.
4. Add or edit custom fields. Custom fields store the data for your big object records.

5. Add an index. The index defines the composite primary key for a big object and is used to query and filter the big object data.
6. Save the big object.

After you create an index and deploy a big object, you can't edit or delete the index. To change the index, you must start over with a new big object.

Big Object Definition Details

When you create a custom big object, several required fields define how you can access the big object.

Custom Big Object Fields

Custom fields store the data for your custom big object records.

Considerations When Creating an Index

Plan carefully when creating an index for your custom big object. The index is used for querying and filtering the big object data and must be designed properly.

Add an Index to a Big Object

The index defines the composite primary key for a custom big object and is used for querying and filtering the big object data. Each big object requires an index.

Changing Big Object Fields

Learn how big object fields differ from standard object fields when you update or delete them. Make sure that your big object is correct before you save it.

Delete a Big Object

When you delete a custom big object, Salesforce doesn't add it to the Recycle Bin. Instead, deleted big objects appear in the Deleted Objects list for 15 days. During this time, the big object and its data are soft deleted, meaning you can restore or permanently erase (flag for hard delete) the big object and its data. After 15 days, the big object and its data are automatically flagged for hard delete.

Big Object Definition Details

When you create a custom big object, several required fields define how you can access the big object.

The big object definition details page displays information about the big object's fields and index.

Field Name	Description
Label	This name is used to refer to the object in a user interface page.
Plural Label	The plural name of the object.
Starts with a vowel sound	If it's appropriate for your organization's default language, indicate whether "an" or "a" precedes the label.
Object Name	A unique name used to refer to the object when using the API. In managed packages, this name prevents naming conflicts with package installations. Use only alphanumeric characters and underscores. The

Field Name	Description
	<p>name must begin with a letter and have no spaces. It cannot end with an underscore nor have two consecutive underscores.</p> <p>In the API, the names of custom big objects have a suffix of two underscores immediately followed by a lowercase “b” (_b). For example, a big object named “HistoricalInventoryLevels” is seen as HistoricalInventoryLevels__b in that organization’s WSDL.</p>
Description	<p>An optional description of the object. A meaningful description helps you remember the differences between objects when you are viewing them in a list.</p>
Context-Sensitive Help Setting	<p>Defines the URL that displays when a user clicks Help for this Page from the object record’s home (overview), edit, and detail pages, list views, and related lists. This setting doesn’t affect the Help link at the top of a page. That link always opens the Help window.</p> <ul style="list-style-type: none"> • To display the standard Salesforce Help available for any custom object record, select Open the standard Salesforce Help & Training window. • To display custom object-level help for your custom object, select Open a window using a Visualforce page and then select the Visualforce page to use as the target of the context-sensitive help link from that custom object’s pages.
Deployment Status	<p>When you create a big object, the status is set to In Development. You can’t deploy a big object until it includes an index that contains at least one custom field. Only required custom fields are allowed in an index. After you create an index, you see a second status of Deployed. Once you’re ready to grant users access, change the status to Deployed.</p>

Custom Big Object Fields

Custom fields store the data for your custom big object records.

Custom Fields for Big Objects

Create custom fields to store information unique to your org. You can also create custom relationship fields to associate your big object with another object in Salesforce. Big objects support these field types:

- Lookup Relationship
- Date/Time
- Email

- Number
- Phone
- Text
- Text Area (Long)
- URL

To create an index for your big object, at least one custom field must be marked as required.

Standard Fields for Big Objects

Currently, big objects don't include standard fields.

See Also

[Create a Custom Field](#)

Considerations When Creating an Index

Plan carefully when creating an index for your custom big object. The index is used for querying and filtering the big object data and must be designed properly.

Keep these considerations in mind when creating the index.

- An index must include at least one custom field and can have up to five custom fields total.
 - All custom fields that are part of the index must be marked as required.
 - You can't include Long Text Area and URL fields in the index.
 - The total number of characters across all text fields in an index can't exceed 100.
-  **Note** Email fields are 80 characters. Phone fields are 40 characters. Keep these lengths in mind when designing your index because they count toward the 100 character limit.
- After you've created the index, you can't edit or delete it. To change the index, you must start over with a new big object.
 - Design your index so that you assign the most frequently used field in a query filter to Index Position 1. The order in which you define the fields determines the order that they're listed in the index.

Add an Index to a Big Object

The index defines the composite primary key for a custom big object and is used for querying and filtering the big object data. Each big object requires an index.

To create an index, you must have at least one required custom field defined on the big object. See [Considerations When Creating an Index](#) for more details.

1. From the Index section of a big object detail page, click **New**. This button displays only if the big object has at least one required custom field.
2. Add a Label and Name for the index.

 **Warning** When querying a big object record via SOQL and passing the results as arguments to the

delete API, if any index field name has a leading or trailing white space, you can't delete the big object record.

3. For each custom field listed, set the Index Position and Index Direction. The order in which you define the fields determines the order that they're listed in the index. Set the Index Position to **1** for the most frequently used filter parameter.



4. Save the index.

Changing Big Object Fields

Learn how big object fields differ from standard object fields when you update or delete them. Make sure that your big object is correct before you save it.

Updating Fields for Big Objects

If a field has an existing value, you can update it with a different value. You can't update it with a NULL value. You can delete a record and recreate it if you want to change the value to NULL.

Most properties cause data mutation. You can't change big object field properties that cause data mutation and you can't edit the properties after you create an object. Properties that don't cause data mutation are related to information about the data, such as labels or help text.

Deleting Fields for Big Objects

You can't delete an individual field after the object is created. So if you want to delete a field from a Big Object, delete the object and then recreate it.

Delete a Big Object

When you delete a custom big object, Salesforce doesn't add it to the Recycle Bin. Instead, deleted big objects appear in the Deleted Objects list for 15 days. During this time, the big object and its data are soft deleted, meaning you can restore or permanently erase (flag for hard delete) the big object and its data. After 15 days, the big object and its data are automatically flagged for hard delete.

Salesforce runs a background job to completely remove big objects that have been flagged for hard-delete from your org. The job doesn't run immediately, but you can't access these large objects because they no longer appear in the Deleted Objects list.

Soft-deleted big objects and their data always count against your org's limits. Big objects that are flagged for hard delete, but the background job hasn't yet removed, also count. After the background job removes these big objects, they no longer count against your org's limit.

1. From Setup, enter *Big Objects* in the Quick Find box, then select **Big Objects**.
2. Click **Del** next to the object that you want to delete.

3. When prompted, select the **Yes, I want to delete the Big Object** checkbox to confirm and click **Delete**.
The big object is soft deleted and remains in the Deleted Objects list for 15 days.
4. To permanently erase (flag for hard-delete) the big object, click **Deleted Big Object**.
5. Click **Erase** next to the big object you want to permanently erase.
6. When prompted, select the **Yes, I want to permanently delete the Big Object** checkbox to confirm and click **Delete**.
The big object is flagged for hard-delete and can no longer be restored.

Object Relationships Overview

Create relationships to link objects with each other, so that when your users view records, they can also see related data. For example, link a custom object called Bugs to cases to track product defects that are associated with customer cases.

REQUIRED EDITIONS

 **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

You can define different types of relationships by creating custom relationship fields on an object. Before you begin creating relationships, determine the type of relationship that suits your needs.

Different types of relationships between objects in Salesforce determine how they handle data deletion, sharing, and required fields in page layouts. Let's review the types of relationships.

Master-detail

Closely links objects together such that the master record controls certain behaviors of the detail and subdetail record. For example, you can define a two-object master-detail relationship, such as Account–Expense Report that extends the relationship to subdetail records, such as Account–Expense Report–Expense Line Item. You can then perform operations across the master–detail–subdetail relationship.

 **Tip** Create a master-detail relationship before a custom object contains data.

Behaviors of master-detail relationships:

- Deleting a detail record moves it to the Recycle Bin and leaves the master record intact; deleting a master record also deletes related detail and subdetail records. Undeleting a detail record restores it,

and undeleting a master record also undeletes related detail and subdetail records. However, if you delete a detail record and later separately delete its master record, you can't undelete the detail record, as it no longer has a master record to relate to.

- By default, records can't be reparented in master-detail relationships. Administrators can, however, allow child records in master-detail relationships on custom objects to be reparented to different parent records by selecting the **Allow reparenting** option in the master-detail relationship definition.
- The Owner field on the detail and subdetail records isn't available and is automatically set to the owner of the master record. Custom objects on the detail side of a master-detail relationship can't have sharing rules, manual sharing, or queues, as these require the Owner field.
- Detail and subdetail records inherit security settings and permissions from the master record. You can't set permissions on the detail record independently.
- The master-detail relationship field (which is the field linking the objects) is required on the page layout of the detail and subdetail records.
- The master object can be a standard object, such as Account or Opportunity, or a custom object.
- As a best practice, don't exceed 10,000 child records for a master-detail relationship.
- Each custom object can have up to two master-detail relationships and up to 40 total relationships.
- The Related To entry can't be changed after you save the relationship.
- A profile or a permission set can have an object, such as Account, with a master-detail relationship. A broken permission dependency exists if the child object has permissions that the parent must have. For updates made in Setup, Salesforce updates the parent object for a broken permission dependency on the first save action for the profile or permission set. For updates made using the API, you must manually fix broken permission dependencies.

If the child object has these permissions	These permissions are enabled on the parent object
Modify All Records OR View All Records	View All Records
View All Records OR Read	Read

- When you create a draft Knowledge Article version from a published version, the Roll Up Summary field on the draft article carries forward the Roll Up Summary field values of the published article. As per design, when you edit an article, a new draft version is created and custom field values from the published version are carried over to the new draft version. However, custom object records associated with a KnowledgeArticleVersion (published article) are not carried over or attached to the new draft version.

Many-to-many

You can use master-detail relationships to model *many-to-many* relationships between any two objects. A many-to-many relationship allows each record of one object to be linked to multiple records from another object and vice versa. For example, you create a custom object called Bug that relates to the standard case object such that a bug could be related to multiple cases and a case could also be related to multiple bugs.

Lookup

Links two objects together. Lookup relationships are similar to master-detail relationships, except they don't support sharing or roll-up summary fields. With a lookup relationship, you can:

- Link two different objects.
- Link an object with itself (except for the user object; see the Hierarchical section in this topic). For example, link a custom object called Bug with itself to show how two different bugs are related to the same problem.

 **Note** Lookup relationships from objects related to the campaign member object aren't supported; however, you can create lookup relationships from the campaign member object related to other objects.

When you create a lookup relationship, you can set these options:

- Make the lookup field required for saving a record, requiring it on the corresponding page layout as well.
- If the lookup field is optional, you can specify one of three behaviors to occur if the lookup record is deleted:
 - **Clear the value of this field** This is the default. Clearing the field is a good choice when the field doesn't have to contain a value from the associated lookup record.
 - **Don't allow deletion of the lookup record that's part of a lookup relationship** If you have dependencies built on the lookup relationship, such as a workflow rule, this option doesn't allow the lookup record to be deleted.

 **Note** Deleting a record that has child records isn't allowed, except when the child records are soft-deleted (sent to the Recycle Bin). If all the child records of a parent record are soft-deleted, then the parent record is deleted. Furthermore, any soft-deleted children are then removed from the recycle bin and permanently deleted.

- **Delete this record also** Available only if a custom object contains the lookup relationship, not if it's contained by a standard object. However, the lookup object can be either standard or custom. Choose when the lookup field and its associated record are tightly coupled and you want to completely delete related data. For example, say that you have an expense report record with a lookup relationship to individual expense records. When you delete the report, you probably want to delete all the expense records, too.

 **Warning** Choosing **Delete this record also** can result in a cascade-delete. A cascade-delete bypasses security and sharing settings, which means users can delete records when the target lookup record is deleted even if they don't have access to the records. To prevent records from being accidentally deleted, cascade-delete is disabled by default. Contact Salesforce to get the cascade-delete option enabled for your org. Cascade-delete and its related options aren't available for lookup relationships to business hours, network, lead, price book, product, or user objects.

When you define a lookup relationship, you can include a lookup field on the page layouts for that object

and create a related list on the associated object's page layouts. For example, if you have a custom object called PTO Requests and you want your users to link a PTO request with the employee submitting the request, create a lookup relationship from the PTO Request custom object with the user object.

If the parent record in a lookup relationship is deleted, the field history tracking for the child record doesn't record the deletion. For example, if a parent account is deleted, the Account History related list for the child account doesn't show the deletion.

You can't delete an object or record in a lookup relationship if the combined number of records between the two linked objects is more than 100,000. To delete an object or record in a lookup relationship, first delete an appropriate number of its child records.

When you delete an object used by a lookup field, delete the field, too. To delete both the object and the field, use the Metadata API with a delete manifest that uses *purgeOnDelete*. Or, use Setup in the UI to delete the field first. Otherwise, the object can't be deleted.

External lookup

An external lookup relationship links a child standard, custom, or external object to a parent external object. When you create an external lookup relationship field, the standard External ID field on the parent external object is matched against the values of the child's external lookup relationship field. External object field values come from an external data source.

Indirect lookup

An indirect lookup relationship links a child external object to a parent standard or custom object. When you create an indirect lookup relationship field on an external object, you specify the parent object field and the child object field to match and associate records in the relationship. Specifically, you select a custom unique, external ID field on the parent object to match against the child's indirect lookup relationship field, whose values come from an external data source.

Hierarchical

A special lookup relationship available for only the user object. It lets users use a lookup field to associate one user with another that doesn't directly or indirectly refer to itself. For example, you can create a custom hierarchical relationship field to store each user's direct manager.

 **Tip** When creating a hierarchical field in Personal, Contact Manager, Group, and Professional Editions, you can select the Restricted Field checkbox so that only users with the Manage Internal Users permission can edit it. In Professional, Enterprise, Unlimited, Performance, and Developer Edition, use field-level security instead.

Create a Many-to-Many Object Relationship

You can use master-detail relationships to model *many-to-many* relationships between any two objects. A many-to-many relationship allows each record of one object to be linked to multiple records

from another object and vice versa. For example, you create a custom object called Bug that relates to the standard case object such that a bug could be related to multiple cases and a case could also be related to multiple bugs. When modeling a many-to-many relationship, you use a *junction object* to connect the two objects you want to relate to each other.

Considerations for Object Relationships

Review these considerations before creating relationships between objects.

See Also

- [Considerations for Object Relationships](#)
- [External Object Relationships](#)
- [Create a Many-to-Many Object Relationship](#)
- [Create a Custom Object](#)

Create a Many-to-Many Object Relationship

You can use master-detail relationships to model *many-to-many* relationships between any two objects. A many-to-many relationship allows each record of one object to be linked to multiple records from another object and vice versa. For example, you create a custom object called Bug that relates to the standard case object such that a bug could be related to multiple cases and a case could also be related to multiple bugs. When modeling a many-to-many relationship, you use a *junction object* to connect the two objects you want to relate to each other.

REQUIRED EDITIONS

 **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Reports aren't available in **Database.com**.

USER PERMISSIONS NEEDED

To create a many-to-many relationship: Customize Application

Junction Object

A custom object with two master-detail relationships. Using a custom junction object, you can model a “many-to-many” relationship between two objects.

For example, you create a custom object called “Bug” that relates to the standard case object such that a bug could be related to multiple cases and a case could also be related to multiple bugs.

How to create many-to-many relationships:

1. [Create the junction object.](#)
2. [Create the two master-detail relationships.](#)
3. [Customize the related lists](#) on the page layouts of the two master objects.
4. [Customize reports](#) to maximize the effectiveness of the many-to-many relationship.

Create the Junction Object

1. [Create a custom object to be your junction object.](#)
2. In the custom object wizard, consider these tips specifically for junction objects:
 - Name the object with a label that indicates its purpose, such as *BugCaseAssociation*.
 - For the Record Name field, we recommend that you use the auto-number data type.
 - Don't launch the custom tab wizard before clicking **Save**. Junction objects don't need a tab.

Create the Two Master-Detail Relationships

To create the two master-detail relationships:

1. Verify that the two objects you want to relate to each other exist. For example, you want to relate the standard case object to a custom bug object.
2. On the junction object, create the first master-detail relationship field. In the custom field wizard:
 - a. Choose **Master-Detail Relationship** as the field type.
 - b. Select one of the objects to relate to your junction object. For example, select **Case**.

The first master-detail relationship you create on your junction object becomes the *primary* relationship. This relationship affects the following for the junction object records.

- Look and feel: The junction object's detail and edit pages use the color and any associated icon of the primary master object.
 - Record ownership: The junction object records inherit the value of the Owner field from their associated primary master record. Because objects on the detail side of a relationship don't have a visible Owner field, this inherited value is only relevant if you later delete both master-detail relationships on your junction object.
 - Division: If your org uses divisions to segment data, the junction object records inherit their division from their associated primary master record. Similar to the record ownership, this inherited division is only relevant if you later delete both master-detail relationships.
- c. Select a Sharing Setting option. For master-detail relationship fields, the Sharing Setting attribute determines the sharing access that users must have to a master record to create, edit, or delete its associated detail records.
 - d. For the Related List Label that's displayed on the page layout of the master object, don't accept the default value. Change the value to use the name of the other master object in your many-to-many relationship. For example, change the value to *Bugs* so users see a Bugs related list on the case detail page.
3. On the junction object, create the second master-detail relationship. In the custom field wizard:
 - a. Choose **Master-Detail Relationship** as the field type.
 - b. Select the other desired master object to relate to your junction object. For example, select **Bug**.

The second master-detail relationship you create on your junction object becomes the *secondary* relationship. If you delete the primary master-detail relationship or convert it to a lookup relationship, the secondary master object becomes primary.

- c. Select a Sharing Setting option. For master-detail relationship fields, the Sharing Setting attribute determines the sharing access that users must have to a master record to create, edit, or delete its associated detail records.
- d. For the Related List Label that displays on the page layout of the master object, don't accept the default value. Change this value to use the name of the other master object in your many-to-many relationship. For example, change this value to *Cases* so users see a Cases related list on the bug detail page.

Customize Many-to-Many Relationship Related Lists

For a many-to-many relationship in Salesforce, each master object record displays a related list of the associated junction object records. To create a seamless user experience, you can change the name of the junction object related list on each of the master object page layouts to have the name of the other master object. For example, you can change the BugCaseAssociations related list to *Cases* on the bugs page layout and to *Bugs* on the cases page layout. You can further customize these related lists to display fields from the other master object.

To customize the fields that display in the junction object related list on each master object page layout:

1. Edit the page layout of each master object that is related to the junction object. For example, modify the BugCaseAssociations related list for case records by editing the page layout for cases.
2. Edit the properties of the related list you want to modify. For example, on cases the BugCaseAssociations related list was renamed to Bugs, so select the Bugs related list.
3. Add the fields to display in the related list. You can add fields from the junction object itself, but more importantly, you can add fields from the other master object.

Each field is prefixed with its object name in the window. In the related list itself, only fields from the junction object are prefixed with the object name; fields from the other master object aren't.

 **Note** The junction object related list doesn't include an icon on the master record's detail pages because the junction object doesn't have a custom tab. If you make a tab for the junction object, the icon is included.

Customize Reports for Many-to-Many Relationships

Many-to-many relationships provide two standard report types that join the master objects and the junction object. The report types are:

- “Primary master with junction object and secondary master” in the primary master object's report category.
- “Secondary master with junction object and primary master” in the secondary master object's report category.

The order of the master objects in the report type is important. The master object listed first determines the scope of records that can be displayed in the report.

You can create custom reports based on these standard report types. In addition, you can create custom report types to customize which related objects are joined in the report.

See Also

- [Find Object Management Settings](#)
- [Object Relationships Overview](#)
- [Considerations for Object Relationships](#)
- [Create a Custom Object](#)

Considerations for Object Relationships

Review these considerations before creating relationships between objects.

REQUIRED EDITIONS

 **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Salesforce Connect external objects are available in **Developer** Edition and for an extra cost in **Enterprise, Performance, and Unlimited** Editions

Relationship Limits

Each custom object can have up to 2 master-detail relationships and many lookup relationships. Each relationship is included in the maximum number of custom fields allowed.

Converting Relationships

You can convert a master-detail relationship to a lookup relationship as long as no roll-up summary fields exist on the master object.

Converting a master-detail relationship to a lookup for a custom object on the “detail” side, changes the organization-wide default for the object to public read/write.

You can convert a lookup relationship to a master-detail relationship if the lookup field in all the records contains a value.

A lookup relationship can't be changed to a master-detail relationship if the organization-wide default of the child object access level in the relationship is Controlled by Parent.

Converting a lookup to a master-detail-relationship changes the organization-wide default to Controlled by Parent and the sharing model is updated to public read/write.

Self-Relationships

You can create a relationship from an object to itself, but it must be a lookup relationship, and a single record can't be linked to itself. However, a record can indirectly relate to itself. For example, the Holiday Promotion campaign can select the Direct Mail campaign in the lookup relationship, and the Direct Mail campaign can select the Holiday Promotion campaign in the lookup relationship.

You can't create a many-to-many self-relationship, that is, the two master-detail relationships on the junction object can't have the same master object.

Icons for Custom Related Lists

The icon you select for the associated custom tab also displays in any custom-related list you create based on a relationship.

Custom related lists don't include an icon if they're based on a relationship with a custom object that doesn't have a custom tab.

Master-Detail Relationships

To create multilevel master-detail relationships, you need the Customize Application user permission.

When you define a master-detail relationship, the custom object on which you're working is the detail side. Its data appears as a custom related list on page layouts for the other object.

By default, records can't be reparented in master-detail relationships. Administrators can, however, allow child records in master-detail relationships on custom objects to be reparented to different parent records by selecting the **Allow reparenting** option in the master-detail relationship definition.

You can have up to 3 custom detail levels.

Standard objects can't be on the detail side of a custom object in a master-detail relationship.

An object can appear one time in multilevel master-detail relationships. For example, a subdetail object in one multilevel master-detail relationship can't also be the owner of the master object in another multilevel master-detail relationship. A subdetail object can't also be the master object of the subdetail object's detail object.

Multilevel master-detail relationships don't support division transfers.

You can't create a master-detail relationship if the custom object already contains data. You can,

however, create the relationship as a lookup and then convert it to master-detail if the lookup field in all records contains a value.

Converting relationships from lookup to master-detail, or from master-detail to lookup, behaves the same as for two-object master-detail relationships. That is, the two linked objects in the detail-subdetail1 or subdetail1-subdetail2 relationship have the same conversion limits as the master-detail relationship.

Roll-up summary fields work as in two-object master-detail relationships. A master can roll up fields on detail records; however, it can't directly roll up fields on subdetail records. The detail record must have a roll-up summary field for the field on the subdetail record, allowing the master to roll up from the detail's roll-up summary field.

You can use multilevel master-detail relationships in custom report types. The Allow Reports checkbox must be selected when you create the custom object. Custom report types created for multilevel master-detail relationships count toward the organization's custom report type limit, and no reports generate if this limit is exceeded.

Custom junction objects can't have detail objects. That is, a custom junction object can't become the master object in a multilevel master-detail relationship.

You can't delete a custom object if it is on the master side of a master-detail relationship. If you delete a custom object that is on the detail side of a master-detail relationship, the relationship is converted to a lookup relationship.

Deleting a detail record moves it to the Recycle Bin and leaves the master record intact; deleting a master record also deletes related detail and subdetail records. Undeleting a detail record restores it, and undeleting a master record also undeletes related detail and subdetail records. However, if you delete a detail record and later separately delete its master record, you can't undelete the detail record, as it no longer has a master record to relate to.

A Metadata API deployment that includes Master-Detail relationships deletes all detail records in the Recycle Bin in these cases.

- For a deployment with a new Master-Detail field, soft delete (send to the Recycle Bin) all detail records before proceeding to deploy the Master-Detail field, or the deployment fails. During the deployment, detail records are permanently deleted from the Recycle Bin and can't be recovered.
- For a deployment that converts a Lookup field relationship to a Master-Detail relationship, detail records must reference a master record or be soft-deleted (sent to the Recycle Bin) for the deployment to succeed. However, a successful deployment permanently deletes any detail records in the Recycle Bin.

As a best practice, don't exceed 10,000 child records for a master-detail relationship.

A profile or a permission set can have an entity, such as Account, with a master-detail

relationship. A broken permission dependency exists if the child entity has permissions that the parent should have. Salesforce updates the parent entity for a broken permission dependency on the first save action for the profile or permission set.

If the child entity has these permissions	These permissions are enabled on the parent entity
Modify All Records OR View All Records	View All Records
View All Records OR Read	Read

Many-to-Many Relationships

Junction object records are deleted when either associated master record is deleted and placed in the Recycle Bin. If both associated master records are deleted, the junction object record is deleted permanently and can't be restored.

Sharing access to a junction object record is determined by a user's sharing access to both associated master records and the Sharing Setting option on the relationship field. See [Custom Object Security](#). For example, if the sharing setting on both parents is Read/Write, then the user must have Read/Write access to *both* parents in order to have Read/Write access to the junction object. If the sharing setting on both masters is Read-Only, a user with Read-Only rights on the master records would have Read access to the junction object.

In a many-to-many relationship, a user can't delete a parent record if there are more than 200 junction object records associated with it *and* if the junction object has a roll-up summary field that rolls up to the other parent. To delete this object, manually delete junction object records until the count is fewer than 200.

The first master-detail relationship that you create on your junction object becomes the *primary* relationship. This relationship affects junction object records in these ways.

- Look and feel: The junction object's detail and edit pages use the color and any associated icon of the primary master object.
- Record ownership: The junction object records inherit the value of the Owner field from their associated primary master record. Because objects on the detail side of a relationship don't have a visible Owner field, this is only relevant if you later delete both master-detail relationships on your junction object.
- Division: If your organization uses divisions to segment data, the junction object records inherit their division from their associated primary master record. Similar to the record ownership, this is only relevant if you later delete both master-detail relationships.

The second master-detail relationship you create on your junction object becomes the *secondary* relationship. If you delete the primary master-detail relationship or convert it to a lookup relationship, the secondary master object becomes primary.

Roll-up summary fields that summarize data from the junction object can be created on both master objects.

Formula fields and validation rules on the junction object can reference fields on both master objects.

You can define Apex triggers on both master objects and the junction object.

A junction object can't be on the master side of another master-detail relationship.

You can't create a many-to-many self-relationship, that is, the two master-detail relationships on the junction object can't have the same master object.

Lookup Relationships

If the lookup field is optional, you can specify one of three behaviors to occur if the lookup record is deleted:

- Clear the value of this field—This is the default. Clearing the field is a good choice when the field doesn't have to contain a value from the associated lookup record.
- Don't allow deletion of the lookup record that's part of a lookup relationship—if you have dependencies built on the lookup relationship, such as a workflow rule, this option doesn't allow the lookup record to be deleted.

 **Note** Deleting a record that has child records isn't allowed, except when the child records are soft-deleted (sent to the Recycle Bin). If all the child records of a parent record are soft-deleted, then the parent record is deleted. Furthermore, any soft-deleted children are then removed from the recycle bin and permanently deleted.

- Delete this record also—Available only for a custom lookup field on a custom object. This option isn't available for a custom lookup field that refers to a standard object. Choose when the lookup field and its associated record are tightly coupled and you want to completely delete related data.

 **Warning** Choosing **Delete this record also** can result in a *cascade-delete*. A cascade-delete bypasses security and sharing settings, which means users can delete records when the target lookup record is deleted *even if they don't have access to the records*. To prevent records from being accidentally deleted, cascade-delete is disabled by default. Contact Salesforce to get the cascade-delete option enabled for your organization. Cascade-delete and its related options aren't available for lookup relationships to standard objects.

In a chain of lookup relationships, these behaviors work independently on each target field at each level. Say, for example, field A is the target lookup of field B, which in turn is the target lookup of field C. You can have a delete restriction on A and none on B, which means that A can't be deleted but B can. After B is deleted, the relationship between A and B no longer exists and C holds an empty value for the lookup.

In a multilevel lookup relationship, these options can conflict. For example, if field A is the target

lookup of field B, which in turn is the target lookup of field C, you can specify that A deletes B, but B can't be deleted because it's in a relationship with C. If you try to delete A, you get an error that B can't be deleted because it's linked to C.

If the parent record in a lookup relationship is deleted, the field history tracking for the child record doesn't record the deletion. For example, if a parent account is deleted, the Account History related list for the child account doesn't show the deletion.

You can't select indirect lookup fields in the parent field when you add the Related List - Single component to a Lightning Page. Instead, select the related list that's associated with the indirect lookup field. It doesn't show data in the related list, but shows the lookup field with no issue.

Relationships on External Objects

Lookup, external lookup, and indirect lookup relationships have some special behaviors and limitations.

- Only lookup, external lookup, and indirect lookup relationships are available for external objects. No other relationship types are supported.
- Depending on the availability of the external system, related lists of child external objects can load slowly when users view the parent record detail pages.
- Relationships that involve external objects allow users to create child records from the record detail pages of parent records. However, the relationship field on each new child record isn't automatically populated to identify the parent record.
- Syncing doesn't create relationship fields on the external objects in your Salesforce org. However, you can change the field type of a sync-created custom field to Lookup Relationship, External Lookup Relationship, or Indirect Lookup Relationship. Changing the field type of an existing custom field is simpler and more efficient than manually creating a relationship field on the external object.

For example, suppose that the external system has a foreign key relationship. Syncing the related tables creates a text field in your org for the external column that identifies the foreign keys. To reflect the foreign key relationship within your org, change the field type of that text field to External Lookup Relationship.

- A relationship field is a type of custom field. Therefore, like all custom fields on an external object, relationship fields can be overwritten when you sync the external object. See the sync considerations for each Salesforce Connect adapter that you use.
- Cascade-delete isn't available for external object relationships.
- In Salesforce Classic, indirect lookup relationship fields don't display the expected names of parent records. Instead, each indirect lookup relationship field displays the value of the target field on the parent object. To find related records, target field values are matched against the values of the indirect lookup relationship field on the child object. The target field, which has the `External ID` and `Unique` attributes, is selected when an indirect lookup relationship field is created.
- In Salesforce Classic, external lookup relationship fields don't always display the expected names of parent records.
 - In a list view, an external lookup relationship field displays the parent object ID or the value of the parent object's External ID standard field. The latter appears by default, but if a custom field

on the parent object has the `Is Name Field` attribute, the parent object ID is displayed.

- In a record detail page, an external lookup relationship field displays the name as expected if the org has previously retrieved the parent record. If you see an ID in an external lookup relationship field, reload the page to replace the ID with the name.
- Lookup search isn't available for external lookup relationship fields. To edit an external lookup relationship field, manually enter the value of the External ID standard field for the parent record. This limitation doesn't apply when the parent external object is associated with the cross-org adapter for Salesforce Connect.
- Lookup search isn't available for indirect lookup relationship fields. To edit an indirect lookup relationship field, manually enter the value of the target field of the parent record. The target field is the custom field with `External ID` and `Unique` attributes that was selected when the indirect lookup relationship was created. To determine related records, Salesforce matches target field values against the values of the indirect lookup relationship field on the child object.
- With external lookup and indirect lookup relationships, the parent record appears as a clickable link in the relationship field on the child record. If the child record is viewed by a user who doesn't have access to the parent record, the parent record appears in the relationship field as plain text instead of a link.
- Lookup filters aren't available for external lookup relationship fields.
- Indirect lookup relationship fields can be created on external objects only.
- Only objects that have a custom field with the `External ID` and `Unique` attributes are available as parent objects in indirect lookup relationships. If you don't see the desired object when you create an indirect lookup relationship field, add a custom unique, external ID field to that object.
- If the external system uses case-sensitive values in the specified External Column Name, make sure that the parent object field is also case-sensitive. When you define the parent object's custom field, select **External ID, Unique, and Treat "ABC" and "abc" as different values (case sensitive)**.

Impact of Relationships on Reports

The type of relationship you create affects which standard report types are available and how they're categorized. These report types determine which related objects can be included in the report.

- Lookup relationships allow data from the two related objects to be joined in one report.
- Master-detail relationships allow data from three objects to be joined in one report: the master object, the detail object, plus one other lookup object. If the detail object has multiple lookup relationships, a separate report type is available based on each lookup.
- Many-to-many relationships provide two standard report types that join the master objects and the junction object. The order of the master objects in the report type is important. The master object listed first determines the scope of records that can be displayed in the report.
 - “Primary master with junction object and secondary master” in the primary master object’s report category.
 - “Secondary master with junction object and primary master” in the secondary master object’s report category.

The reporting impact of each relationship type is summarized in this table.

Relationship Type	Standard Report Types	Report Type Category
Lookup	Object by itself Object with first lookup Object with second lookup Object with third lookup	Based on the object
Master-Detail	Master object by itself Master object with detail object Master object with detail object and first lookup Master object with detail object and second lookup Master object with detail object and third lookup	Master object
Many-to-Many	Primary master object by itself Secondary master object by itself Primary master object with junction object and secondary master object Secondary master object with junction object and primary master object	Primary master object and Secondary master object

Custom report types give you more flexibility to join data from multiple objects, including lookups and master-detail relationships.

- ❗ **Important** Converting a relationship from lookup to master-detail or vice versa can cause existing custom reports to become unusable due to the different standard report types available for each type of relationship. We recommend that you test your custom reports

immediately after converting the relationship type. If you revert your relationship back to the original type, the reports are restored and become usable again.

See Also

- [Object Relationships Overview](#)
- [Create a Many-to-Many Object Relationship](#)
- [External Object Relationships](#)

Custom Object Security

Learn how security settings work together so you can control access to your custom objects with great flexibility.

REQUIRED EDITIONS

! **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Tabs aren't available in **Database.com**.

Set custom object security at the following levels

- Tab—display the custom tab for the appropriate users based on their user profiles.
- Object—set the access users have to create, read, edit, and delete records for each object.
- Records—set the default sharing model for all your users. This determines the access users have to custom object records that they don't own.
- Relationship—for objects on the detail side of a master-detail relationship, specify the sharing access that users must have to the master record in order to create, edit, or delete the associated detail records. This is specified in the Sharing Setting attribute of the master-detail relationship field on the detail object.
- Fields—set the level of access users have to fields on your custom object page layout.

These requirements apply to custom objects with no master-detail relationship.

Action	Required Privileges
Create a record	Create permission. The user must have the tab displayed to create a record from the Create New dropdown list in the sidebar.

Action	Required Privileges
View a record	Read permission and Public Read Only or Public Read/Write sharing model if not the record owner.
Edit a record	Edit permission and Public Read/Write sharing model if not the record owner.
Delete a record	Delete permission and must be the record owner or above the record owner in the role hierarchy.

These requirements apply to custom objects that have a master-detail relationship with a standard or custom object.

Action	Required Privileges
Create a record	Create permission and either read or read/write access to the related master record, depending on the value of the Sharing Setting attribute of the master-detail relationship field on the detail object.
View a record	Read permission and read access to the related master record. If the record has two master records in a many-to-many relationship, the user must have read access to both master records.
Edit a record	Edit permission and either read or read/write access to the related master record, depending on the value of the Sharing Setting attribute of the master-detail relationship field on the detail object.
Delete a record	Delete permission and either read or read/write access to the related master record, depending on the value of the Sharing Setting attribute of the master-detail relationship field on the detail object. When a user deletes a record that has related custom object records, all related custom object records are deleted regardless of whether the user has delete permission to the custom object.

Delegated administrators can manage nearly every aspect of specified custom objects, but they can't create or modify relationships on the object or set organization-wide sharing defaults.

Notes on Enabling Activities for Custom Objects

Learn about things to consider when enabling activities for custom objects.

REQUIRED EDITIONS

 **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

- If you enable activities when creating a custom object, the activity related lists are added to the default page layout automatically. If you enable activities later, after the custom object already exists, you must add the related lists to the page layout manually.
- Disabling activities for a custom object doesn't delete existing activity records. However, activity related lists are removed from custom object pages, and reports containing activities and the custom object are deleted.
- If a custom object has a master-detail relationship with accounts, the custom object's activities roll up to the account and cause the account's Last Activity date to be updated. For custom objects related to other types of records, the activities don't roll up.
- The ability to send emails or create mail merge documents is available for activities on custom objects. The email must be sent to a contact or lead.
- When you change the ownership of a custom object record, any open activities related to that custom object are also transferred to the new record owner.
- You can't disable activity tracking for a custom object if any workflow tasks are associated with that custom object.
- Custom object records can only be associated with a call log in Salesforce CRM Call Center if activities are enabled for the object.

See Also

[Create a Custom Object](#)

Something Went Wrong...There's No Help for This Custom Object

You landed on this page because you were looking for more information about a particular custom object, but your Salesforce admin didn't create a help page for that object. Review the suggested resources. If you need information on a specific custom object, contact your administrator about creating custom help for your custom objects.

See Also

[Work with Related Lists on Records in Salesforce Classic](#)

[Customize Your Data View](#)

[Define Object-Level Help in Salesforce Classic](#)

Something Went Wrong...There's No Help for This Custom Object

You landed on this page because you were looking for more information about a particular custom object, but your Salesforce admin didn't create a help page for that object. Review the suggested resources. If you need information on a specific custom object, contact your administrator about creating custom help for your custom objects.

See Also

- [Find and View Your Data](#)
- [Customize Your Data View](#)
- [Define Object-Level Help in Salesforce Classic](#)

Store Customers' Data Privacy Preferences

Store certain data privacy preferences for your customers.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions, including partner and customer community users.

Regardless of whether you're working toward complying with data protection and privacy regulations, data privacy records can help you track and store your customers' consent. We've listed some of the regulations that are important to many companies collecting and processing their customers' data.

- General Data Protection Regulation (GDPR), European Union
- Gramm-Leach-Bliley Act (GLB Act), United States
- Canada's Anti-Spam Law (CASL)

[Set Up Tracking and Storage of Certain Data Privacy Preferences](#)

Let your users store and track certain data privacy preferences for your customers.

[Consent Management Objects](#)

Get familiar with the objects that you can use for managing your customers' privacy and consent.

[Tracking Customer Requests for Data Privacy Updates](#)

If you store certain data privacy preferences in data privacy records based on the Individual object, track customers' requests so that you can honor their wishes.

[Best Practices for Tracking Data Privacy](#)

Keep these best practices in mind when storing certain data privacy preferences in data privacy records based on the Individual object.

[Track Certain Data Privacy Preferences for Leads and Contacts Already in Salesforce](#)

Create data privacy records based on the Individual object for leads and contacts already in Salesforce

using scripts.

Manage Duplicate Data Privacy Records

Keep your records clean and free of duplicates so that you can reach more customers and maintain better relationships with them.

Set Up Tracking and Storage of Certain Data Privacy Preferences

Let your users store and track certain data privacy preferences for your customers.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions, including partner and customer community users.

Data protection details are available in all your org's records by default. Enable the Consent Data model in the settings. You can easily verify that they're available or disable them from the Setup page.

1. From Setup, in the Quick Find box, enter *Data Protection and Privacy*, and then select **Data Protection and Privacy**.
2. Click **Edit**, select or unselect **Make data protection details available in records**, and then click **Save**.

You can make data protection information visible to users by adding the **Individual** field to Lead, Contact, and Person Accounts page layouts. Consider renaming this field to something meaningful to your users. (*Example:* Manage data privacy or Track customer consent)

 **Note** If you don't see tabs for the consent management objects, [set the objects' tab settings to Default On](#) for the profiles where you want to enable tabs.

Consider enabling or updating these settings.

- Create custom fields for data privacy records
- Create sharing rules for data privacy records
- Encrypt personal data in certain data privacy fields (Shield customers)
- Set the organization-wide sharing default
- Track field history for individuals

See Also

[Track Certain Data Privacy Preferences for Leads and Contacts Already in Salesforce](#)

[Best Practices for Tracking Data Privacy](#)

[Respect Consent Preferences in Marketing Cloud Engagement with the Consent Data Model](#)

Consent Management Objects

Get familiar with the objects that you can use for managing your customers' privacy and consent.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions. The Individual object is available for partner and customer community users.

After you learn about these objects and fields, you can track and store certain data privacy preferences. But keep in mind that you must decide how to honor your customers' privacy, and then implement a process to accomplish that.

For example, if you send communications to customers with Marketing Cloud Engagement, you can manage customers' preferences for how they want to be contacted using consent management objects. To learn more, read the [Respect Consent Preferences in Marketing Cloud Engagement with the Consent Data Model](#) solution kit.

 **Note** Consent management object records aren't counted toward your storage usage.

Consent Management Objects

Object	What It Tracks
Authorization Form	The version number and effective dates of the authorization form.
Authorization Form Consent	Information related to the customer's consent to the authorization form.
Authorization Form Data Use	The data use purpose associated with the authorization form.
Authorization Form Text	The text and language of the authorization form.
Communication Subscription	The customer's preferences for a communication subscription.
Communication Subscription Channel Type	The engagement channel through which to contact a customer for a communication subscription.
Communication Subscription Consent	Information related to the customer's consent to the communication subscription.
Communication Subscription Timing	The customer's timing preferences for receiving a communication subscription.
Contact Point Consent	Information related to the customer's consent to be contacted via a specific contact point.
Contact Point Email	The customer's preference for the time that they prefer to be contacted via email.

Object	What It Tracks
Contact Point Phone	The customer's preference for the time that they prefer to be contacted via phone.
Contact Point Type Consent	Customer allows contact via a specific contact point type like email, but not through phone calls and mail.
Data Use Purpose	The reason for collecting customer data.
Data Use Legal Basis	The legal basis for contacting an individual or party, such as legitimate interest.
Engagement Channel Type	The channel to use to reach a customer, such as email or web.
Individual	<p>The customer's preferences for:</p> <ul style="list-style-type: none"> • Collecting, storing, and sharing their personal data. • Packaging their personal data so they can take ownership of it. • Deleting records and personal data related to them. • Solicitation of products and services. • Tracking their geolocation and web activity.
Party Consent	Information related to an individual's consent.

Authorization Form Objects

Keep track of data related to authorization forms, such as terms of service, privacy policy, and other consent forms. Each authorization form object stores different data. You can use them together to create a full picture of your customer's consent to the authorization form.

Communication Subscription Objects

You can store and manage the data related to your customer's communication subscriptions, such as newsletters or appointment reminders. Different communication subscription objects store different data, such as how the customer consented to the communication subscription and preferred contact timing.

Fields in Data Privacy Records

Get the skinny on the fields that appear in data privacy records based on the Individual object. Data privacy records store certain privacy settings for your customers.

Fields in Contact Point Consent Records

Represents a customer's consent to be contacted via a specific contact point, such as an email address or phone number.

Fields in Contact Point Email Records

Fields in contact point email records store multiple email records related to an individual.

Fields in Contact Point Phone Records

Fields in contact point phone records store multiple phone records related to an individual.

Fields in Contact Point Type Consent Records

Contact point type consent records let you enter details about how a customer has agreed to be contacted by your company. For example, you can indicate that a customer consented to a specific contact point like email, but not phone calls and mail.

Fields in Data Use Purpose Records

Represents a category that defines the reason for contacting an individual. For example, billing, marketing, or surveys.

Fields in Data Use Legal Basis Records

Represents the legal basis for contacting an individual. For example, billing or contract.

Fields in Party Consent Records

Represents consent preferences for an individual.

Authorization Form Objects

Keep track of data related to authorization forms, such as terms of service, privacy policy, and other consent forms. Each authorization form object stores different data. You can use them together to create a full picture of your customer's consent to the authorization form.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

Authorization Form

Stores the name, version, and effective dates of the authorization form. You can use the revision number to determine if you're using the current version. You can also link to a default authorization form text record to use if text isn't available for a specific language.

Authorization Form Consent

Tracks information related to each customer's consent to the authorization form. Indicates the individual that consented and when, how, and to which version of the authorization form.

Authorization Form Data Use

Optionally, you can connect an authorization form to one or more [data use purpose](#) records. Indicate the data use purpose associated with the form, such as billing or marketing, and its corresponding legal basis.

Authorization Form Text

Manages the text associated with the authorization form. You can create multiple text versions for the same authorization form to support different languages, regions, and situations. You can also include a summary to describe the form's purpose and display to customers when asking for their consent.

Communication Subscription Objects

You can store and manage the data related to your customer's communication subscriptions, such as newsletters or appointment reminders. Different communication subscription objects store different data, such as how the customer consented to the communication subscription and preferred contact timing.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions.

Communication Subscription

Store a customer's subscription preferences for a specific communication.

Communication Subscription Channel Type

Specify the engagement channel that you can use to contact the customer for a communication subscription.

Communication Subscription Consent

Track information related to each customer's consent to the communication subscription. Indicate the individual or person account that consented, when consent was captured, and through what method. You can also specify if another person consented on behalf of the customer listed in the record. The user's actual consent isn't stored, but rather the action or communication types that the user chooses to subscribe to.

Communication Subscription Timing

Manage the customer's time preferences for receiving a communication subscription. For example, track whether to send a reminder to a patient 2 weeks before an appointment or a newsletter to a customer every Tuesday. You can also ensure that you reach customers when they prefer and in the correct time zone.

Engagement Channel Type

Store the channels by which you can communicate with a customer, such as email, phone, or web.

Fields in Data Privacy Records

Get the skinny on the fields that appear in data privacy records based on the Individual object. Data privacy records store certain privacy settings for your customers.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions, including partner and customer community users.

Field	Description
Birth Date	The customer's birth date.
Block Geolocation Tracking	Preference to not track geolocation on mobile devices.
Consumer Credit Score	The individual's credit score (for example, 740).
Consumer Credit Score Provider Name	The name of the company that provided the credit score.
Conviction Count	The number of convictions for the customer.
Death Date	The individual's death date.
Don't Market	Preference to not receive marketing materials.
Don't Process	Preference to not process personal data, which can include collecting, storing, and sharing personal data.
Don't Profile	Preference to not process data for predicting personal attributes, such as interests, behavior, and location.
Don't Solicit	Preference to not solicit products and services.
Don't Track	Preference to not track customer web activity and whether the customer opens email sent through Salesforce.
Export Individual's Data	Preference to export personal data for delivery to the individual.

Field	Description
Forget This Individual	Preference to delete records and personal data related to this individual.
Individual's Age	Indication for whether the individual is considered to be a minor.
Influencer Rating	A measure of the person's influence, irrespective of how we do business with them.
Is Homeowner	Indicates whether the customer owns a home.
Military Service	Indicates whether the customer has served in the military.
Number of Children	The number of children the customer has.
Occupation	The customer's occupation. Maximum length is 150 characters.
OK to Store PII Data Elsewhere	Indication that you can store personally identifiable information outside of their legislation area. For example, you could store an EU citizen's personal data in the US.

Fields in Contact Point Consent Records

Represents a customer's consent to be contacted via a specific contact point, such as an email address or phone number.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions.

Field	Description
Business Brand	Represents a unique brand for a business that belongs to a parent entity.
Capture Contact Point Type	<p>The contact point used to capture consent.</p> <ul style="list-style-type: none"> • Email • Mailing Address • Phone

Field	Description
	<ul style="list-style-type: none"> • Social • Web
Capture Date	The date when consent was captured.
Capture Source	The way you captured consent. For example, a website or online form.
Contact Point	The contact point record, such as for phone or email, that you want to associate this consent with.
Data Use Purpose	The data use purpose record that you want to associate this consent with.
Double Consent Capture Date	The date when double opt-in was captured. Double opt-in is captured when the customer confirms for a second time that they want to give consent.
Effective From	The date when consent starts.
Effective To	The date when consent ends.
Engagement Channel Type	ID of the engagement channel record through which the customer is consenting to be contacted.
Party Role	The record, based on the individual object you want to associate consent with.
Privacy Consent Status	<p>Whether the customer associated with this record agrees to this form of contact.</p> <ul style="list-style-type: none"> • Not Seen • Seen • Opt In • Opt Out

Fields in Contact Point Email Records

Fields in contact point email records store multiple email records related to an individual.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

Field	Description
Active from Date	The date when the contact's email became active.
Active to Date	The date when the contact's email is no longer active.
Best Time to Contact End Time	The end time for when the customer prefers to be contacted.
Best Time to Contact Start Time	The start time for when the customer prefers to be contacted.
Best Time to Contact Timezone	The timezone for when it's best to contact the customer.
Email Address	The email address of the contact.
Email Domain	The domain of the contact's email, which is everything after the @ sign.
Email Latest Bounce Date Time	The date and time when an email failed to reach its recipient.
Email Latest Bounce Reason Text	The reason why the email didn't reach its recipient.
Email Mail Box	A subset of the contact's email, which is everything before the @ sign.
Is Primary	Indicates whether a contact's email is their primary email (true) or not (false).
Name	The email of the contact.
Parent	The ID of the contact's parent. Only an individual or account can be a contact's parent.
Preference Rank	Specify how this email ranks in terms of preference among the contact's other emails.
Usage Type	Specify the usage type of this email. For instance, whether it's a work email or a temporary email. Possible values are:

Field	Description
	<ul style="list-style-type: none"> • Home • Temp • Work

Fields in Contact Point Phone Records

Fields in contact point phone records store multiple phone records related to an individual.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

Field	Description
Active from Date	The date when the contact's phone number became active.
Active to Date	The date when the contact's phone number is no longer active.
Area Code	The area code of the phone number's location for the contact.
Best Time to Contact End Time	The end time for when the customer prefers to be contacted.
Best Time to Contact Start Time	The start time for when the customer prefers to be contacted.
Best Time to Contact Timezone	The timezone for when it's best to contact the customer.
Extension Number	The phone number extension for the contact.
Formatted International Phone Number	The internationally recognized format for the contact's phone number.
Formatted National Phone Number	The nationally recognized format for the contact's phone number.
Is Business Phone	Indicates whether a contact's phone number is a business number (true) or not (false).
Is Fax Capable	Indicates whether a contact's phone number is a fax number (true) or not (false).

Field	Description
	fax number (true) or not (false).
Is Personal Phone	Indicates whether a contact's phone number is a personal number (true) or not (false).
Is Primary	Indicates whether a contact's phone number is their primary phone number (true) or not (false).
Is SMS Capable	Indicates whether a contact's phone number can receive text messages (true) or not (false).
Name	The phone number of the contact.
Parent	The ID of the contact's parent. Only an individual or account can be a contact's parent.
Phone Type	<p>The type of phone number for the contact. Possible values are:</p> <ul style="list-style-type: none"> • Home • Mobile
Preference Rank	Specify how this phone number ranks in terms of preference among the contact's other phone numbers.
Telephone Number	The phone number for the contact.
Usage Type	<p>Specify the usage type of this phone number. For instance, whether it's a work phone or a temporary phone. Possible values are:</p> <ul style="list-style-type: none"> • Home • Temp • Work

Fields in Contact Point Type Consent Records

Contact point type consent records let you enter details about how a customer has agreed to be contacted by your company. For example, you can indicate that a customer consented to a specific contact point like email, but not phone calls and mail.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

Field	Description
Business Brand	Represents a unique brand for a business that belongs to a parent entity.
Capture Contact Point Type	<p>The contact point used to capture consent.</p> <ul style="list-style-type: none"> • Email • Mailing Address • Phone • Social • Web
Capture Date	The date when consent was captured.
Capture Source	The way you captured consent. For example, a website or online form.
Contact Point Type	<p>The contact method you want to apply consent to.</p> <ul style="list-style-type: none"> • Email • Mailing Address • Phone • Social • Web
Data Use Purpose	The data use purpose record that you want to associate this consent with.
Double Consent Capture Date	Date when double opt-in was captured. Double opt-in is captured when the customer confirms for a second time that they want to give consent.
Data Use Purpose	The data use purpose record that you want to associate this consent with.
Effective From	The date when consent starts.
Effective To	The date when consent ends.
Engagement Channel Type	ID of the engagement channel record through which the customer is consenting to be contacted.

Field	Description
Name	The name of the contact point type consent record.
Party	The record based on the Individual object you want to associate consent with.
Party Role	The party role record that you want to associate this consent with.
Privacy Consent Status	<p>Whether the individual associated with this record agrees to this form of contact.</p> <ul style="list-style-type: none"> • Not Seen • Seen • Opt In • Opt Out

Fields in Data Use Purpose Records

Represents a category that defines the reason for contacting an individual. For example, billing, marketing, or surveys.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

Field	Description
Can Data Subject Opt Out	Indication of whether an individual can decline contact for the described purpose.
Description	Indicates the purpose for contacting a customer.
Legal Basis	The legal basis record associated with this purpose.
Name	The reason for contacting an individual. For example, “billing” or “marketing.”

Fields in Data Use Legal Basis Records

Represents the legal basis for contacting an individual. For example, billing or contract.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

Field	Description
Description	Description of the data use legal basis.
Name	The name for the legal basis. For example, “billing” or “contract.”
Source	The source of the legal basis. For example, the URL of a contract.

Fields in Party Consent Records

Represents consent preferences for an individual.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions.

Field	Description
Action	The action that the individual is consenting to, such as collecting data.
Business Brand	Represents a unique brand for a business that belongs to a parent entity.
Capture Contact Point Type	<p>The contact point used to capture consent.</p> <ul style="list-style-type: none"> • Email • Mailing Address • Phone • Social • Web
Capture Date	The date when consent was captured.
Capture Source	The way you captured consent. For example, a website or online form.
Double Consent Capture Date	The date when double opt-in was captured.

Field	Description
	Double opt-in is captured when the customer confirms for a second time that they want to give consent.
Effective From	The date when consent starts.
Effective To	The date when consent ends.
Name	The name of the contact point type consent record.
Party	The record based on the Individual object you want to associate consent with.
Party Role	The party role record that you want to associate this consent with.
Privacy Consent Status	Whether the individual associated with this record agrees to this form of contact. <ul style="list-style-type: none"> • Not Seen • Seen • Opt In • Opt Out

Tracking Customer Requests for Data Privacy Updates

If you store certain data privacy preferences in data privacy records based on the Individual object, track customers' requests so that you can honor their wishes.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions, including partner and customer community users.

To track customers' requests for updates to their data privacy preferences, run an Apex trigger or query records in the SOAP API.

Apex Triggers

From Setup, enter *Individual* in the Quick Find box, and then select **Triggers**. To create a new trigger, select **New**.

To monitor changes to a specific field, customize the trigger.

Example:

```
trigger IndividualUpdated on Individual (after update) {  
    for(String individualId : Trigger.newMap.keySet())  
    {  
        if( Trigger.oldMap.get(individualId).ShouldForget !=  
            Trigger.newMap.get(individualId).ShouldForget )  
        {  
            // Do something when ShouldForget ("Forget this Individual") changes  
        }  
    }  
}
```

To track customers' requests, either query for triggers in the API, or create reports on flagged items using custom report types.

Reports

Use data privacy reports to understand the information stored in data privacy records.

Standard Report: Field History—If your organization tracks field history on data privacy records based on the Individual object, you can report on that information using the individual history report.

SOAP API

Query records for changes to data privacy flags using the SOAP API.

For more details, see [SOAP API: Individual](#)

See Also

[Best Practices for Tracking Data Privacy](#)

[Define Apex Triggers](#)

Best Practices for Tracking Data Privacy

Keep these best practices in mind when storing certain data privacy preferences in data privacy records based on the Individual object.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions, including partner and customer community users.

Encrypting Personal Data

The **Name** field in data privacy records is personal data. Salesforce Shield customers can use [Shield to encrypt this field](#). If you've added custom fields that contain personal data, consider encrypting them.

Helping Your Users Decode Individual

After you add the **Individual** field to your page layouts, consider renaming the field to something meaningful for your users. (*Example:* Manage data privacy or Track customer consent)

Keeping Renamed Contacts, Leads, and Person Accounts Consistent with Data Privacy Record Names

Renaming a contact, lead, or person account doesn't update any corresponding privacy data. To keep the name of data privacy records up to date, write an Apex trigger.

Converting Leads to Existing Contacts

When you convert a lead, you can decide which data privacy record you maintain between the converted lead and the contact.

To Maintain the Data Privacy Record From the	You Need To
Contact	Sit back and do nothing.
Lead	Edit the contact's IndividualId to replace the ID with the one from the converted lead. If your contact doesn't include the ID before the conversion, the conversion process populates the field from the converted lead.

Creating Community Users from Contacts with Associated Data Privacy Records

To create a new community member from a contact record, first disable **Don't Process** and **Forget this Individual** in the associated data privacy record.

See Also

[Lead Conversion Field Mapping](#)

Track Certain Data Privacy Preferences for Leads and Contacts Already in Salesforce

Create data privacy records based on the Individual object for leads and contacts already in Salesforce using scripts.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions, including partner and customer community users.

These scripts create unique data privacy records for each lead and contact. Keep in mind, if you already have any data privacy records for your leads and contacts, this script creates duplicate records.

Create Data Privacy Records for Contacts

Create data privacy records and link them to contacts already in Salesforce when you run this script.

```
global class CreateIndividualFromContact implements Database.Batchable<sObject> {
    global Database.Querylocator start(Database.BatchableContext BC) {
        //Query to fetch contacts that don't have Individual created. You may modify the query to add custom fields
        //Please add "IsPersonAccount = false" condition to below query to exclude person accounts
        return Database.getQueryLocator('Select FirstName, LastName, Salutation from Contact where IndividualId = NULL');
    }

    global void execute(Database.BatchableContext BC, List<Contact> contactList) {
        Map<Id, Individual> individualRecordsToCreate = new Map<Id, Individual>();
        for(Contact con : contactList) {
            individualRecordsToCreate.put(con.Id, new Individual(FirstName = con.FirstName, LastName = con.LastName, Salutation=con.Salutation));
        }
        insert individualRecordsToCreate.values();
        for(Contact con : contactList) {
            con.IndividualId = individualRecordsToCreate.get(con.Id).Id;
        }
        update contactList;
    }
}
```

```
global void finish(Database.BatchableContext BC) {}  
}
```

Create Data Privacy Records for Leads

Create data privacy records and link them to leads already in Salesforce when you run this script.

```
global class CreateIndividualFromLead implements Database.Batchable<sObject> {  
  
    global Database.Querylocator start(Database.BatchableContext BC) {  
        //Query to fetch non-converted Leads that don't have Individual create  
        d. You may modify the query to add custom fields  
        return Database.getQueryLocator('Select FirstName, LastName, Salutation  
        from Lead where IsConverted = false and IndividualId = NULL');  
    }  
  
    global void execute(Database.BatchableContext BC, List<Lead> leadList) {  
        Map<Id, Individual> individualRecordsToCreate = new Map<Id, Individual>();  
        for(Lead l : leadList) {  
            individualRecordsToCreate.put(l.Id, new Individual(FirstName = l.F  
irstName, LastName = l.LastName, Salutation=l.Salutation));  
        }  
        insert individualRecordsToCreate.values();  
        for(Lead l : leadList) {  
            l.IndividualId = individualRecordsToCreate.get(l.Id).Id;  
        }  
        update leadList;  
    }  
  
    global void finish(Database.BatchableContext BC) {}  
}
```

Manage Duplicate Data Privacy Records

Keep your records clean and free of duplicates so that you can reach more customers and maintain better relationships with them.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions, including partner and customer community users.

You can work with duplicate data privacy records individually or in bulk. You can't merge duplicate data privacy records automatically.

You can create a custom matching rule for the Individual object to identify duplicate data privacy records and a custom duplicate rule to handle the duplicates.

The Individual object doesn't include standard matching and duplicate rules, so you must create your own custom matching and duplicate rules to manage your data privacy duplicates.

[Merge Duplicate Data Privacy Records in Lightning Experience](#)

Merge duplicate data privacy records based on the Individual object in Lightning Experience.

See Also

[Manage Duplicates One at a Time](#)

[Manage Duplicates Globally](#)

[Duplicate Detection and Handling Process](#)

[Customize Duplicate Management](#)

Merge Duplicate Data Privacy Records in Lightning Experience

Merge duplicate data privacy records based on the Individual object in Lightning Experience.

REQUIRED EDITIONS

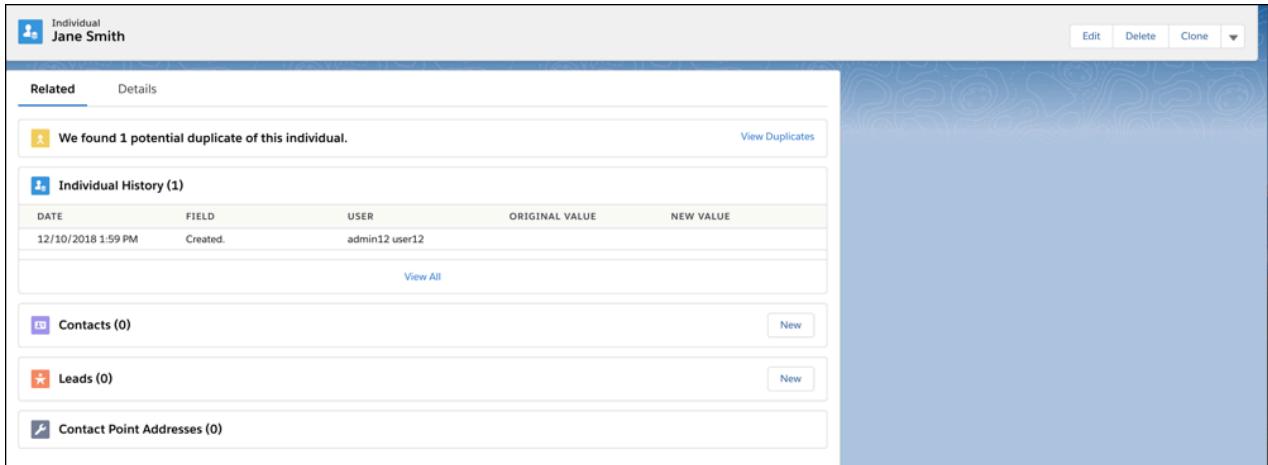
Available in: Lightning Experience

Available in: all editions, including partner and customer community users.

USER PERMISSIONS NEEDED

To merge data privacy records based on the Individual object:	Delete on individuals
--	-----------------------

1. Choose a data privacy record based on the Individual object. A message tells you if duplicates exist for that record. To see them, click **View Duplicates**.



2. Choose up to three data privacy records to merge. Click **Next**.

Potential Duplicate Records

View Duplicates

To merge duplicates, choose up to 3 individuals. Then click Next and choose the fields to keep.

▼ Individuals (2)

NAME	ALIAS	CREATED BY ALIAS	LAST MODIFIED BY ALIAS	LAST MODIFIED DATE
Jane Smith	auser	auser	auser	12/10/2018 1:59 PM
<input checked="" type="checkbox"/> Jane Smith	<input checked="" type="checkbox"/> auser	<input checked="" type="checkbox"/> auser	<input checked="" type="checkbox"/> auser	<input checked="" type="checkbox"/> 12/10/2018 1:59 PM

Next

3. Choose one data privacy record as the master, and choose the field values that you want to keep. Click **Next**.

Potential Duplicate Records

Compare Individuals

When you merge, the master record is updated with the values you choose, and relationships to other items are shifted to the master record.

	Jane Smith Select All	Jane Smith Select All
MASTER RECORD ⓘ	<input checked="" type="radio"/> Use as master	<input type="radio"/> Use as master
IS HOMEOWNER	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DON'T PROCESS	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DON'T MARKET	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CREATED DATE ⓘ	12/10/2018 1:59 PM	12/10/2018 1:59 PM
LAST MODIFIED DATE	12/10/2018 1:59 PM	12/10/2018 1:59 PM

Showing fields with different values. [Show All Fields](#)

Back Next

4. Confirm your choices and merge.

Potential Duplicate Records

Confirm Merge

Are you sure you want to merge these records?

You're about to merge these individuals. You can't undo merging.

Back Merge Individuals

Classify Sensitive Data to Support Data Management Policies

Record data sensitivity and compliance categorization at the field level. Data classification can be used to guide decisions around access, reporting, and data compliance.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **All** editions

USER PERMISSIONS NEEDED

To edit or view data classification fields:

Customize Application or Modify Data
Classification

Set Up Data Classification Metadata

Customize your data classification settings to meet your organization's needs. You can also upload or download data classification values.

Data Classification Metadata Fields

Record the data owner, field usage, data sensitivity, and compliance categorization for any standard or custom object field. You can also access data classification metadata in the Salesforce API and Apex.

Create Reports from Data Classification Metadata

Run reports to help meet your data management policies.

Set Up Data Classification Metadata

Customize your data classification settings to meet your organization's needs. You can also upload or download data classification values.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **All** editions

USER PERMISSIONS NEEDED

To edit or view data classification fields:

Customize Application or Modify Data
Classification

Data classification is automatically enabled in all orgs.

1. Select whether to use default data sensitivity levels.
 - a. From Setup, in the Quick Find box, enter *Data Classification Settings*, and then select **Data Classification Settings**.
 - b. Select or deselect **Use default data sensitivity level**. Some fields contain data sensitivity values by default. Those values are visible in the UI after you enable them. For example, the email field has a confidential value. When you enable that checkbox, those default values are applied and are then visible in the UI.
2. Edit data classification values.
 - a. From Setup, select **Object Manager**, and then select the object you want to edit.
 - b. Select **Fields & Relationships** from the sidebar. Select the field where you want to set up data classification, and click **Edit**.
 - c. Select the value for each metadata field from the dropdown lists.
 - d. Click **Save**.



Note To edit data classification values for fields that aren't available in the Object Manager, use the CustomField Metadata API or Data Classification Upload tool.

3. (Optional) Edit Data Sensitivity picklist values.
 - a. From Setup, in the Quick Find box, enter *Data Classification Settings*, and then select **Data Classification Settings**.
 - b. Select **Edit Data Sensitivity Picklist Values**
 - c. Add, delete, rename, or reorder an existing picklist value.
You can also add a **Description** or indicate that fields with this picklist value contain data highly sensitive to your company by selecting **High-risk level**.
4. (Optional) Edit Compliance Categorization picklist values.
 - a. From Setup, in the Quick Find box, enter *Data Classification Settings*, and then select **Data Classification Settings**.
 - b. Select **Edit Compliance Categorization Picklist Values**
 - c. Add, delete, rename, or reorder an existing picklist value.
5. (Optional) Upload data classification information in a CSV file.
 - a. From Setup, in the Quick Find box, enter *Data Classification Upload*, and then select **Data Classification Upload**.
 - b. Click **Choose File**. A CSV file can contain a maximum of 10,000 rows. Each row must include the object name, field name, and data sensitivity level.
6. (Optional) Download data classification information in a CSV file.
 - a. From Setup, in the Quick Find box, enter *Data Classification Download*, and then select **Data Classification Download**.
 - b. Click **Download**. The maximum size of the CSV file is 5 MB.
Only fields that have an associated data sensitivity value are included in the file. The file includes the first 2,000 records for each data sensitivity value.

You can update multiple fields at once using the [CustomField](#) Metadata API.

Data Classification Metadata Fields

Record the data owner, field usage, data sensitivity, and compliance categorization for any standard or custom object field. You can also access data classification metadata in the Salesforce API and Apex.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **All** editions

USER PERMISSIONS NEEDED

To edit data classification fields:

Customize Application or Modify Data Classification

You can use the data classification metadata fields on most standard and custom objects. Before you

begin, review these limitations.

- Compliance categorizations don't appear on fields suffixed with __pc when you query the Person Account object. While __pc fields appear in Person Account queries, they reside on the Contact object. To see their compliance categorization, query the Contact object instead. The same limitation applies whether you're using Tooling API (FieldDefinition) or Metadata API (CustomField). Learn more about [Person Accounts](#) and their relationship to Contacts.
- If you're working in Object Manager, these metadata fields aren't available for the User object: Data Owner, Data Sensitivity Level, and Field Usage. To apply them to the User object, use the [CustomField Metadata API](#) or the Data Classification Upload tool.

Field	Description
Compliance Categorization	<p>The compliance acts, definitions, or regulations that are related to the field's data. Default values:</p> <ul style="list-style-type: none"> • CCPA–California Consumer Privacy Act • COPPA–Children's Online Privacy Protection Act • GDPR–General Data Protection Regulation • HIPAA–Health Insurance Portability and Accountability Act • PCI–Payment Card Industry • PersonalInfo–Personal information. For use with the Enhanced Personal Information Management feature. Only available if Enhanced Personal Information Management and Digital Experiences are enabled. • PII–Personally Identifiable Information <p>The field corresponds to the ComplianceGroup field on the FieldDefinition Tooling API.</p>
Data Owner	<p>The person or group associated with this field. The data owner understands the importance of the field's data to your company and might be responsible for determining the minimum data sensitivity level.</p> <p>The field corresponds to the BusinessOwnerId field on the FieldDefinition Tooling API.</p>
Data Sensitivity Level	<p>The sensitivity of the data contained in this field. Default values:</p>

Field	Description
	<ul style="list-style-type: none"> • Public—Available to the public to view but not alter. • Internal—Available to company employees and contractors. This data must not be shared publicly, but it can be shared with customers, partners, and others under a non-disclosure agreement (NDA). • Confidential—Available to an approved group of employees and contractors. This data isn't restricted by law, regulation, or a company master service agreement (MSA). It can be shared with customers, partners, and others under an NDA. • Restricted—Available only to an approved group of employees and contractors. This data is likely restricted by law, regulation, an NDA, or a company MSA. • MissionCritical—Available only to a small group of approved employees and contractors. Third parties who are given access could be subject to heightened contractual requirements. This data is almost always restricted by law, regulation, an NDA, or a company MSA. <p>The field corresponds to the SecurityClassification field on the FieldDefinition Tooling API and the FieldSecurityClassification SOAP API.</p>
Field Usage	<p>Tracks whether the field is in use. Default values:</p> <ul style="list-style-type: none"> • Active—In use and visible. • DeprecateCandidate—Planned for deprecation and no longer in use. • Hidden—Not visible and possibly planned for deprecation. Use with caution. <p>Changing the value of Field Usage doesn't hide or expose the field.</p> <p>The field corresponds to the BusinessStatus field on the FieldDefinition Tooling API.</p>

You can customize the values for the Compliance Categorization, Data Sensitivity Level, and Field Usage fields.

- To edit the Compliance Categorization values, select **Edit Compliance Categorization Picklist Values** on the Data Classification Settings Setup page or update the **ComplianceGroup** picklist using the [StandardValueSet](#) Metadata API type.
- To edit the Data Sensitivity Level values, select **Edit Data Sensitivity Picklist Values** on the Data Classification Settings Setup page or update the **SecurityClassification** picklist using the [StandardValueSet](#) Metadata API type.
- To edit the Field Usage values, update the **FieldBusinessStatus** picklist using the [StandardValueSet](#) Metadata API type.

 **Note** You can also access data classification metadata by querying your Salesforce data. For example, this sample query retrieves values for all data classification metadata fields in account and lead records.

```
SELECT Id, DeveloperName, Description, BusinessOwnerId, BusinessStatus, SecurityClassification  
FROM FieldDefinition  
WHERE EntityDefinitionId in ('Account', 'Lead')
```

Create Reports from Data Classification Metadata

Run reports to help meet your data management policies.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **All** editions

USER PERMISSIONS NEEDED

To edit or view data classification fields:

Customize Application or Modify Data Classification

1. To expose the metadata fields, [create a custom report type](#). From Setup, enter *Report Types* in the Quick Find box, then select **Report Types**.
2. Click **New Custom Report Type**.
3. For Primary Object, select **Entity Definition**.
4. Fill in the remaining fields and click **Next**.
5. [Add a child object](#). Click the box under the primary object and select **Field Definitions**.
6. Click **Save**.
7. Exit Setup and click the Reports tab.
8. Build a report in [Lightning Experience](#) or [Classic](#) using the report type you made.

Design Your Own Data Model With Schema Builder

Schema Builder provides a dynamic environment for viewing and modifying all the objects and relationships in your app. This greatly simplifies the task of designing, implementing, and modifying your data model, or schema. Schema Builder is enabled by default.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: All Editions

Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

You can view your existing schema and interactively add new custom objects, custom fields, and relationships, simply by dragging and dropping. Schema Builder automatically implements the changes and saves the layout of your schema any time you move an object. This eliminates the need to click from page to page to find the details of a relationship or to add a new custom field to an object in your schema.

Schema Builder provides details like the field values, required fields, and how objects are related by displaying lookup and master-detail relationships. You can view the fields and relationships for both standard and custom objects.

To access Schema Builder, from Setup, enter *Schema Builder* in the Quick Find box, then select **Schema Builder**.

Schema Builder lets you add the following to your schema:

- Custom objects
- Lookup relationships
- Master-detail relationships
- All custom fields except: Geolocation

See Also

[Create Objects with Schema Builder](#)

[Custom Field Types](#)

[Change Sets](#)

Create Objects with Schema Builder

Custom objects are objects that you create to store information that's specific to your company or industry. You can create them via Setup or in the Schema Builder.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To create new custom objects in Schema Builder: Customize Application

1. Click the **Elements** tab.
2. Click **Object** and drag it onto the canvas.
3. Enter information to define your object. For a list of object definitions, see [Schema Builder Custom Object Definition](#).
4. Click **Save**.

See Also

[Schema Builder Custom Object Definition](#)

Create Fields with Schema Builder

Add new custom fields to an object right inside Schema Builder.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To create fields in Schema Builder: Customize Application

1. From Setup, enter *Schema Builder* in the Quick Find box, then select **Schema Builder**.
2. Click the **Elements** tab.
3. Click a field and drag it onto an object on the canvas.
4. Enter a field label.

Salesforce populates Field Name using the field label. Field Name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.

When creating a custom field, ensure that its name and label are unique for that object. If a standard and custom field have identical names or labels, the merge field displays the custom field value. If two custom fields have identical names or labels, the merge field can display an unexpected value.

If you create a field label called *Email* and a standard field labeled *Email* exists, the merge field is unable to distinguish between the fields. Adding a character to the custom field name makes it

unique. For example, *Email12*.

5. Enter a description of the custom field.
6. Enter help text to detail the purpose and function of a custom field.
7. Enter a default value to automatically insert a value of a custom field when a new record is created.
8. Depending on the custom field type you chose, enter any remaining field attributes.
9. Click **Save**.

Any field you add through Schema Builder isn't automatically added to the object's page layout. You must edit the page layout to specify where the field should be displayed.

 **Note** By default, the field level security for custom fields is set to visible and editable for internal profiles. Fields that aren't normally editable, such as formulas and roll-up summary fields, are visible and read-only. To manage permissions of a custom field, click the element name or label and select **Manage Field Permissions**. Use the dialog box that appears to manage the field's visibility and writeability for all standard and custom profiles.

Delete Custom Objects with Schema Builder

You can delete the custom objects that you no longer need by using Schema Builder.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To delete custom objects in Schema Builder: Customize Application

Schema Builder displays a list of side effects when you try to delete a custom object. Be sure you're ready to accept these side effects before finalizing the deletion.

1. Click  on the custom object's icon.
2. Select **Delete Object....**
A dialog box displays that explains the side effects of deleting an object. Read this information carefully.
3. If you accept the conditions, select **Yes, I want to delete the custom object**.
4. Click **Delete**.

Delete Custom Fields with Schema Builder

Avoid custom field clutter by using Schema Builder to delete custom fields that you no longer need.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To delete custom fields in Schema Builder: Customize Application

Schema Builder displays a list of side effects when you try to delete a custom field. Be sure you're ready to accept these side effects before finalizing the deletion.

1. Right-click on the custom field.
2. Select **Delete Field...**.
A dialog box displays that explains the side effects of deleting a custom field. Read this information carefully.
3. If you accept the conditions, select **Yes, I want to delete the custom field.**
4. Click **Delete**.

Schema Builder Custom Object Definition

These fields are key to defining your custom object.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

! **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Field	Description
Label	A name used to refer to the object in any user interface pages.
Plural Label	The plural name of the object. If you create a tab for this object, this name is used for the tab.
Starts with a vowel sound	If it's appropriate for your org's default language, check if your label should be preceded by "an" instead of "a".
Object Name	A unique name used to refer to the object when using the API. In managed packages, this unique name prevents naming conflicts on package

Field	Description
	installations. The Object Name field can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
Description	An optional description of the object. A meaningful description helps you remember the differences between your custom objects when you're viewing them in a list.
Record Name	The name used in page layouts, list views, related lists, and search results.
Data Type	The type of field (text or auto-number) for the record name. Records that have unique IDs instead of names use auto-numbers. An auto-number is a unique number assigned automatically. It's always a read-only field.
Allow Reports	<p>Makes the data in the custom object records available for reporting purposes.</p> <p>To create reports on custom objects, choose the Other Reports report type category, unless the custom object has a relationship with a standard object. When the custom object has a master-detail relationship with a standard object or is a lookup object on a standard object, select the standard object for the report type category instead.</p>
Allow Activities	Allows users to associate tasks and scheduled calendar events related to the custom object records.
Track Field History	Enables your org to track changes to fields on the custom object records, such as who changed the value of a field, when it was changed, and what the value of the field was before and after the edit. History data is available for reporting, so users can easily create audit trail reports when this feature is enabled.
Enable Divisions	If your org has divisions enabled, select this option to enable the custom object for divisions.

Field	Description
	Divisions group records for simplified search results, list views, reports, and other areas within Salesforce. Salesforce adds a Division field to the custom object. If the custom object is the master in a master-detail relationship, custom objects on the detail side also get the Division field and inherit their division from the master record.
Available for Customer Portal	This option makes the custom object available through the Salesforce Customer Portal.
Namespace Prefix	In a packaging context, a namespace prefix is a one to 15-character alphanumeric identifier that distinguishes your package and its contents from packages of other developers on AppExchange. Namespace prefixes are case-insensitive. For example, ABC and abc aren't recognized as unique. Your namespace prefix must be globally unique across all Salesforce orgs. It keeps your managed package under your control exclusively.
Deployment Status	Indicates whether the custom object is visible to other users.
Add Notes & Attachments...	<p>Allows users to attach notes and attachments to custom object records. You can attach external documents to any object record, in much the same way that you can add a PDF or photo as an attachment to an email.</p> <p>This option is only available when you're creating an object.</p>

Schema Builder Considerations

Keep these items in mind when working with Schema Builder.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: All Editions

- Schema Builder saves the layout of your schema anytime you move an object.

- If your schema contains many objects and fields, loading can take a long time. Click **Hide Relationships** to improve Schema Builder performance.
- When creating a custom field, ensure that its name and label are unique for that object.
 - If a standard and custom field have identical names or labels, the merge field displays the custom field value.
 - If two custom fields have identical names or labels, the merge field can display an unexpected value.

If you create a field label called *Email* and a standard field labeled *Email* exists, the merge field is unable to distinguish between the fields. Adding a character to the custom field name makes it unique. For example, *Email12*.

- You can't export your schema from Schema Builder (for example, to use the schema in another org).
- When you click **Auto-Layout**, you can't undo it.
- Objects created outside of Schema Builder, such as through an app or the API, don't automatically display on the canvas. Select the checkbox for the object created outside Schema Builder to display it on the canvas.
- By default, the field-level security for custom fields is set to visible and editable for internal profiles—those not cloned from Partner User or Customer Portal Manager. Fields that aren't normally editable, such as formulas and roll-up summary fields, are visible and read only.
- You can't save the level of zoom when closing Schema Builder.

Create Custom Settings

Use custom settings to create custom sets of data, or to create and associate custom data for an org, profile, or user.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience.

Available in: **Group, Professional, Developer, Enterprise, Performance, Unlimited, and Database.com** Editions.

Packages aren't available in **Database.com**.

USER PERMISSIONS NEEDED

Customize Application

Custom settings are similar to custom objects in that they let you customize org data. Unlike custom objects, which have records based on them, custom settings let you utilize custom data sets across your org. Custom settings also let you distinguish particular users or profiles based on custom criteria.

Custom settings data is exposed in the application cache, which enables efficient access without the cost of repeated queries to the database. This data can then be used by formula fields, validation rules, flows, Apex, and SOAP API

 **Note** If you're thinking of using List Custom Settings, consider using [Custom Metadata Types](#) instead. Unlike List Custom Settings, you can migrate the records of Custom Metadata Types using Packages or Metadata API tools.

1. Review the protection and privacy options.
2. Create the custom setting.
3. Add fields and data.
4. Reference the custom setting data in your application using formula fields, validation rules, Apex, or SOAP API.

 **Example** These examples illustrate how you can use custom settings.

- A shipping application requires users to fill in the country codes for international deliveries. By creating a list setting of all country codes, users have quick access to this data without needing to query the database.
- An application calculates and tracks compensation for its sales reps, but seniority determines commission percentages. By creating a hierarchy setting, the administrator can associate a different commission percentage for each profile in the sales organization. Within the application, one formula field can then be used to calculate compensation for all users. The personalized setting at the profile level inserts the correct commission percentage.
- An application displays a map of account locations, the best route to take, and traffic conditions. This information is useful for sales reps, but account executives only want to see account locations. By creating a hierarchy setting with custom checkbox fields for route and traffic, you can enable this data for just the Sales Rep profile.

[Protection and Privacy Options for Custom Settings](#)

Manage the visibility and permissions of custom settings.

[Define Custom Settings](#)

Create custom sets of data.

[View and Edit Custom Settings](#)

After you create a custom setting, you can view the details of the custom setting, manage the custom setting, and add fields.

[View Custom Settings Usage and Data](#)

View the percentage of custom settings data your org has used.

[Grant Permissions on Custom Settings](#)

By default, access to custom settings is limited through the **Restrict access to custom settings** org-wide preference. Admins can grant API Read access through profiles and permission sets to users without the Customize Application permission.

[Grant Read Access to All Custom Settings](#)

Admins with the Customize Application permission can grant API read access to all custom settings.

[Access Custom Settings with Code](#)

You can access custom settings from Apex, SOAP API, and formulas.

Custom Settings Limits and Considerations

When working with custom settings, be aware of the following considerations and limits on the amount of cached data.

Protection and Privacy Options for Custom Settings

Manage the visibility and permissions of custom settings.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience.

Available in: **Group, Professional, Developer, Enterprise, Performance, Unlimited, and Database.com** Editions.

Packages aren't available in **Database.com**.

Packages

Only custom settings definitions are included in packages, not data. To include data, populate the custom settings using a standard Apex or API script run by the subscribing organization after they install the package.

When a custom setting is contained in a managed package, and the visibility is set to Protected, access is allowed only through the Apex code that is part of that managed package. Subscriber organizations can't directly access, read, or modify the protected custom settings.

-  **Note** Protected custom settings in an unmanaged package behave like public custom settings. Make sure that secrets, personally identifying information, or any private data are stored in protected custom metadata types that are installed as part of a managed package. Outside of a managed package, use named credentials or encrypted custom fields to store secrets like OAuth tokens, passwords, and other confidential material.

Visibility

You can create protected custom settings in developer and scratch orgs. The options for custom settings are.

- Protected—Custom settings in a managed package are visible through Apex and formulas within the same package and namespace. However, they are not visible to subscribing organizations through Apex and API. Custom settings contained in an unmanaged package behave like public custom settings and don't provide protection.
- Public—Regardless of the type of package (managed or unmanaged), the following have access:
 - Apex

- Formulas
- Flows
- API for users with Customize Application permission or permissions granted through profiles or permission sets.

Schema Settings

The schema setting options for custom settings are.

- Restrict access to custom settings—This org-wide preference is enabled by default and limits access to custom setting values. Admins with Customize Application permission can grant Read access to users through profiles and permission sets using the Custom Setting Definitions or View All Custom Settings permissions.
- Enable SOSL on custom settings—Custom settings values are not returned in Salesforce Object Search language (SOSL) queries. If your Apex operations require this functionality, enable this option.

Grant Permissions on Profiles or Permission Sets

The Restrict access to custom settings org-wide preference is enabled by default and Read access to custom settings must be explicitly granted.

Admins with Customize Application permission can grant Read permission through profiles and permission sets to users without the Customize Application permission.

- To grant permission to specific custom settings, use the Custom Setting Definitions permission.
- To grant permission to all custom settings, use the View All Custom Settings permission.

Behavior of Apex, Visualforce, and Aura

There are different execution code modes within Salesforce that affect the accessibility of custom settings.

Apex code that is run in system mode ignores user permissions and your Apex code is given access to all objects and fields. Object permissions, field-level security, and sharing rules aren't applied for the current user. Running in system mode ensures that the code doesn't fail because of hidden fields or objects for a user.

In user mode, functionality such as Visualforce Components, Visualforce Email templates, and Aura, is run with respect to the user's permissions and sharing of records.

-  **Note** Functionality that runs in system mode, such as Apex, is not affected by the Restrict access to custom settings org preference. Also, the `with sharing` modifier in the Apex class, doesn't affect query behavior such as, `isAccessible()` and `isCreatable()`. If a field value is retrieved in Apex and assigned to a non-sObject variable, the behavior is the same whether the preference is enabled or not.

When functionality is run in user mode, such as Visualforce Components, Visualforce Email templates, and Aura, you must have permission to access the custom settings. For example, without permission, the fields on Visualforce pages that you don't have access to aren't displayed. The `$Setup` global variable (available in Visualforce and formulas) continues to load values by direct reference (meaning, data that is assigned to an sObject type) regardless of the running user.

 **Example** Consider the following scenario:

- Apex loads a record that is a row included in a variable such as `MySetting__c`.
- What Visualforce displays is `MySetting__c.MyPath__c`.
- Access checks are run when the page is loaded.
- However, the checks are not run in system mode, which is the standard Visualforce behavior. Users without permission to the custom settings can't display the Visualforce page because Visualforce is reinitiating the access check.

In this scenario, if a user isn't allowed permission to the custom setting, there are two workarounds. You can create a string for each object, which can be passed through, or create a wrapper class. Use these options instead of assigning a variable such as `MySetting__c`, then rendering `mySetting.Path__c mySetting.Name`. For example,

```
class DataHolder{  
    public string path {get;set;}  
    public boolean active {get;set;}  
}
```

When you load the rows into a collection, the Visualforce checks are bypassed because the type is a data type instead of an sObject.

Here's an example that includes the `@AuraEnabled` annotation for an Aura or Lightning components controller.

```
class with sharing MyController {  
    @AuraEnabled  
    public static List<My__mdt> thisWillNotWork() {  
        return [select developername from my__mdt];  
    }  
    @AuraEnabled  
    public static List<String> thisWill() {  
        List<String> retVal = new List<String>();  
        for(My__mdt config: [select developername from my__mdt]) {  
            retVal.add(config.DeveloperName);  
        }  
        return retVal;  
    }  
}
```

See Also

- [Grant Permissions on Custom Settings](#)
- [Access Custom Settings with Code](#)
- [Grant Read Access to All Custom Settings](#)

Define Custom Settings

Create custom sets of data.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience.

Available in: **Group, Professional, Developer, Enterprise, Performance, Unlimited, and Database.com** Editions.

Packages aren't available in **Database.com**.

USER PERMISSIONS NEEDED

To manage, create, edit, and delete custom settings: Customize Application

-  **Note** We strongly suggest using custom metadata types instead of custom settings. Unlike list custom settings, you can migrate the records of custom metadata types using second-generation packages or Metadata API tools.

To create a custom setting:

1. From Setup, enter *Custom Settings* in the Quick Find box, then select **Custom Settings**.
2. Click **New**.

 **Note** A  icon indicates that the custom setting is in an installed managed package. You can't edit or delete a protected custom setting installed from a managed package.
3. Give the custom setting a label.
Enter the label displayed in the application.
4. Define the object name.
Enter the name to be used when the custom setting is referenced by formula fields, validation rules, Apex, or SOAP API.

 **Note** Salesforce recommends using ASCII for the Object Name. The name can't exceed 38 ASCII characters. If you use double byte, there are additional limits on the number of characters allowed.

5. Define the setting type.
Select a type of List or Hierarchy. After you save a custom setting, you can't change this value.
 - List—Defines application-level data, such as country codes or state abbreviations, and provides a reusable set of static data that can be accessed across your organization. If you use a particular set

of data frequently within your application, putting that data in a list custom setting streamlines access to it. Data in list settings does not vary by profile or user, but it is available organization-wide. Examples of list data include two-letter state abbreviations, international dialing prefixes, and catalog numbers for products. Because the data is cached, access is low-cost and efficient—you don't have to use SOQL queries that count against your governor limits.

- **Hierarchy**—Uses a built-in hierarchical logic that lets you personalize settings for specific profiles or users. The hierarchy logic checks the organization, profile, and user settings for the current user and returns the most specific, or lowest, value. In the hierarchy, settings for an organization are overridden by profile settings, which, in turn, are overridden by user settings.

6. Define the visibility setting.

(Available in developer and scratch orgs) After you save a custom setting, you can't change this value.

- **Protected**—If the custom setting is contained in a managed package, subscribing organizations can't see the custom setting—it doesn't display as part of the package list. In addition, subscribing organizations can't access the custom setting using Apex or the API. Custom settings can only be accessed by the Apex code that is part of the managed package. If the custom setting is contained in an unmanaged package, they behave like public custom settings.
- **Public**—Regardless of the type of package (managed or unmanaged), the following have access: Apex, formulas, flows, and API for users with Customize Application permission or permissions granted through profiles or permission sets.

7. Enter an optional description of the custom setting. A meaningful description helps you remember the differences between your custom settings when you view them in a list.

8. Click **Save**.

After you create a custom setting, add fields to the custom setting.

Add Custom Settings Fields

After you define custom settings, add fields to them. The custom fields contain the data used by the custom setting.

Create Custom Settings Records

After you define your custom settings and add fields, you can populate the fields with data.

Manage Custom Settings Data

After creating custom setting and adding fields, you can add records, then use the values in these records in your Apex code and validation rules.

Add Custom Settings Fields

After you define custom settings, add fields to them. The custom fields contain the data used by the custom setting.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience.

Available in: **Group, Professional, Developer, Enterprise, Performance, Unlimited, and Database.com Editions.**

Packages aren't available in **Database.com**.

USER PERMISSIONS NEEDED

To manage, create, edit, and delete custom settings: Customize Application

1. From Setup, enter *Custom Settings* in the Quick Find box, then select **Custom Settings**.
2. Click the custom setting that you want to add fields to. (If you just created the custom setting, the Custom Setting Detail page appears.)
3. Click **New**.
4. Select a field type and click **Next**.



Note Record size is based on the maximum field size of each field type, not the actual storage that's used in each field. When adding fields to a custom setting definition, use the appropriate type and specify a length that doesn't exceed what's needed for your data. This action helps you avoid reaching the cached data limit. For example, if you create a US social security number (SSN) field, select the `Text` data type and specify a length of 9. If you select a `Text Area` data type, the field would add 255 characters to the usage count for each record, regardless of the number of characters entered.

5. Enter the details for the field.
6. Confirm the information, and then click **Save** or **Save & New**.

After you add fields, add data, and for hierarchy custom settings, specify the access level.

See Also

[Create Custom Settings Records](#)

[Create Custom Settings Records](#)

After you define your custom settings and add fields, you can populate the fields with data.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience.

Available in: **Group, Professional, Developer, Enterprise, Performance, Unlimited, and Database.com Editions.**

Packages aren't available in **Database.com**.

USER PERMISSIONS NEEDED

Customize Application

You can define one or more data sets. For list custom settings, each data set is named and can be accessed by that name using Apex, formula fields, and so on.

For custom settings that are hierarchies, the access level—user, profile, or organization—determines how you access the data. The user level is the lowest level, so it is used first, unless otherwise specified in your application. For example, you can specify different contact numbers for your application: one for the general user, and one that is only displayed for system administrators.

To add data to custom setting fields:

1. From Setup, enter *Custom Settings* in the Quick Find box, select **Custom Settings**, then click **Manage** next to a custom setting. Or from the detail page for a custom setting, click **Manage**.
2. Click **New** or **Edit** next to an existing data set.
3. Add or change data for custom settings that are lists.
 - a. Specify or change the name for the data set. This name is used by Apex, formula fields, and so on.
 - b. Enter or change data for all fields.
 - c. Click **Save**.
4. Add or change data for custom settings that are hierarchies.
 - a. For the default organization level values, enter or change the data for the fields. The default organization location is automatically populated.
 - b. For profile or user level values, select either **Profile** or **User** from the Location picklist. Enter the name of the profile or user, or use the lookup dialog search. Then enter or change the data for the fields.
 - c. Click **Save**.



Note For a hierarchy custom setting, you can add only one record for a profile or user. Adding two records for the same profile or user results in an error.

See Also

[Manage Custom Settings Data](#)

[Add Custom Settings Fields](#)

Manage Custom Settings Data

After creating custom setting and adding fields, you can add records, then use the values in these records in your Apex code and validation rules.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience.

Available in: **Group, Professional, Developer, Enterprise, Performance, Unlimited, and Database.com Editions.**

Packages aren't available in **Database.com**.

USER PERMISSIONS NEEDED

Customize Application

1. From Setup, enter *Custom Settings* in the Quick Find box, then select **Custom Settings**.

2. Click **Manage** next to a custom setting, or from the detail page for a custom setting.

3. Provide or change values for the custom setting.

If you're managing a list setting:

- To add data to the fields, click **New**.
- To change the name of the data set or to change the data, click **Edit** next to the name of an existing set of data.
- To delete the data set, click **Delete** next to the name of an existing set of data.

If you're managing a hierarchy setting, decide where in the permission hierarchy you want to add default data (organization, profile, or user).

- To add default data at the organization level, click **New** in the Default Organization Level Value section. If data has already been defined for the organization, you can only edit or delete it.
- To add default data at the profile or user level, click **New** in the lower section of the page, near the Setup Owner.

After you define data:

- To change the default data set at the organization level, click **Edit** in the Default Organization Level Value section.
- To delete the data set (only for hierarchical custom settings), click **Delete** in the Default Organization Level Value section.
- To view the data (only for hierarchical custom settings), click **View** next to the name of an existing set of data.

See Also

[Custom Settings Limits and Considerations](#)

View and Edit Custom Settings

After you create a custom setting, you can view the details of the custom setting, manage the custom setting, and add fields.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience.

Available in: **Group, Professional, Developer, Enterprise, Performance, Unlimited, and Database.com Editions.**

Packages aren't available in **Database.com**.

USER PERMISSIONS NEEDED

To manage, create, edit, and delete custom settings: **Customize Application**

From Setup, enter *Custom Settings* in the Quick Find box, then select **Custom Settings**, then click the name of the custom setting you'd like to view. While viewing a custom setting, you can:

- To change a custom setting, click **Edit**.
- To delete a custom setting, click **Delete**.

 **Note** A  icon indicates that the custom setting is in an installed managed package. You can't edit or delete a protected custom setting installed from a managed package.

- To add data to a custom setting, click **Manage**.
- To add fields and data to the custom setting, click **New**.

See Also

[Create Custom Settings Records](#)

[Add Custom Settings Fields](#)

View Custom Settings Usage and Data

View the percentage of custom settings data your org has used.

[View the System Usage Data](#)

View the number of custom objects and custom settings in your org, including installed objects from AppExchange.

[View the Percentage of Custom Settings Data](#)

View the percentage of custom settings data used in your organization, out of an allowed limit.

View the System Usage Data

View the number of custom objects and custom settings in your org, including installed objects from AppExchange.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience.

Available in: **Group, Professional, Developer, Enterprise, Performance, Unlimited, and Database.com** Editions.

Packages aren't available in **Database.com**.

1. From Setup, enter *System Overview* in the Quick Find box, then select **System Overview**.
2. In the Schema section, you see the number of custom object and custom settings:
 - Your Custom Objects and Custom Settings—The number of custom object and custom settings you created. The edition of Salesforce used determines the limit.
 - Total Custom Objects and Total Custom Settings—Total number of custom objects and custom settings in your org, and it includes objects installed from AppExchange managed packages.

View the Percentage of Custom Settings Data

View the percentage of custom settings data used in your organization, out of an allowed limit.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience.

Available in: **Group, Professional, Developer, Enterprise, Performance, Unlimited, and Database.com** Editions.

Packages aren't available in **Database.com**.

1. From Setup, enter *Custom Settings* in the Quick Find box, then select **Custom Settings**.
2. To view the percentage of custom settings data your org has used, click **Get Usage**. Once the usage information is returned, the button no longer appears on the page.
3. To manage a custom setting, click **Manage** next to a previously defined custom setting, and click **View** next to the data set you want to view. (only for hierarchical custom settings.)

Grant Permissions on Custom Settings

By default, access to custom settings is limited through the **Restrict access to custom settings** org-wide preference. Admins can grant API Read access through profiles and permission sets to users without the Customize Application permission.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience.

Available in: **Group, Professional, Developer, Enterprise, Performance, Unlimited, and Database.com**

Editions.

Packages aren't available in **Database.com**.

1. From Setup, enter *Schema Settings* in the **Quick Find** box, and make sure that the Restrict access to custom settings org permission is enabled.
2. Enter *User Management Settings* in the **Quick Find** box, and enable **Enhanced Profile User Interface**.
This setting provides a uniform and streamlined interface, but isn't a requirement for granting permissions.
3. Enter *Profiles or Permission Sets* in the **Quick Find** box.
4. Click the name of the profile or permission set that you want to edit.
5. Click **Custom Setting Definitions**.
6. Click **Edit**.
7. Add the custom setting to the **Enabled Custom Setting Definitions** list.
8. Click **Save**.

Grant Read Access to All Custom Settings

Admins with the Customize Application permission can grant API read access to all custom settings.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience.

Available in: **Group, Professional, Developer, Enterprise, Performance, Unlimited, and Database.com Editions.**

Packages aren't available in **Database.com**.

USER PERMISSIONS NEEDED

To grant access to custom settings: Customize Application

1. From Setup, enter *Schema Settings* in the Quick Find box, and make sure that the Restrict access to custom settings org permission is enabled.
2. Enter *User Management Settings* in the Quick Find box, and enable **Enhanced Profile User Interface**.
This setting provides a uniform and streamlined interface, but isn't a requirement for granting permissions.
3. Enter *Profiles or Permission Sets* in the Quick Find box.
4. Click the name of the profile or permission set that you want to edit.
5. Click **System permissions**.
6. Check the **View All Custom Settings** permission.

7. Click **Save**.

Access Custom Settings with Code

You can access custom settings from Apex, SOAP API, and formulas.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience.

Available in: **Group, Professional, Developer, Enterprise, Performance, Unlimited, and Database.com** Editions.

Packages aren't available in **Database.com**.

 **Note** Formulas include: flows, workflow rules, approval processes, validation rules, formula fields, and Process Builder processes.

Here are some sample code segments:

Formula Fields

Formula fields only work for hierarchy custom settings; they can't be used for list custom settings.

```
{ !$Setup.CustomSettingName__c.CustomFieldName__c }
```

Apex

Apex can access both custom setting types.

Samples for List Custom Settings

When you add data to a custom setting, name each data set so that you can distinguish them. The following returns a map of custom settings data. The `getAll` method returns values for all custom fields associated with the list setting.

```
Map<String_dataset_name, CustomSettingName__c> mcs = CustomSettingName__c.getAll();
```

The following example uses the `getValues` method to return all the field values associated with the specified data set. This method can be used with list and hierarchy custom settings, using different parameters.

```
CustomSettingName__c mc = CustomSettingName__c.getValues(data_set_name);
```

Samples for Hierarchy Custom Settings

The following example uses the `getOrgDefaults` method to return the data set values for the organization level:

```
CustomSettingName__c mc = CustomSettingName__c.getOrgDefaults();
```

The following example uses the `getInstance` method to return the data set values for the specified profile. The `getInstance` method can also be used with a user ID.

```
CustomSettingName__c mc = CustomSettingName__c.getInstance(Profile_ID);
```

SOAP API

Custom settings that have Privacy defined as Public have the same type of exposure to the API as custom objects. When a custom setting is contained in a managed package, and Privacy for a custom setting is Protected, the settings can only be accessed by the Apex code or formulas that are part of the managed package.



Note You can also access custom settings data through a Standard Object Query Language (SOQL) query, but this method doesn't use the application cache. It's similar to querying a custom object.

See Also

[Grant Permissions on Custom Settings](#)

Custom Settings Limits and Considerations

When working with custom settings, be aware of the following considerations and limits on the amount of cached data.

- Custom settings are a type of custom object. Each custom setting counts against the total number of custom objects available for your organization.
- The total amount of cached data allowed for your org is the lesser of these two values.
 - 20 MB
 - 2 MB multiplied by the number of full-featured user licenses in your org

For example, if your org has three full licenses, you have 6 MB of custom setting storage. If your org has 20 full licenses, you have 20 MB of storage.

Each certified managed package gets a separate limit in addition to your org limit. Each certified managed package gets the same limit as the org, based on the number of licenses up to 20 MB. For example, let's say your org has two certified managed packages installed and you have three full licenses. Each certified managed package can have 6 MB of custom setting storage in addition to your org's 6-MB custom setting storage limit.

The org custom setting limits and certified managed package custom setting limits aren't shared.

- Custom settings that are added to an org via a certified managed package count against that

package's storage limit.

- Custom settings that are added to an org directly or from a non-certified managed package count against the org's storage limit.

Using the previous example, if you have three full licenses, one non-certified managed package, and two certified managed packages, your custom settings storage limits are: 6 MB combined for the org and non-certified managed package, and 6 MB for each certified managed package.

- You can add up to 300 fields per custom setting, unless your field limit for custom objects is lower than 300. If your custom objects field limit is lower than 300, your field limit for custom settings is equal to your custom objects field limit.
- You can't share a custom setting object or record.
- No owner is assigned when a custom setting is created, so the owner can't be changed.
- Accessing an undeleted custom setting in a formula field results in an error if the user doesn't have the "Customize Application" permission. To prevent this error, redeploy this custom setting to the organization. Alternatively, delete this custom setting, re-create it with the same name and data, and then delete and re-create all formula fields that use this setting.
- If a cross-object formula references a currency field from a custom setting, this field value isn't converted to the currency of the record containing the formula. An inaccurate formula result is possible if the custom setting field's currency and the record's currency are different.
- You can't disable specific permissions with a muting permission set.

To see how much custom settings data your organization is using, from Setup, enter *Custom Settings* in the **Quick Find** box, then select **Custom Settings**. For each custom setting, this page lists the size of one record, the number of records created, and the total size used for each custom setting.

Record size is based on the maximum field size of each field type, not the actual storage that's used in each field. When adding fields to a custom setting definition, use the appropriate type and specify a length that doesn't exceed what's needed for your data. This action helps you avoid reaching the cached data limit. For example, if you create a US social security number (SSN) field, select the `Text` data type and specify a length of 9. If you select a `Text Area` data type, the field would add 255 characters to the usage count for each record, regardless of the number of characters entered.



Note A icon indicates that the custom setting is in an installed managed package. You can't edit or delete a protected custom setting installed from a managed package.

Customize Fields

Customize standard and custom fields to tailor your org to your own unique requirements.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

Standard Fields and Page Layouts are not available in **Database.com**

USER PERMISSIONS NEEDED

To create or change custom fields: [Customize Application](#)

You can:

- Modify some aspects of standard fields
- Change or add values to standard and custom picklist fields
- Define dependency rules between fields
- Create custom fields to capture additional information
- Create formula fields that automatically calculate values based on the contents of other fields
- Define default values for custom fields
- Define validation rules for your fields
- Make a field required
- Set fields to track changes, including the date, time, nature of the change, and who made the change
- Create page layouts to control the display of fields
- Set field-level security to control access to fields
- Create or modify field sets

Customize Standard Fields

You can customize several aspects of standard fields, such as the values in picklists, the format for auto-number fields, tracking field history, lookup filters on relationship fields, and field-level help.

Modify Standard Auto-Number Fields in Salesforce Classic

The unique identifiers for solution, case, and contract records are standard auto-number fields. Each record is assigned a unique number with a specified format upon creation. You can modify the format and numbering for these auto-number fields in Salesforce Classic.

Define Default Field Values

Define a default value for a field. Use a formula to generate dynamic values or constants for static values.

Validation Rules

Improve the quality of your data using validation rules. Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record.

Examples of Validation Rules

Review examples of validation rules for various types of apps that you can use and modify for your own purposes. *Validation rules* verify that the data a user enters in a record meets the standards you specify before the user can save the record.

Require Field Input to Ensure Data Quality

Improve the quality of data that users enter in Salesforce by creating universally required fields.

About Field Sets

A field set is a grouping of fields. For example, you could have a field set that contains fields describing a user's first name, middle name, last name, and business title.

Roll-Up Summary Field

A roll-up summary field calculates values from related records, such as those in a related list. You can create a roll-up summary field to display a value in a master record based on the values of fields in a detail record. The detail record must be related to the master through a master-detail relationship. For example, you want to display the sum of invoice amounts for all related invoice custom object records in an account's Invoices related list. You can display this total in a custom account field called Total Invoice Amount.

Lookup Filters

Improve user productivity and data quality with lookup filters. Lookup filters are administrator settings that restrict the valid values and lookup dialog results for lookup, master-detail, and hierarchical relationship fields.

Fields: What's Different or Not Available in the Salesforce Mobile App

Not every Lightning Experience feature is in the Salesforce mobile app. Find out what's different.

Customize Standard Fields

You can customize several aspects of standard fields, such as the values in picklists, the format for auto-number fields, tracking field history, lookup filters on relationship fields, and field-level help.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: All Editions except for **Database.com**.

USER PERMISSIONS NEEDED

To change standard fields: Customize Application

 **Tip** You can't delete standard fields, but you can remove them from your page layouts.

1. Navigate to the fields page for your object.
2. Click the field label.
3. To add custom help text, click **Edit**.
4. On the field's information page, you can—depending on your edition—set field-level security, view accessibility settings, and configure validation rules.

You can also do more, depending on the field's type. For example, if the field is a picklist, you can add, delete, and reorder its values, and set dependencies. You can't increase the field length of a standard field. If you need a longer text area, consider creating a custom field.

 **Note** What does that **Indexed** checkbox mean on a field, and how did it get there? If a field is indexed, you can use sidebar search or advanced search to find values in the field. Having a field indexed can also speed up other operations on the field, such as reporting. See this blog post to find out more: [Know Thy Salesforce Field Indexes for Fast Reports, List Views, and SOQL](#).

Capturing Gender-Related Data with Standard Fields

Represent the diversity of your customer base by recording gender-related data in Salesforce. Use standard, optional fields with vetted value sets to help you personalize your data, build trust, and close data gaps in your system of record.

See Also

- [Create a Custom Field](#)
- [Add or Edit Picklist Values](#)
- [Rename Object, Tab, and Field Labels](#)
- [Field-Level Help](#)
- [Lookup Filters](#)

Capturing Gender-Related Data with Standard Fields

Represent the diversity of your customer base by recording gender-related data in Salesforce. Use standard, optional fields with vetted value sets to help you personalize your data, build trust, and close data gaps in your system of record.

Considerations and Guidelines for Capturing Gender-Related Data

Capturing demographic information boosts data quality, builds trust, and lets you personalize offerings to retain and attract new customers. But release of gender information could result in safety concerns for individuals or liability issues for your company. It's important to classify your data correctly so that you can protect it.

Capture Gender-Related Data

Set up the Gender Identity and Pronouns fields on the Lead, Contact, and Person Account objects to record gender-related data. And, add nonbinary honorifics like Mx. to the Salutation picklist value list to keep the data in sync. You can customize aspects of these fields to suit the unique needs of your company.

Considerations and Guidelines for Capturing Gender-Related Data

Capturing demographic information boosts data quality, builds trust, and lets you personalize offerings to retain and attract new customers. But release of gender information could result in safety concerns for individuals or liability issues for your company. It's important to classify your data correctly so that you can protect it.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

USER PERMISSIONS NEEDED

To customize standard fields:

Customize Application

Pronouns and Gender Identity are standard fields on the Lead, Contact, and Person Account objects. Non-binary picklist values on the Salutation field help keep all your data in sync. Before setting up these fields, ask yourself if it's necessary for your use cases. Discuss your goals with project stakeholders, and take measures to ensure the data can't inadvertently be used to discriminate. Always review the privacy and legal ramifications with your legal department, and refer to your industry's standards for guidance.

When you gather pronouns or gender identity data from your customers, follow these guidelines.

- Explain why you're asking for this specific information.
- Describe how the data is used.
- Offer how it benefits them to share their personal information with you.
- Indicate how their data is protected.
- Give customers an option to opt out.
- Set up data classification levels.
- Manage access to the fields and to the data within them.
- To prevent mismatched data, add custom business processes.

To report on gender-related data, make sure that the top-level object in the report is Contact, Lead, or Person Account. For example, a report with Campaign as the top-level object doesn't have gender-related fields available.

See Also

[Classify Sensitive Data to Support Data Management Policies](#)

[Store Customers' Data Privacy Preferences](#)

[Manage Data Access](#)

[Ways to Control User Access to Fields](#)

[Automate Your Business Processes](#)

Capture Gender-Related Data

Set up the Gender Identity and Pronouns fields on the Lead, Contact, and Person Account objects to record gender-related data. And, add nonbinary honorifics like Mx. to the Salutation picklist value list to keep the data in sync. You can customize aspects of these fields to suit the unique needs of your company.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

USER PERMISSIONS NEEDED

To customize standard fields:

Customize Application

 **Note** Review your data classification settings and assess legal and privacy considerations before

collecting and storing gender-related data.

1. From the object management settings for the object, go to Fields and Relationships. Next, click the name of the field you want to set up.
2. Customize the Data Sensitivity Level field, Compliance Categorization field, and Help Text field.
3. Add or edit picklist values.
4. Add the fields to page layouts.
5. Set the field level security.

See Also

- [Considerations and Guidelines for Capturing Gender-Related Data](#)
[Customize Standard Fields](#)
[Add or Edit Picklist Values](#)
[Page Layouts](#)
[Field-Level Security](#)

Modify Standard Auto-Number Fields in Salesforce Classic

The unique identifiers for solution, case, and contract records are standard auto-number fields. Each record is assigned a unique number with a specified format upon creation. You can modify the format and numbering for these auto-number fields in Salesforce Classic.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To modify standard auto-number fields: [Customize Application](#)

1. From Setup, enter the name of the object whose field you want to modify in the Quick Find box, then select **Fields**.
2. Click **Edit** next to the name of the auto-number field.
For example, for cases, edit the Case Number field.
3. Enter a Display Format to control such formatting details as the minimum number of leading zeros as well as any prefix or suffix for the number.
For more information, see [Custom Field Attributes](#).

Format changes don't affect existing records; they're applied only to new records.

4. Enter the number to be assigned to the next record that is created after you save your changes.
5. Click **Save**.



Warning Salesforce warns you if the next number you enter isn't higher than existing numbers. However, it's possible to create duplicate numbers if you change the auto-number format multiple times using similar formats each time.

See Also

[Custom Field Types](#)

Create a Custom Field

Capture your unique business data by storing it in custom fields. When you create a custom field, you configure where you want it to appear and optionally control security at the field level.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Essentials, Starter, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Salesforce Connect external objects are available in: **Developer** Edition and for an extra cost in: **Enterprise, Performance, and Unlimited** Editions

Custom fields aren't available on Activities in **Group** Edition

Layouts aren't available in **Database.com**

USER PERMISSIONS NEEDED

To create or change custom fields: Customize Application

To add field-level security to profiles or permission sets: Manage Profiles and Permission Sets

Important Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Before you begin, determine the [type of field](#) you want to create and review the considerations. See [Considerations for Creating a Custom Field](#).

1. From the management settings for the object you want to add a field to, go to Fields & Relationships. Custom task and event fields are accessible from the object management settings for Activities.
2. Click **New**.
3. Choose the type of field and then click **Next**.
4. For relationship fields, associate an object with the field and click **Next**.
5. For indirect lookup relationship fields, select a unique, external ID field on the parent object, and then click **Next**. The parent field values are matched against the values of the child indirect lookup

relationship field to determine which records are related to each other.

6. Enter a field label.

Salesforce populates Field Name using the field label. Use the field name for merge fields in custom links, custom s-controls, and when referencing the field from the API.

7. To base a picklist field on a global picklist value set, select the value set to use.
8. To specify whether the field must be populated and what happens if the record is deleted, enter field attributes and select the appropriate checkboxes.
9. For master-detail relationships on custom objects, optionally select **Allow reparenting** to allow a child record in the master-detail relationship to be reparented to a different parent record.
10. For a relationship field, optionally limit search results for the field by creating a lookup filter. Lookup filters aren't available for external objects.
11. Click **Next**.
12. In Enterprise, Unlimited, Performance, and Developer Editions, specify the field's access settings for each profile or permission set, and then click **Next**.

To specify the field's access settings for permission sets instead of profiles, enable **Field-Level Security for Permission Sets during Field Creation** on the User Management Settings page. If you specify access for permission sets, select **Permission sets with object permissions** to filter the list to permission sets that have Create, Read, Edit, or Delete access on the field's object. To show all permission sets, deselect this option. If no permission sets have object permissions for the field's object, the list contains all permission sets.

Access Level	Enabled Settings (Profiles)	Enabled Settings (Permission Sets)
Users can read and edit the field.	Visible	Edit Access (Read Access is selected automatically)
Users can read but not edit the field.	Visible and Read-Only	Read Access
Users can't read or edit the field.	None	None

By default, a custom field isn't visible or editable for portal profiles unless the field is universally required.

13. Select the Dynamic Forms-enabled Lightning record pages that should include the field, then click **Next**.
If you don't have any Dynamic Forms-enabled Lightning record pages for the object, this step doesn't appear.
14. Select the page layouts that should include the field, and then click **Next**.
15. For relationship fields, optionally click **Related List Label**, enter a new name to create an associated records related list, and then add it to the page layouts for that object. To add the related list to customized page layouts, select **Append related list to users' existing personal customizations**.

16. Click **Save** to finish or **Save & New** to create more custom fields.

The number of custom fields allowed per object varies according to your Salesforce edition. For the total number of custom fields that you can create, see [Custom Fields Allowed Per Object](#).

Creating fields can require changing a large number of records at once. If your request is queued to process these changes efficiently, you receive an email notification when the process has been completed.

Want to customize Salesforce so it captures all your business data? This short video walks you through how to create a custom picklist field, from choosing the correct field type to applying field-level security.

Watch a Demo:  [How to Create a Custom Field in Salesforce \(Salesforce Classic\)](#)

Want to add and arrange a new field while viewing an individual record for an object? This short video walks you through creating a picklist field while viewing a contact and then changing the page layout for the field.

Watch a Demo:  [How to Add a Custom Field in Salesforce \(Lightning Experience\)](#)

See Also

[Trailhead: Custom Fields: Quick Look](#)

[Salesforce Help: External Object Relationships](#)

[Salesforce Help: Which Custom Fields Can I Encrypt?](#)

[Salesforce Help: Differences Between Classic Encryption and Shield Platform Encryption](#)

[Salesforce Help: Find Object Management Settings](#)

[Salesforce Help: Map Custom Lead Fields for Lead Conversion](#)

[Salesforce Help: Enable Field-Level Security for Permission Sets during Field Creation](#)

[Salesforce Help: Set Field-Level Security for a Field on All Permission Sets](#)

Considerations for Creating a Custom Field

Keep these considerations in mind when you create a custom field.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Essentials, Starter, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Salesforce Connect external objects are available in: **Developer** Edition and for an extra cost in: **Enterprise, Performance, and Unlimited** Editions

Custom fields aren't available on Activities in **Group** Edition

Layouts aren't available in **Database.com**

 **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Data Types

- Some data types are available for certain configurations only. For example, the Master-Detail Relationship option is available for custom objects only when the custom object doesn't already have a master-detail relationship.
- Custom settings and external objects allow only a subset of the available data types.
- Relationship fields count toward custom field limits.
- Field types correspond to API data types.
- If your org uses Shield Platform Encryption, ensure that you understand how to encrypt custom fields using the Shield Platform Encryption offering.
- Only dates within a certain range are valid. The earliest valid date is 1700-01-01T00:00:00Z GMT, or just after midnight on January 1, 1700. The latest valid date is 4000-12-31T00:00:00Z GMT, or just after midnight on December 31, 4000. These values are offset by your time zone. For example, in the Pacific time zone, the earliest valid date is 1699-12-31T16:00:00, or 4:00 PM on December 31, 1699.

Availability

- You can't add a multi-select picklist, rich text area, or dependent picklist custom field to opportunity splits.
- Additional field types can appear if an AppExchange package using those field types is installed.
- The [Roll-Up Summary](#) option is available only on certain objects.
- By default, a custom field isn't visible or editable for portal profiles unless the field is [universally required](#).
- When you're close to the limit of 800 custom fields and you delete or create fields, field creation can fail. The physical delete process reclaims and cleans fields, making them count temporarily toward the limit. The delete process runs only when the queue is full, so it can take days or weeks to start. In the meantime, the deleted fields are still counted as part of the limit. To request immediate deletion of fields, contact Salesforce Support.
- For lead custom fields, you can click **Map Lead Fields** to specify which custom lead fields to map to custom account, contact, and opportunity fields during a lead conversion.
- If your org uses Translation Workbench, notify your translators that new fields need translation.

Merge Fields

- Ensure that the custom field name and label are unique for the object.
 - If standard and custom fields have identical names or labels, the merge field displays the custom field value.
 - If two custom fields have identical names or labels, the merge field can display an unexpected value.

If you create a custom field label called *Email* and a standard field labeled *Email* exists, the merge

field is unable to distinguish between the fields. Add a character to the custom field name to make it unique. For example, `Email12`.

Layouts

- Newly created custom fields are added as the last field in the first two-column section of the page layout, with these exceptions.

Field	Location on Page Layout
Long text area	End of the first one-column section.
User	Bottom of the user detail page.
Universally required	Can't remove it from page layouts or make it read only.

- When new custom fields are added to Dynamic Forms-enabled Lightning pages, here's what happens.
 - Fields are added as the last field in the first column on the first field section on the page in top-down, left-to-right order. If the field section is in a container, such as a Tabs or Accordion component, then the first field section is chosen from the first tab or the first accordion section. The "active" tab doesn't matter. If the first field section on a page is the Dynamic Highlights Panel, then new fields are added to the second field section instead.
 - After the field creation wizard is complete, it can take up to 5 seconds after you save the field for the page you selected to be updated with the new field. This process can take longer if your system is under heavy loads.
 - Standard Lightning pages and pages installed from managed packages don't appear in the list of available pages for your custom field.

Notifications

- Creating fields can require changing a large number of records at once. If your request is queued to process these changes efficiently, you receive an email notification when the process is completed.
- Sometimes an error notification email is sent even if a field is successfully created.

Custom Fields Allowed Per Object

The number of custom fields allowed per object varies according to your Salesforce edition.

Personal Edition	Contact Manager	Group Edition	Essential s Edition	Starter Edition	Professio nal Edition	Enterpris e Edition	Unlimite d and Performa nce Edition	Develope r Edition
5	25	100	100	25	100	500	800	500

To enable installation of AppExchange apps on all Salesforce editions, you can install additional fields beyond the edition limits if those fields come from an AppExchange certified managed package. You can install additional fields up to the maximum hard limit allowed per object.

- The objects in the following list have a maximum hard limit of 900 custom fields.
- All other objects have a maximum hard limit of 800 custom fields.

Let's look at a few examples using the objects in the list. For Enterprise Edition, you can create 500 custom fields on an object and install an additional 400 fields from an AppExchange certified managed package. For Unlimited Edition, you can create 800 custom fields on an object and install an additional 100 fields from an AppExchange certified managed package.

These objects have a maximum hard limit of 900 custom fields.

- Account
- AccountContactRelation
- Asset
- Campaign
- CampaignMember
- Case
- Contact
- ContentVersion
- Contract
- Custom Object
- Individual
- KnowledgeArticleVersion
- Lead
- Opportunity
- OpportunityLineItem
- Order
- OrderItems (Order Product)
- Product2 (Products)
- Solution
- Users
- UserRole (Role)

This object has a maximum hard limit of 50 custom fields.

- PriceBookEntry

If you have an AppExchange certified managed package installed in your org and that package expires, the custom fields in that package count against the edition custom field limit regardless of whether the **Don't Count towards Apps, Tabs, and Objects Org Limits** field is selected. For more information about AppExchange packages and limits, see the edition and org limits in [Before You Install an AppExchange Package](#).

You can create up to 300 custom fields for activities, as long as your Salesforce org has fewer than 700

million tasks and events. If your org has more than 100 custom fields and now has more than 700 million tasks and events, you can't create any more custom fields. However, your preexisting fields are retained to prevent data loss. To regain the ability to create up to 300 custom fields, use [Salesforce Archive](#), [Data Cloud](#), or third-party data archiving solutions to archive your activities, and then [delete archived activities](#) to reduce the number of activities in your org.

Custom Field Type Allocations

The maximum number of activities, long text area fields, rich text area fields, relationship fields, and roll-up summary fields varies according to your Salesforce edition.

Field Type	Essentials Edition	Starter Edition	Personal Edition	Contact Manager	Group Edition	Professional Edition	Enterprise Edition	Developer Edition	Unlimited and Performance Edition
Activities	No additional allocation					20	100		
Long text area	An object can contain unlimited rich text area and long text area fields, although your edition's allocation for total custom fields allowed on an object, regardless of field type, applies. Each object can contain 1,638,400 characters across long text area and rich text area fields. When you create a long text area or rich text area field, you set a character limit for the field—the maximum length of the text that can be entered. The default character limit for long text area and rich text area fields is 32,768 (32 KB). The maximum character limit for long text area and rich text area fields is 131,072 (128 KB). The minimum character limit is 256. The maximum size of an image that can be uploaded in a rich text area field is 1 MB.								
Relationship			40						
Roll-up summary	No additional allocation		25						

-  **Note** For custom compound fields, each component counts as one custom field toward your org's limits. Each geolocation field counts as three custom fields: one for latitude, one for longitude, and one for internal use. Similarly, each custom address field counts as nine custom fields: one each for street, city, postal code, country code, state code, geocode accuracy level, longitude, and latitude, plus one for internal use.

See Also

[Salesforce Features and Edition Allocations](#)

[Knowledge Article: Increase the maximum relationships \(master-detail plus lookup\) allowed per](#)

object

Create a Custom Picklist Field

Create custom picklist fields to let your users select values from lists that you define.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: all editions

USER PERMISSIONS NEEDED

To create or change custom fields: Customize Application

Watch a Demo:  [Custom Fields: Picklists](#)

You can create these types of picklist fields:

- Local picklist—Lets users select a single value from a list that you define. This picklist is unique and has its own set of values.
- Shared picklist—Lets users select a single value from a global picklist value set that you define in Setup. All custom picklist fields that use a global value set inherit its values and can't have additional values.
- Multi-select picklist—Allows users to select more than one picklist value from a list that you define. These fields display each value separated by a semicolon.

You can't add a multi-select picklist, rich text area, or dependent picklist custom field to opportunity splits.

1. Go to the fields area of the object you want to create a picklist field for.
2. In the custom fields related list, click **New**.
3. Select **Picklist or Picklist (Multi-Select)**, and then click **Next**.
4. Enter a label for the picklist field.
5. To use the values from an existing global picklist, select **Use global picklist value set**. To use values that you create specifically for this picklist, select **Enter values for the picklist, with each value separated by a new line**.
6. If you didn't use a global picklist value set, enter picklist values.
Put each value on a separate line. Values can be up to 255 characters long.
7. Optionally, sort the values alphabetically or use the first value in the list as the default value, or both.
If you select both options, Salesforce alphabetizes the entries and then sets the first alphabetized value as the default.



Note Don't assign default values to fields that are both required and unique, because uniqueness errors can result. See [Default Field Values](#).

You can use a formula to assign a default value dynamically. For example, you can assign a value based

on the current user. The following formula sets an Opportunity priority to High for all users in the Fast Response Sales profile. Other users see the default listed in the Values set.

```
IF($Profile.Name = "Fast Response Sales", "High", "")
```

For picklists, a valid formula result is either a constant or the API name of an entry in the Values list. The formula result has higher precedence than the default assigned in the Values list. If the formula doesn't generate a valid result, the default assigned in the Values list is entered in the field. If a default isn't assigned to the Values list, no value is entered in the picklist field.

8. Choose whether to restrict this picklist's values to an admin-approved list. Selecting **Restrict picklist to the values defined in the value set** prevents users from loading unapproved values through the API. When you set a picklist to be unrestricted, users can't enter new values through the user interface, but they can add new values via the API, automation, or other apps.
9. If you're creating a multi-select picklist, enter how many values you want displayed at a time on edit pages. The number of values determines the box height.
10. Enter a description or help text if desired, and then click **Next**.
11. Set field-level security for the picklist field, and then click **Next**.
12. Select the Dynamic Forms-enabled Lightning record pages that should include the field, then click **Next**.
If you don't have any Dynamic Forms-enabled Lightning record pages for the object, this step doesn't appear.
13. Choose the page layouts on which to include the picklist field.
14. Save your changes.

See Also

- [Create a Custom Field](#)
- [Add or Edit Picklist Values](#)
- [Create a Global Picklist Value Set](#)
- [Make Your Custom Picklist Field Values Global](#)

Create a Global Picklist Value Set

Use a global picklist value set to share values across objects and custom picklist fields, and to restrict the picklists to only the values that you specify.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: all editions

USER PERMISSIONS NEEDED

To create or change custom fields: Customize Application

A custom picklist is tied to a particular object as a field on the object. Unlike a custom picklist field, a

global picklist exists independently as a *global picklist value set*. Its values are shared with any picklist that's based on it. A global picklist is a restricted picklist by nature. Only a Salesforce admin can add to or modify its values. Users can't add unapproved values, even through the API.

1. If you need to, first review the considerations information at the end of this topic. From Setup, enter **Picklist** in the **Quick Find** box, then select **Picklist Value Sets**.
2. Next to Global Value Sets, click **New**.
3. Enter a label for the global value set. This name appears in Setup, and when users create a picklist based on this global value set.
4. To tell users what these values are for, enter a specific description of the global value set. This text appears on the Picklist Value Sets list page in Setup.
5. Enter the values, one per line.
6. Optionally choose to sort the values alphabetically or to use the first value in the list as the default value, or both.
If you select both options, Salesforce alphabetizes the entries and then sets the first alphabetized value as the default.
7. Click **Save**.

Your global value set is ready to be used in custom picklist fields. To arrange values or re-alphabetize them, use **Reorder**. You can't undo a custom picklist field's association with a global value set. If you need a picklist field to use a different global value set or different individual values, delete the custom picklist field, and create a new one in its place.

To create a picklist that uses a global picklist value set, see [Create a Custom Picklist Field](#). To see all the fields where this value set is used, look under Fields Where Used on the global picklist's detail page.

As you add new values to an existing global picklist, you can add the new values to all record types that use the picklist. Select **Add the new picklist values to all Record Types that use this Global Value Set**; otherwise, you have to add the new values to existing records types manually.

There are limits for global picklist value sets.

- Global picklist value sets have a combined active and inactive limit of 1,000.
- You can have up to 500 picklist global value sets in an org.
- There's no limit on the number of custom picklists that use global picklist value sets.
- If you apply a global picklist value set to more than 13 different objects, you can deactivate values from the picklist value set, but you can't replace any picklist values or delete values from the set.

See Also

- [Manage Inactive Picklist Values](#)
- [Create a Custom Picklist Field](#)
- [Make Your Custom Picklist Field Values Global](#)

Make Your Custom Picklist Field Values Global

When you create a custom picklist, it's only available to the current object. To share the values with other

objects, promote the picklist to a global value set. The original custom picklist references the global value set for its values, and the global value set is also available to other custom picklists.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: all editions

USER PERMISSIONS NEEDED

To create or change custom fields: Customize Application

After a picklist is promoted to a global value set, you can't demote it. You manage the values in the global value set or edit the custom picklist field to use different values.

1. If you need to, first review the considerations information at the end of this topic. Go to the fields area of the object you want to create a picklist field for.
2. In the Custom Fields related list, click **Edit**.
3. Click **Promote to Global Value Set**.



4. Enter a label for the global value set.
5. Accept the Field Name or edit it.
6. Optionally, enter a description to identify it when using the values for other custom picklists.
7. Click **Promote to Global Value Set**, again.

You can use the new global value set for other fields and manage the values in Picklist Value Sets.

When promoting a custom picklist, be aware of these considerations.

- You can promote fields that have up to 1,000 values (both active and inactive).
- You can promote only restricted picklists. To promote an unrestricted picklist, convert it to a restricted picklist.
- You can't promote a field to an existing global value set.
- Picklist values translated using Translation workbench have a limit of 40 characters.
- You can't undo a custom picklist field's association with a global value set. If you need a picklist field to use a different global value set or different individual values, delete the custom picklist field, and create a new one in its place.

See Also

[Create a Custom Picklist Field](#)

[Create a Global Picklist Value Set](#)

Custom Field Types

Salesforce supports various field types, allowing you to select the right one for your data. You can also convert an existing field to a different data type if needed.. When you have data that doesn't match any

of the standard fields, your admin can create a custom field for that data. For example, create a Middle Name field for contacts to capture additional information.

! **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

The first step in creating a custom field is to choose the type of the field. This table includes a description of each custom field type.

Type	Description
Address	Lets users enter a street, city, state or province, zip or postal code, and country, or to search for an address with an external tool. When a user selects an address using the tool, the street, city, state or province, zip or postal code, and country are populated.
Auto Number	Automatically assigns a unique number to each record. The maximum length of any auto-number field is 30 characters, 20 of which are reserved for prefix or suffix text. Not available for external objects.
Checkbox	Lets users check a box, indicating a true or false attribute of a record. When using a checkbox field for a report or list view filter, use “True” for checked values and “False” for unchecked values. The Data Import Wizard and the weekly export tool use “1” for checked values and “0” for unchecked values.
Currency	<p>Lets users enter a currency amount. The system automatically formats the field as a currency amount. This formatting is useful if you export data to a spreadsheet application. Not available for external objects.</p> <p>The way Salesforce rounds a currency custom field value depends on the number of decimal places (number of digits to the right of the decimal point) defined for the field.</p> <ul style="list-style-type: none"> • If the number of decimal places is zero (no decimal values) so that the field can contain only whole numbers with no digits to the right of the decimal point, Salesforce uses truncation rounding. All numbers to the right of the decimal point are removed. <ul style="list-style-type: none"> - 23.5 becomes 23 - -23.5 becomes -23 - -23.999999 becomes -23 • If the number of decimal places is 1 or greater (1 or more decimal values), Salesforce uses half-away-from-zero rounding. <ul style="list-style-type: none"> - If the custom field is defined with 1 decimal place: <ul style="list-style-type: none"> - -13.55 becomes -13.6 - 23.44 becomes 23.4

Type	Description
	<ul style="list-style-type: none"> - 36.77 becomes 36.8 - If the custom field is defined with 4 decimal places: <ul style="list-style-type: none"> - -14.5555 becomes -14.5556 - 27.59994 becomes 27.5999 <p>Values lose precision after 15 decimal places.</p>
Date	Lets users enter a date or pick a date from a popup calendar. In reports, you can limit the data by specific dates using any custom date field.
Date/Time	Lets users enter a date or pick a date from a popup calendar and enter a time of day. There are visual and behavioral differences for Date/Time fields in Lightning Experience and Salesforce Classic. In Lightning Experience, the date and time fields are separate, and the initial time is set to 12:00 PM when you select a date in the calendar. In Salesforce Classic, the date/time field is a single field. You can set the field to the current date and time by clicking the date and time link next to the field. The time of day includes AM or PM notation. In reports, you can limit the data by specific dates and times using any custom date field.
Email	<p>Lets users enter an email address of up to 80 characters, which is validated to ensure proper format. If this field is specified for contacts or leads, users can choose the address when clicking Send an Email.</p> <p>You can't use custom email addresses for mass emails or list emails.</p> <p>Emails sent to a record's custom email address fields aren't logged against that record.</p> <p>This field can be encrypted using Shield Platform Encryption.</p>
External Lookup Relationship	When you create an external lookup relationship field, the standard External ID field on the parent external object is matched against the values of the child's external lookup relationship field. External object field values come from an external data source.
Formula	<p>Lets users automatically calculate values based on other values or fields such as merge fields. Not available for external objects.</p> <p>Salesforce uses the round half up tie-breaking rule for numbers in formula fields. For example, 12.345 becomes 12.35 and -12.345 becomes -12.35.</p>
Geolocation	Lets users specify a location by its latitude and longitude. Geolocation is a compound field that counts toward your org's limits as three custom fields: one for latitude, one for longitude, and one for internal use. Not available for external objects.

Type	Description
Hierarchical Relationship	Creates a hierarchical lookup relationship between users. Lets users use a lookup field to associate one user with another that doesn't directly or indirectly refers to itself. For example, you can create a custom hierarchical relationship field to store each user's direct manager.
Indirect Lookup Relationship	An indirect lookup relationship links a child external object to a parent standard or custom object. When you create an indirect lookup relationship field on an external object, you specify the parent object field and the child object field to match and associate records in the relationship. Specifically, you select a custom unique, external ID field on the parent object to match against the child's indirect lookup relationship field. The child lookup field's value comes from an external data source.
Lookup Relationship	<p>Creates a relationship between two records so you can associate them with each other. For example, opportunities have a lookup relationship with cases that lets you associate a particular case with an opportunity.</p> <ul style="list-style-type: none"> On a standard or custom object, a lookup relationship creates a field that allows users to click a lookup icon and select another record from a window. On an external object, the lookup relationship field references 18-character Salesforce IDs that are stored in an external data source. Those IDs are matched against the parent object to determine which records are related to each other. <p>On the parent record, you can display a related list to show all the records that are linked to it. You can create lookup relationship fields that link to users, standard objects, or custom objects. If a lookup field references a record that has been deleted, Salesforce clears the value of the lookup field by default. Alternatively, you can choose to prevent records from being deleted if they're in a lookup relationship.</p> <p>Lookup relationship fields aren't available in Personal Edition.</p> <p>Lookup relationship fields to campaign members aren't available; however, lookup relationship fields from campaign members to standard or custom objects are available.</p>
Master-Detail Relationship	<p>Creates a relationship between records where the main record controls certain behaviors of the detail record such as record deletion and security.</p> <p>Not available for standard objects or external objects, although you can create a master-detail relationship field on a custom object that links to a standard object.</p> <p>Master-detail relationships can't be used with campaign members.</p>

Type	Description
Number	<p>Lets users enter any number. This number is treated as a real number and any leading zeros are removed.</p> <p>Salesforce uses half-away-from-zero rounding for number fields. Examples:</p> <ul style="list-style-type: none"> • 12.345 becomes 12.35 • -12.345 becomes -12.35 • -12.343 becomes -12.34 • -12.347 becomes -12.35 • 13.5 becomes 14 • -13.5 becomes -14 <p>Salesforce rounds numbers referenced in merge fields according to the user's locale, not the number of decimal spaces specified in the number field configuration.</p> <p>When you use Einstein Prediction Builder to build a prediction, a custom field is created for storing and displaying the prediction scores on records. Or, you can manually create a predictive custom field, enable it as an AI prediction field, and enter its name and label when building a prediction in Einstein Prediction Builder. For example, to display how likely a customer is to pay an invoice on time.</p> <p>In Lightning Experience, custom objects can store more decimal places than you define. If you enter 90.678 on a field that accepts 2 decimal places, the number is displayed as 90.68 on the record form. When you inline edit, the field shows the original input, 90.678. Similarly, the value is stored as 90.678 in the database. In Salesforce Classic, the input 90.678 is saved as 90.68.</p>
Percent	<p>Lets users enter a percentage number as a decimal—for example, 0.10. The system automatically converts the decimal to a percentage—for example, 10%.</p> <p>Values lose precision after 15 decimal places. Also, if you enter a value with more than 15 decimal places and add a percent sign to the number, a runtime error occurs.</p>
Phone	<p>Lets users enter any phone number. Character limit is 40.</p> <p>Salesforce automatically formats it as a phone number.</p> <p>If you use Salesforce CRM Call Center, custom phone fields are displayed with the  button, allowing click-to-dial functionality. Therefore, Salesforce recommends that you don't use a custom phone field for fax numbers.</p> <p>This field can be encrypted using Shield Platform Encryption.</p>

Type	Description
Picklist	Lets users select a single value from a list that you define. Available for external objects only with the cross-org adapter for Salesforce Connect.
Picklist (Multi-select)	Lets users select more than one picklist value from a list that you define. These fields display each value separated by a semicolon. Available for external objects only with the cross-org adapter for Salesforce Connect.
Roll-Up Summary	Automatically displays the record count of related records or calculates the sum, minimum, or maximum value of related records. The records must be directly related to the selected record and on the detail side of a custom master-detail relationship with the object that contains the roll-up summary field. For example, a custom field called “Total Number of Guests” displays the number of guest custom object records in the Guests related list. Not available for external objects.
Text	<p>Lets users enter any combination of letters, numbers, or symbols. You can set a maximum length, up to 255 characters.</p> <p>This field can be encrypted using Shield Platform Encryption.</p>
Text (Encrypted)	<p>Lets users enter any combination of letters, numbers, or symbols that are stored in encrypted form. You can set a maximum length of up to 175 characters. Encrypted fields are encrypted with 128-bit master keys and use the Advanced Encryption Standard (AES) algorithm. You can archive, delete, and import your master encryption key. To enable master encryption key management, contact Salesforce. Not available for external objects.</p> <p>This field can be encrypted using Classic Encryption. If your org uses Shield Platform Encryption, use Text to create an encrypted text field.</p>
Text Area	Lets users enter up to 255 characters that display on separate lines similar to a Description field.
Text Area (Long)	<p>Lets users enter up to 131,072 characters that display on separate lines similar to a Description field. You can set the length of this field type to a lower limit, if desired. Any length from 256 to 131,072 characters is allowed. The default is 32,768 characters. Every time you press Enter within a long text area field, a line break, and a return character are added to the text. These two characters count toward the 131,072 character limit. This data type isn’t available for activities or products on opportunities. The first 999 characters in a standard rich text area or a long text area are displayed in a report. For custom fields, only the first 255 characters are displayed. If you download the report as Details Only, the entire field is available.</p> <p>This field can be encrypted using Shield Platform Encryption.</p>

Type	Description
Text Area (Rich)	With the use of a toolbar, users can format the field content and add images and hyperlinks. The toolbar allows the user to undo, redo, bold, italicize, underline, strike-out, add a hyperlink, upload or link to an image, modify alignment, add a numbered or non-numbered list, indent, and outdent. The maximum field size is 131,072 characters, inclusive of all the formatting and HTML tags. The first 999 characters in a standard rich text area or a long text area are displayed in a report. For custom fields, only the first 255 characters are displayed. If you download the report as Details Only , the entire field is available. The maximum size for uploaded images is 1 MB. Only gif, jpeg, and png file types are supported. Not available for external objects. There are visual and formatting differences for rich text areas in Lightning Experience and the Salesforce mobile app, compared to Salesforce Classic.
Time	Lets users enter a time of day, including hours, minutes, seconds, and milliseconds. The time displays in a 12-hour notation with AM or PM. The displayed time depends on the Locale setting on the Company Information page in Setup.
URL	<p>Lets users enter up to 255 characters of any valid website address. Only the first 50 characters are displayed on the record detail pages. When a user clicks the field in Salesforce Classic, the URL opens in a separate browser window. In Lightning Experience, internal URLs open in the same window and external URLs open in a separate browser window. In Salesforce console apps, the URL opens in a new workspace tab. In Lightning console apps, internal URLs open in a new workspace tab and external URLs open in a separate browser window.</p> <p>When opening an external URL, a message asks for the user's permission. To prevent the window from displaying every time you open an external URL, disable your browser's popup blocker.</p> <p>This field can be encrypted using Shield Platform Encryption.</p>

See Also

[Custom Field Attributes](#)

[Considerations for AI Prediction Fields](#)

[Einstein Prediction Builder Editions and Permissions](#)

Considerations for AI Prediction Fields

When you use Einstein Prediction Builder to build a prediction, a custom field is created for storing and displaying the prediction scores on records. Or, you can create a predictive custom field manually, and enter its name and label when building a prediction in Einstein Prediction Builder.

We recommended that you create a custom field for storing prediction scores through the setup flow in

Einstein Prediction Builder, which creates the field for you. If you choose to manually create the custom field instead, choose the Number type. When you name the field, select the **AI Prediction** checkbox. After the field is enabled, use Einstein Prediction Builder to build a prediction, entering the custom field's exact name and label when setting up the location for storing your prediction scores.

When you let Einstein Prediction Builder create the field for you while building a prediction, it ensures proper scale and precision. If you manually create the field and try to use it when building a prediction, you may get an error message if the custom field is invalid. When creating the custom predictive field manually, be sure to give field-level security permissions to the Admin role. Otherwise, prediction scores won't be able to be stored in this field.

Using an AI Prediction Field in a Custom Formula Field

Custom formula fields can also reference AI prediction fields. For example, create a number field to predict the payment for a service named *LikelyCost__c*. Select the **AI Prediction** checkbox. Then, create a formula field called *InitialOffer__c* with a formula of *LikelyCost__c * 1.5*. In this case, *LikelyCost__c* is a numeric AI Prediction field and *InitialOffer__c* is just a custom formula field. Once the field is used in a prediction, the resulting value is powered by Einstein Prediction Builder.

Considerations for AI Prediction Fields and Custom Formula Fields

Whether Einstein Prediction Builder creates the field for you, or you do it on your own, these fields have some limitations. These limitations apply to both AI prediction fields and custom formula fields that reference AI prediction fields.

- An AI prediction field can be used for only one prediction. Once it's being used for a prediction, whether created automatically or manually, you can't use it for another prediction.
- The field value changes only when new predictions are made about the corresponding object. Value changes in this field do not trigger Process Builder, Apex triggers, or workflows.
- The Roll-Up Summary field type doesn't support AI prediction fields.
- Fields enabled as AI prediction fields included in packages aren't uploaded, so they can't be distributed in packages.
- We don't recommend using a custom number field that Einstein Prediction Builder manages in a validation rule. If the prediction changes the field value in a way that violates the validation rule, you can't save changes to a record that uses the field.

Considerations for AI Prediction Number Fields

These limitations are specific to AI prediction number fields:

- The number of decimal places and the character length of the number field must match the settings in Einstein Prediction Builder. Einstein Prediction Builder handles these settings for you when it creates the field. If you create the field, ensure that the number and length match the prediction in Einstein Prediction Builder. If the field being predicted is a Boolean field type, the number of decimal places defaults to 0 and the character length defaults to 3.

- To create and grant permissions to the field, the Manage Profiles and Permission Sets permission is required. Typically, this permission is set as part of the Customize App permission, but large orgs sometimes keep Manage Profiles and Permission Sets separate.
- You can't disable the field setting. If you have to replace it, delete it and create another field.
- You can't delete a prediction field if Einstein Prediction Builder is still referencing it.
- You can't convert number fields enabled as AI prediction fields to other field types.
- The custom field setting, as well as the prediction results (scores) from Einstein Prediction Builder, are available in Salesforce Classic. The Einstein Prediction Builder setup flow is available only in Lightning Experience.

See Also

[Einstein Prediction Builder Editions and Permissions](#)

Geolocation Custom Field

The geolocation custom field allows you to identify locations by their latitude and longitude and to calculate distances between locations.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

You can calculate the distance between two geolocation fields, such as between a warehouse and a store. Or you can calculate the distance between a geolocation field and a pair of latitude and longitude coordinates, such as between a warehouse and 37.794016°, -122.395016°—the location also known as San Francisco. Latitude values must be within `-90` and `90`. Longitude values must be within `-180` and `180`.

Geolocation is a compound field that counts toward your org's limits as three custom fields: one for latitude, one for longitude, and one for internal use. Support for the compound field (geolocation) versus the field's components (latitude and longitude) varies depending on the functionality you're using in Salesforce. For example, you can create list views that show the field and its components, but you can't select the compound geolocation field in Apex. You can run SOQL queries only on a geolocation field's components.

Considerations for Compound and Geolocation Fields

Compound fields, including geolocation fields, have these limitations.

- Compound fields are read-only. To update field values, modify the individual field components.
- Compound fields are accessible only through the SOAP API, REST API, and Apex. The compound versions of fields aren't accessible anywhere in the Salesforce user interface.
- Although compound fields can be queried with the `Location` and `Address` Apex classes, they're editable only as components of the actual field. Read and set geolocation field components by

appending “`_latitude_s`” or “`_longitude_s`” to the field name, instead of the usual “`_c`.” For example:

```
Double theLatitude = myObject__c.aLocation__latitude__s;  
myObject__c.aLocation__longitude__s = theLongitude;
```

You can't access or set the compound value.

- You can't use compound fields in Visualforce—for example, in an `<apex:outputField>`. To access or update field values, use the individual field components.
- If you select compound fields for export in the Data Loader, they cause error messages. To export values, use individual field components.
- Custom geolocation and location fields on standard addresses aren't supported with email templates.
- You can't use compound fields in lookup filters, except to filter distances that are within or not within given ranges. You can use distance lookup filters only in the Metadata API.
- The only formula functions that you can use with compound fields are `ISBLANK`, `ISCHANGED`, and `ISNULL`. You can't use `BLANKVALUE`, `CASE`, `NULLVALUE`, `PRIORVALUE`, or the equality and comparison operators with compound fields. The equality and comparison operators include `=` and `==` (equal), `<>` and `!=` (not equal), `<` (less than), `>` (greater than), `<=` (less than or equal), `>=` (greater than or equal), `&&` (AND), and `||` (OR).

Considerations for Geolocation Fields

This list contains limitations for geolocation fields.

- Geolocation fields aren't supported in custom settings.
- Geolocation fields aren't available in dashboards or Schema Builder.
- Geolocation fields are available in Visual Workflow and in formula-based workflow and approvals, but they can't be used in filter-based workflow updates and approvals.
- `DISTANCE` formulas are supported in:
 - Entry criteria for workflow rules and approval processes
 - Field update actions in workflow rules and approval processes
 - Custom validation rules
 - Lookup filters (in the Metadata API only)
- Geolocation fields and latitude and longitude on standard addresses aren't supported in Salesforce to Salesforce.
- In Developer, Professional, Enterprise, Unlimited, and Performance editions, Salesforce can automatically add or update geolocation fields for Account, Contact, Lead, and WorkOrder records. To use this feature, your administrator must enable the geo data integration rule for each object. For all other objects and editions, set values for latitude and longitude by using SOQL, SOAP or REST API, or a geocoding service. You can then use address fields as locatable values. To find geocoding services, search AppExchange.
- Geolocation fields are supported in SOQL with these limitations.
 - `DISTANCE` and `GEOLOCATION` are supported in `WHERE` and `ORDER BY` clauses in SOQL, but not in `GROUP BY`. `DISTANCE` is supported in `SELECT` clauses.
 - `DISTANCE` supports only the logical operators `>` and `<`, returning values within (`<`) or beyond (`>`) a

specified radius.

- When using the `GEOLOCATION` function in SOQL queries, the geolocation field must precede the latitude and longitude coordinates. For example, `DISTANCE(warehouse_location__c, GEOLOCATION(37.775,-122.418), 'km')` works but `DISTANCE(GEOLOCATION(37.775,-122.418), warehouse_location__c, 'km')` doesn't work.
- Apex bind variables aren't supported for the units parameter in the `DISTANCE` function. This query doesn't work.

```
String units = 'mi';
List<Account> accountList =
    [SELECT ID, Name, BillingLatitude, BillingLongitude
     FROM Account
     WHERE DISTANCE(My_Location_Field__c, GEOLOCATION(10,10), :units) < 1
0];
```

For more information and examples, see the [SOQL and SOSL Reference](#).

See Also

[Custom Field Types](#)

[Custom Field Attributes](#)

Considerations for the Time Custom Field Type

Track time, unbound to a date, with the Time custom field type, which is useful for time management, event planning, and project management.

You can select the Time field type when you create a custom field. The Time type is a timestamp without the date included. A Time field value's precision is in milliseconds. A date/time field value's precision is in seconds. Use the time field type when you require a time of day that isn't specific to a single date. For example, use it to show business hours, or if you want to compare times of day to calculate a duration.

Time Field Behavior

Time field values follow these behaviors.

- In formulas, get a time value from the `TIMEVALUE` function. Use `TIMENOW`, `HOUR`, `MINUTE`, `SECOND`, and `MILLISECOND` functions with time in formulas.
- The Time field type is represented in Metadata API as a FieldType enumeration value.
- A time field value is saved according to the saving user's time zone. When the time value is displayed in Salesforce, the value matches what's saved. It isn't converted to a user's local time zone per the user's Locale setting on the Language & Time Zone page.
- The stored time value is appended with a "Z", but the time isn't UTC, it's the time of the saving user's time zone. When you query a time field value using SOQL or the API, disregard the appended "Z." For example, a user saves the time, which is 7:00 AM in their time zone. When you query the time via

SOQL or the API 07:00:00.000Z is returned.

- The time is saved in a 24-hour format. For example, 07:00:00.000Z is 7:00 AM, and 19:00:00.000Z is 7:00 PM.
- The unit for adding or subtracting a time value is milliseconds. For example, Timefield1__c has the value “5:00pm.”
 - `Timefield1__c + 600000` is “5:10pm”
 - `Timefield1__c - 600000` is “4:50pm”

Time fields don't include a date. So, adding 25 hours to a time value is the same as adding one hour. The clock restarts after 24 hours.

- You can subtract one time field from another in a formula. The result is in milliseconds. For example, TimeField1__c has the value “10:00pm” and TimeField2__c has the value “9:00pm”:

`- TimeField1__c - TimeField2__c` is 3600000

The result is never a negative number. Subtraction is the difference between two time values, using a 24-hour clock. For example, when calculating the number of hours that a business is open, you use this formula: `(ClosedTime - OpenTime) / 3600000`.

`- ClosedTime = 5 PM, OpenTime = 8 AM, ClosedTime - OpenTime = 9 hours`

`- ClosedTime = 5 AM, OpenTime = 7 AM, ClosedTime - OpenTime = 22 hours`

- In formula expressions, use the international date format (ISO) for text arguments. For example, use `TIMEVALUE("11:30:00.000")` instead of `TIMEVALUE("11:30 AM")`.

Time Field Limitations

Be aware of these limitations when using a field created from the Time field type. The time field:

- Isn't supported in Process Builder, and Schema Builder
- Doesn't support the creation of custom index for SOQL queries
- Isn't available for standard lookup relationships in external objects

Formatting Time Fields in Salesforce Classic

In Salesforce Classic, you have several formatting options when you set a time value. For example, you can set a time value to include seconds, milliseconds, time zone, and use 24-hour notation.

- The time custom field type can use 24-hour notation. You can save a time value in HH:MM, for example, 14:40.
- Time fields support the following input formats.

Format	Example
hh:mm:ss aa	10:30:25 AM
hh:mm:ss.SSS a	10:30:25.125 AM
HH:mm:ss.SSS	14:30:25.125
HH:mm:ss.SSSZ	14:30:25.125Z displays as GMT

Format	Example
hh:mm a	10:30 AM
hh:mm a	10:30AM
h a	4 PM
ha	4PM
H:mm	1:23 is 1:23 AM
H	14 is 2:00 PM
Hmm	123 is 1:23 AM
HHmm	1434 is 2:34 PM

h= Hour of day (1-12), H = Hour of day (0-23), m= minute, s= seconds, S= milliseconds, a= AM or PM, Z= GMT time zone.

- Use the 11:30:00.000Z format when loading values with Data Loader.
- Use the HH:MM:SS.SS format to set a default value for a field, such as `TIMEVALUE("10:30:00.000")` for 10:30 AM.

See Also

[Create a Custom Field](#)

[Locales Overview](#)

[Formula Operators and Functions by Context](#)

[Using Date, Date/Time, and Time Values in Formulas](#)

[Custom Field Attributes](#)

[Define Default Field Values](#)

Custom Field Attributes

A custom field has different attributes depending on its field type.

! **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Field	Description
# Visible Lines	<p>For Long Text Area fields, set the number of lines to be shown on Edit pages. You can show from 2 through 50 lines (the default is 6 lines). If the text doesn't fit in the specified number of visible lines, scroll bars appear. Long Text Area fields are fully shown on detail pages and printable views.</p> <p>This attribute's value isn't respected for Long Text Area and</p>

Field	Description
	Rich Text Area fields in Lightning Experience.
Calculation Options	Determines how a roll-up summary field is recalculated after its properties change. Select Automatic calculation to recalculate a field the next time it's displayed. Select Force a mass recalculation of this field as a safety net option to force recalculation of the roll-up summary field values.
Child Relationship Name	The name used in API SOQL relationship queries.
Data Type	The data type of a field determines what type of information is in the field. For example, a field with the Number data type contains a positive or negative integer. For more information on data types, see Custom Field Types .
Decimal Places	For currency, geolocation, number, and percent fields, this field represents the number of digits you can enter to the right of a decimal point. The system rounds the decimal numbers that you enter, if necessary. For example, if you enter 4.986 in a field with Decimal Places set to 2, the number rounds to 4.99.
Default Value	The value to apply when a user creates a record. For custom checkbox fields, select Checked or Unchecked as the default value to indicate the default when creating records. Don't assign default values to fields that are both required and unique, because uniqueness errors can result. See Default Field Values .
Description	Text that describes the custom field. This description is for administration purposes only and doesn't appear to users on record detail and edit pages that include the field.
Display Format	<p>For auto-number fields, enter a Display Format to control formatting details such as the minimum number of leading zeros and any prefix or suffix for the number.</p> <p>Begin by entering the required minimum <code>{0}</code> as a placeholder for the auto-number without any leading zeros. Add any prefix to your number before this placeholder and insert any suffix text after the placeholder. Insert any date prefixes or suffixes in the form of <code>{YY}</code>, <code>{YYYY}</code>, <code>{MM}</code>, or <code>{DD}</code>, which represent the record creation date in Greenwich Mean Time (GMT).</p>

Field	Description
	For information on using auto-number formats when entering your Display Format, see Auto-Number Formatting Examples .
Encrypted	If checked, this custom field is encrypted using Shield Platform Encryption. See Differences Between Classic Encryption and Shield Platform Encryption
External Column Name	<p>Available on external objects only. Maps the custom field to an external data source's table column.</p> <p>For a lookup relationship field, specify the external table column that contains 18-character Salesforce IDs.</p>
External ID	<p>For each object that can have custom fields, you can set up to 25 custom auto-number, email, number, or text fields as external IDs. An external ID field contains record identifiers from a system outside of Salesforce.</p> <p>You can use an external ID field to update or upsert records using the API. When using the API or the Data Import Wizard for custom objects and solutions, you can use this field to prevent duplicates by also marking the field as Unique.</p> <p>Custom fields marked as Unique count against an object's limit of 25 External ID fields. Custom indexing that occurs automatically in the background by Salesforce doesn't count against External ID limits.</p> <p>Not available for external objects. Each external object has an External ID standard field. Its values uniquely identify each external object record in your org.</p>
Filter Criteria	The criteria used to select a group of records to calculate the value of a roll-up summary field .
Filtering Disabled	For custom fields on external objects, determines whether the field is available in filters.
Formulas	Enter the formula for the custom formula field or the custom summary formula for reports.
Help Text	The text that appears in the field-level help hover text for this field.

Field	Description
Is Name Field	<p>For external object fields of type text, specifies this custom field as the name field for the external object. Not available for text area fields. By default, the External ID standard field is the name field for the external object.</p> <p>If you select this checkbox, make sure that the External Column Name specifies a table column that contains name values. Each external object can have only one name field.</p> <p>For internal use only, Salesforce stores the value of the name field from each row that's retrieved from the external system. This behavior doesn't apply to external objects that are associated with high-data-volume external data sources.</p>
Label	The name of the custom field as you want it to appear.
Latitude and Longitude Display Notation	<p>For geolocation fields, determines how the latitude and longitude notation appears in the Salesforce interface.</p> <p>Degrees, Minutes, Seconds A notation for angular measurement that is based on the number 60: there are 360 degrees to a circle, 60 minutes to a degree, and 60 seconds to a minute.</p>
	<p>Decimal Expresses the value as degrees, and converts the minutes and seconds to a decimal fraction of the degree. Decimal notation doesn't use cardinal points. North and East are positive values; South and West are negative values.</p> <p>For example, the coordinates for San Francisco can be expressed as:</p> <p>Latitude: 37° 46' 30" N, Longitude: 122° 25' 5" W or Latitude: 37.794016°, Longitude: -122.395016°</p> <p>Regardless of the notation you choose to show in the interface, latitude and longitude are stored in Salesforce as decimals.</p>
Length (for text fields)	For text fields, the maximum number of characters that a user can enter in a field (up to 255 characters).

Field	Description
Length (for number, currency, percent fields)	<p>For number, currency, and percent fields, the number of digits you can enter to the left of the decimal point, for example, 123.98 for an entry of 3.</p> <p>Changing the number of digits (either manually or through workflow rules) can result in truncated values in Salesforce Classic and no value changes in Lightning Experience. This example shows this behavior. In Lightning Experience, the record shows the percentage as 1,235,689.22% and in Salesforce Classic, the percentage appears as 689.22%.</p> <ul style="list-style-type: none"> • Create a percentage field with a length of 16 and 2 decimal places. • Create a record that includes the field with set the value to 1235689.22. • Change the percent field digits to a length of 3.
Mask Character	For encrypted text fields, determines the character to use for hidden characters. Available options are * and X.
Mask Type	<p>For text fields encrypted with Classic Encryption, determines which characters are hidden and the use of dashes in the field. Masked characters are hidden using the character selected in Mask Character. Available options are:</p> <p>Mask All Characters All characters in the field are hidden.</p> <p>Last Four Characters Clear All characters are hidden but the last four are shown.</p> <p>Credit Card Number The first 12 characters are hidden and the last four are shown. Salesforce automatically inserts a dash after every fourth character.</p> <p>National Insurance Number All characters are hidden. If the field contains nine characters, Salesforce automatically inserts spaces after each pair of characters. Use this option for UK NINO fields.</p>

Field	Description
	<p>Social Security Number The first five characters are hidden and the last four are shown. Salesforce automatically inserts a dash after the third and fifth characters.</p> <p>Social Insurance Number All characters are hidden but the last three are shown. Salesforce automatically inserts a dash after the third and sixth characters.</p>
Master Object	The object on the master side of a master-detail relationship used to show the value of a roll-up summary field .
Related List Label	For relationship fields, the title for the related list that shows associated records on the parent record.
Related To	For relationship fields, the name of the associated object.
Required	<p>Makes the field required everywhere in Salesforce. Not available for external objects.</p> <p>You must specify a Default Value for required campaign member custom fields.</p> <p>Don't assign default values to fields that are both required and unique, because uniqueness errors can result. See Require Field Input to Ensure Data Quality.</p>
Roll-Up Type	<p>For roll-up summary fields, choose the type of calculation to make:</p> <ul style="list-style-type: none"> • COUNT: Totals the number of related records. • SUM: Totals the values in the field that you select in the Field to Aggregate option. Only number, currency, and percent fields are available. • MIN: Displays the lowest value of the field that you select in the Field to Aggregate option for all directly related records. Only number, currency, percent, date, and date/time fields are available. • MAX: Displays the highest value of the field that you select in the Field to Aggregate option for all directly related records. Only number, currency, percent, date, and date/time fields are available.

Field	Description
	time fields are available.
Starting Number	<p>For auto-number fields, enter a Starting Number that's less than 1 billion. Select Generate Auto Number for existing records to automatically number all current records that begin with the starting number that you enter. If deselected, the next record that you enter is assigned the starting number and your older records are blank in this field. For leads, only unconverted leads are assigned a number.</p> <p>When you create records, the Starting Number value increments to store the number that will be assigned to the next auto-number field created. You can't edit Starting Number after creating an auto-number field. To edit a Starting Number value, change your auto-number field to a text field and then back to an auto-number field. To restart Starting Number values for fields on objects from a managed package, uninstall and then reinstall the package.</p> <p>Be sure that you don't create records with duplicate auto-number values.</p> <p>An auto-number field can contain up to 10 digits and up to 20 extra characters for your prefix or suffix.</p> <ul style="list-style-type: none"> You can't retrieve the starting number of an auto-number field through Metadata API. To specify a Starting Number while deploying, add a <code>startingNumber</code> tag for your field to your <code>package.xml</code> file. For example: <pre><startingNumber>42</startingNumber></pre> <ul style="list-style-type: none"> If you deploy without specifying a Starting Number value in your <code>package.xml</code> file, the default starting number for standard fields is <code>0</code>. The default starting number for custom fields is <code>1</code>.
Sharing Setting	For master-detail relationship fields, the Sharing Setting attribute determines the sharing access that users must have to a master record to create, edit, or delete its associated detail records.
Sorting Disabled	For custom fields on external objects, determines whether the field is sortable.

Field	Description
Summarized Object	The object on the detail side of a master-detail relationship used to provide the values calculated in a roll-up summary field .
Unique	<p>If checked, prevents duplicate field values.</p> <p>For text fields, you can control whether values that are identical except for their case are considered unique. Select Treat "ABC" and "abc" as duplicate values to enforce case-insensitive uniqueness, or select Treat "ABC" and "abc" as different values to enforce case-sensitive uniqueness.</p> <p>Some characters have both single-byte and double-byte versions. For example, all the following characters have single-byte and double-byte versions: “!@#\$%^&*(){}[]\ :‘;,.〈>?/~`”.</p> <p>For the purpose of unique field value comparison, the single-byte and double-byte versions of these characters are considered identical.</p> <p>Superscript characters aren't treated as unique characters. Text such as ABC² and ABC2 are considered identical.</p> <p>Custom fields marked as Unique count against an object's limit of 25 External ID fields. Custom indexing that occurs automatically in the background by Salesforce doesn't count against External ID limits.</p>
Values	For picklist fields, a list of available values (up to 255 characters for each value). For picklists, optionally choose to alphabetize the picklist entries. You can also set the first value as the default selection. If you select both options, Salesforce alphabetizes the entries and then sets the first alphabetized value as the default. For multi-select picklists, enter a list of values, select the sorting options that apply, and then enter how many values you want shown at a time on edit pages. The number of values determines the box height.

Auto-Number Formatting Examples

Use these examples when setting the display format for auto-number fields.

Format	Displayed Values
{0}	3 66 103
{000}	003 066 103
Sample- {00000}	Sample- 00003 Sample- 00666 Sample- 10023
Value- {00} {MM} {DD} {YY}	Value- 03 12 02 04 Value- 76 03 03 04 Value- 123 11 09 04
PO #{0} {MM}-{DD}-{YY}	PO #12233 12-20-04 PO #25 06-07-04 PO #3 07-07-04

See Also

- [Create a Custom Field](#)
- [Create a Many-to-Many Object Relationship](#)
- [Object Reference for Salesforce and Lightning Platform](#)

Custom Address Fields

Improve address data accuracy and the end-user experience with Custom Address Fields. Users can populate a custom address field manually or they can use the Google lookup to search for an address. Admins and APIs can access each address stored in a custom address field as a structured compound data type as well as individual address components.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

Use the Address field type to create custom fields that store address data in a structured, compound-data type. Compound fields are an abstraction that can simplify application code that handles the values, leading to more concise, understandable code. With Custom Address Fields, custom addresses are accessible as a single, structured field or as individual component fields. Users can edit the custom address field data in records and view custom address data in list views and reports.

[Considerations for Custom Address Fields](#)

Before you enable custom address fields, configure State and Country/Territory picklists and review these considerations.

[Enable Custom Address Fields](#)

After you review the feature limitations and configure the State and Country/Territory picklists, enable custom address fields. Then you can add custom fields to standard and custom objects in Object Manager using the compound Address field type.

[Create a Custom Address Field](#)

Use the Address field type to create a custom address field that mimics the behavior of standard

address fields. For example, add a Warehouse Address field to a standard or custom object. Users can populate a custom address field manually, or they can use the Google lookup to search for an address. Then you can access each address that's stored in a custom address field, either as a structured compound data type or as individual address components.

Considerations for Custom Address Fields

Before you enable custom address fields, configure State and Country/Territory picklists and review these considerations.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

Supported Functionality

With Custom Address Fields, your Salesforce end users can add and retrieve address data via custom Address compound fields on standard and custom objects. Users can edit the custom address field data in records and view custom address data in list views and reports.

You can use custom address fields with these features.

Google Address Lookup

When users populate a custom address field, they can enter an address manually or they can use Google lookup to search for an address. When a user selects an address from Google lookup, the street, city, state or province, zip or postal code, and country are populated. To populate address details, Maps and Location Services uses the [Google Maps Geocoding API](#). If the Geocoding API can't map or parse an address component, then Maps and Location Services can't autocomplete the address field.

State and Country/Territory Picklists

End users select state, province, country, and territory values from picklists when adding or editing addresses in custom address fields. Standard address fields work in parallel with new custom address fields. If state and country/territory picklists aren't enabled in your org, the State and Country components of the standard Address fields remain free text fields.

Validation Rules

For example, require that the street, city, state, and ZIP code are all populated before you can save a custom address field.

Apex Classes and Triggers

For example, you create a custom address field, Office Address, on the Opportunity object, and

you want Office Address to always have a value. You then create an Apex trigger that's invoked when the Office Address field is unpopulated upon saving an Opportunity. The trigger populates the unpopulated Office Address field with the Billing Address on the parent Account before the Opportunity record is saved.

List Views

For example, you create a custom address field, Warehouse Address, on the Account object. Include the individual components of the Warehouse Address, such as the street or state, in an Accounts list view.

Reports

For example, a report that sums the number of opportunities by the state or ZIP code within a custom address field.

Field History Tracking

Track and display the history of a custom address field in the History related list of an object.

Managed Packages

For example, include a custom address field in a managed package, or use a package to deploy that field to a sandbox.

Change Sets

Move objects with address fields created using Custom Address Fields from one Salesforce org to another.

Apex and API

To create, edit, or delete records with custom address field data, use Apex. To create a custom address field on an object, use Metadata API. To create, update, or delete a record with custom address data, use SOAP API or REST API. To retrieve information about custom address fields, such as the developer name, use Tooling API. For more information see the [Custom Address Fields Developer Guide](#).

Change Data Capture

Receive real-time events for changes in custom address fields for new, updated, and undeleted records.

Skinny Tables

To avoid joins and improve the performance of certain read-only operations, include custom address fields in skinny tables, which contain frequently used fields. See [Skinny Tables](#) in *Best Practices for Deployments with Large Data Volumes*.

Custom Indexes

To speed up queries, create custom indexes for custom fields created with the Address data type.

See [Indexes](#) in *Best Practices for Deployments with Large Data Volumes*.

Geocode

To give your users precise geographical information, add geocode information to custom address fields.

Considerations for State and Country/Territory Picklists

- Custom address fields use picklists for the State and Country address fields.
- If State and Country/Territory Picklists are enabled, those picklist values are used in standard address fields. With Custom Address Fields, the same picklist values are automatically available in custom address fields. You can't specify separate picklist values for standard and custom address fields.
- If State and Country/Territory Picklists aren't enabled, those picklists are enabled for custom address fields with Custom Address Fields. By default, all countries, territories, and their states and provinces are visible to users. To specify the available picklist values in Salesforce, configure State and Country/Territory Picklists.
- When you configure these picklist values, the behavior of standard address fields is unaffected unless you enable State and Country/Territory Picklists for standard fields through Setup. Enabling the picklists for standard fields isn't required to use Custom Address Fields.

For more information on configuring the picklists, see [Configure State and Country/Territory Picklists](#).

For details on enabling the picklists for standard address fields, see [Let Users Select States, Countries, and Territories from Picklists](#).

Requirement for Package Deployment

If a package contains a custom field with the Address field type, package deployment requires that Custom Address Fields is enabled in the target org.

Org Limits for Custom Address Fields

For custom compound fields, each component counts as one custom field toward your org's allocations. Thus each custom address field counts as nine custom fields: one each for street, city, postal code, country code, state code, geocode accuracy level, longitude, and latitude, plus one for internal use. For more information on the allocations for your org, see [Salesforce Features and Edition Allocations](#).

Limitations for Custom Address Fields

Before you enable Custom Address Fields or add a custom address field, understand the limitations of this feature. Custom address fields aren't supported in these features.

- The conversion of address data into custom fields of type Address from custom fields of other types.
- [Approvals](#)
- [Data Import Wizard](#)
- Fuzzy matching

- [Composite API](#)
- [Field Encryption](#)
- [Field Sets](#)
- [Flow Screen Input Component: Address](#)
- [Lead Conversion](#)
- [Lightning Web Components](#)
- [Mass Update Addresses](#)
- [Merge Fields](#)
- [Visualforce pages](#)
- [Workflow](#)

Salesforce hasn't validated custom address fields with these capabilities.

- [Schema Builder](#)
- [Web-to-Case and Email-to-Case](#)
- [Generating Leads from Your Website](#)
- [Export Backup Data from Salesforce](#)
- [Export Data](#)
- [Filtering in a related list](#)
- [Bulk API 1.0](#)
- [Community profile](#)

This functionality is either unavailable or limited with custom address fields.

- As with standard address fields, you can't mark a custom address field as required.
- You can't use the DISTANCE function with a custom address field.
- To export data stored in custom fields of type Address, use API or SOQL queries. Bulk API doesn't support the export of custom compound fields.
- The error message when you attempt to export a custom address field with Bulk API incorrectly states that the functionality isn't enabled. Bulk API doesn't support the export of custom compound fields.
- To populate a custom address field with imported data, use REST API or Bulk API 2.0.
- Search, including Global Search, lookup search, search manager, and SOSL queries, isn't supported.
- In Skinny Tables, you can't select a component of a custom address field as a partition column.
- Compound address fields aren't supported in reports. To include a custom address field in a report, add the individual address components, such as street, city, state, and zip.
- When using a custom address field in a Data Integration Rule, the Country and State components are unavailable for field mapping.
- You can't rename the labels for the individual components of a custom address field.
- You can localize the label of a custom address field. However, you can't localize the labels of the individual components within a custom address field.
- The word "Address" isn't appended to the section label for a custom address field. If you include the word "Address" in the field label, it's included in the label for every component. For example, "Warehouse Address (State)" instead of "Warehouse (State)". These labels are inconsistent with the label behavior for standard address fields.
- The length of the GeoCodeAccuracy field for custom fields of data type Address isn't consistent with standard field of type Address.

See Also

[Salesforce Features and Edition Allocations](#)
[Custom Address Fields](#)
[Enable Custom Address Fields](#)
[Create a Custom Address Field](#)

Enable Custom Address Fields

After you review the feature limitations and configure the State and Country/Territory picklists, enable custom address fields. Then you can add custom fields to standard and custom objects in Object Manager using the compound Address field type.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

USER PERMISSIONS NEEDED

To modify user interface settings:	Customize Application
------------------------------------	-----------------------

Before you enable custom address fields, review the [requirements and limitations](#).

1. In Setup, in the Quick Find box, enter *User Interface*, and then select **User Interface**.
2. In the Setup section, select **Use custom address fields** and save your changes.

After you enable custom address fields, the Address data type is available when you add a field via Object Manager.



Note This feature can't be disabled.

See Also

[Custom Address Fields](#)
[Create a Custom Address Field](#)
[Salesforce Features and Edition Allocations](#)

Create a Custom Address Field

Use the Address field type to create a custom address field that mimics the behavior of standard address fields. For example, add a Warehouse Address field to a standard or custom object. Users can populate a custom address field manually, or they can use the Google lookup to search for an address. Then you can access each address that's stored in a custom address field, either as a structured compound data type or as individual address components.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

USER PERMISSIONS NEEDED

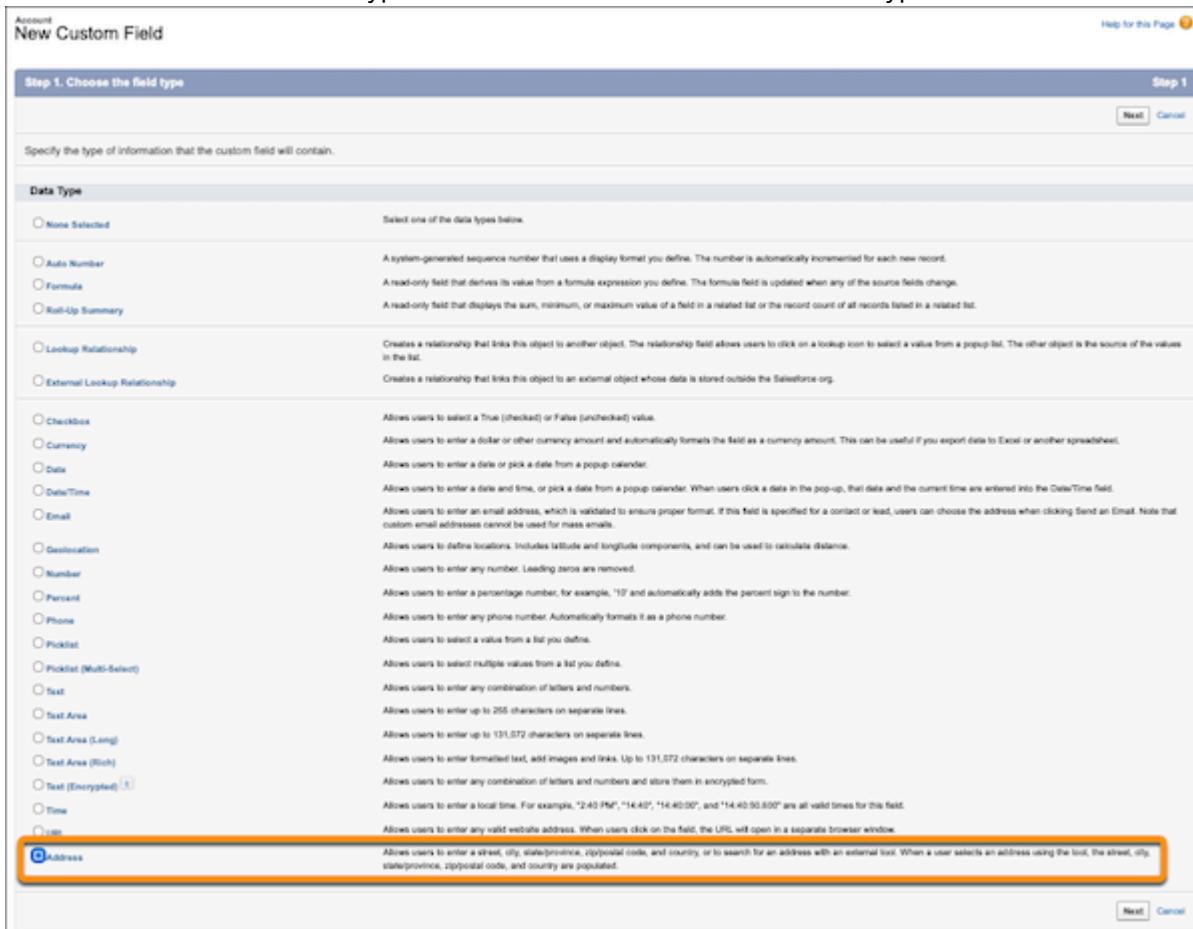
To create or change custom fields:

Customize Application

Before you can create a custom address field, enable this feature in User Interface Settings. For more information, see [Enable Custom Address Fields](#).

Before you create a custom address field, review the [requirements and limitations](#).

1. In your Salesforce org, click  and select **Setup**.
2. Click the Object Manager tab. If you don't see it, enter *Object Manager* in the Quick Find box and select **Object Manager**.
3. From the Object Manager page, select the object to which you want to add the custom address field.
4. From the sidebar, click **Fields & Relationships**.
5. To create a custom field, click **New**.
6. Choose **Address** as the data type. Address is at the bottom of the data type list.



The screenshot shows the 'New Custom Field' wizard, Step 1: Choose the field type. The 'Data Type' section lists various options, with 'Address' highlighted by a yellow border. The 'Address' option is described as allowing users to enter a street, city, state/province, zip/postal code, and country, or to search for an address with an external tool. When a user selects an address using the tool, the street, city, state/province, zip/postal code, and country are populated.

7. Click **Next**.

8. Enter the field label and field name.

The field label appears above the custom address field and at the beginning of each component label. When you specify your field label, consider whether the word Address is needed. In many cases, the field label doesn't require the word Address. To help API users, you can add the word Address to the field name.

Here's the label field when building or editing a field in Object Manager.

Account
New Custom Field

Step 2. Enter the details Step 2 of 4

Field Label

Field Name

Description

Help Text

Auto add to custom report type Add this field to existing custom report types that contain this entity

Previous Next Cancel

And here's the field on the Account object.

Warehouse
Warehouse_CountryCode)

--None--

Warehouse_Street)

Warehouse_City)

Warehouse_(StateCode)

--None--

Warehouse_PostalCode)

9. Optionally, enter a description and help text, and select whether to add the new field to custom reports. Then click **Next**.
10. Select the field's visibility and edit access.
11. Select the Dynamic Forms-enabled Lightning record pages that should include the field, then click **Next**.

If you don't have any Dynamic Forms-enabled Lightning record pages for the object, this step doesn't appear.

12. Click **Next**. Then click **Save**.
13. To edit the placement of the custom address field on the pages associated with this object, click **Page Layouts**.

See Also

[Custom Address Fields](#)

[Considerations for Custom Address Fields](#)

[Page Layouts](#)

Manage Fields for a Specific Object

You can add, edit, delete, and customize object fields in Setup.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Standard fields are not available in **Database.com**

Salesforce Connect external objects are available in Developer Edition and, for an extra cost, in Enterprise, Performance, and Unlimited Editions.

1. From the object management settings for the object whose fields you want to view, go to **Fields**.
2. Click the field label.
3. To modify a custom field, add custom help text, or change the data type, click **Edit**.
4. If a custom field exists in a Managed - Released package, click **Delete** to delete the custom field component from future installations.
5. To set users' access to the field, click **Set Field-Level Security**. This option is available depending on the edition that you have.
6. To view who can access the field based on permissions and record type, click **View Field Accessibility**. This option is available depending on the edition that you have.
7. If the custom field is a dependent picklist, click **Change** next to the controlling field to edit the dependency rules.
8. To change External ID, Required, or other attributes under the General Options section, see **Custom Field Attributes**.
9. To restore the field and its data, click **Undelete**. This option is available only if the field has been deleted but not permanently erased. The field's behavior may be different after restoring it.

See Also

[Edit a Custom Field](#)

[Delete a Custom Field](#)
[Manage Deleted Custom Fields](#)

Find Where a Field Is Used

Find out where a custom field is referenced, such as in a formula or Apex class, with a click of the **Where is this used?** button. You can see where a field is used and where changes to the field appear before you edit it.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Professional, Enterprise, Performance, and Unlimited Editions**

USER PERMISSIONS NEEDED

To view references: [View Setup](#)

On a custom field's detail page, click **Where is this used?** to see the field reference details. To view the settings for the layout, formula, or other reference, click a reference label.

The list can include these references.

- Active validation rule
- Layout
- Formula field
- Visualforce page
- Apex class
- Apex trigger
- Email template (Salesforce Classic, text based)
- Field set
- Flow (query)
- Lightning component markup (attr)
- Process Builder (criteria)
- URL button (formula)
- Lightning page (Related List–Single and Dynamic Related List–Single components)
- Lookup filter (lookup and master detail)
- Report (column, filter)

Considerations for Finding Fields

- References to reports on objects where there's a foreign key relationship don't appear. For example, if a custom object has a lookup relationship to Account or any standard object, and the custom fields of Account are added to a report of type CustomObject with Account, then the report isn't listed for the custom fields on Account.

- Reference labels link to more information only if there's a known settings page for the reference. For example, a report name links to the report settings. But a criteria formula created within a flow doesn't link to the flow settings.
- Within a subscriber org, references in a managed package aren't included in the list of results. For example, a number field is referenced in a formula. If you add the field to a package and then install the package in a subscriber org, the subscriber org's field reference detail page doesn't show that this number field is referenced in a formula field. But new references that are created after installing the managed package in the subscriber org do appear. For example, after you install the managed package and you add the number field to another formula in the subscriber org, the new reference appears.
- Only the IDs of reports that are accessible to the user initiating the query are returned. For example, if an admin creates a report and saves the report in a private folder, then the report isn't listed in the reference detail page for a standard user.
- Joined reports aren't supported and aren't displayed in the reference detail page.
- The list of field references is limited to the first 2,000 entries and sorted alphabetically by reference type.

See Also

[Manage Custom Fields for a Specific Object](#)

Edit a Custom Field

You can modify the field attributes of a custom field. The attributes vary according to the field data type.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

Standard Objects aren't available in **Database.com**

USER PERMISSIONS NEEDED

To create or change fields: Customize Application

! **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

1. From the management settings for the field's object, go to Fields & Relationships.
2. Click the field's name, then click **Edit**.
3. Modify the field attributes. The attributes differ depending on the field type.

If you're editing a picklist, you can change its definition and its values. For picklist settings, see [Add or Edit Picklist Values](#).

To change the type of this custom field, see [Change the Custom Field Type](#).

4. Optionally, define custom help text for the field.
5. For lookup and master-detail relationship fields, optionally define a lookup filter.
6. For formula fields, click **Next** to modify the formula.
7. In Enterprise, Unlimited, Performance, and Developer Editions, click **Next** to set the field-level security for the field.

Keep these considerations in mind when you edit a custom field.

- Editing fields can require changing a large number of records at once. To process these changes efficiently, your request can be queued and you receive an email notification when the process has been completed.
- To customize the way a custom object's related list appears on a parent record's detail page, edit the parent record's page layout. For example, if you want to edit which fields appear on a custom object's related list on accounts, you would edit the account page layout.
- You can't change the Field Name if a custom field is referenced in Apex.
- When editing fields for accounts, opportunities, cases, contacts, or custom objects, check for any criteria-based sharing rules that use the field in the rules. A field change can affect which records are shared.

See Also

[Define Default Field Values](#)

[Find Object Management Settings](#)

Delete a Custom Field

Custom fields can be deleted as long as they meet certain requirements.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To delete custom fields:

Customize Application

AND

View All Data

Before you delete a custom field, find out whether it's referenced by any other field, and if so, which fields. You can't delete a custom field that's referenced by another field. For example, you can't delete a custom field that's referenced by a field update or Apex.

1. From the management settings for the field's object, go to Fields.
2. Click **Del** next to the name of the field.
3. When prompted, select the **Yes, I want to delete the custom field** checkbox to confirm, and click **Delete**.

Keep these considerations in mind when you delete a custom field.

- Deleted custom fields and their data are stored until your org permanently deletes them or 15 days has elapsed, whichever happens first. Until that time, you can restore the field and its data. For information on restoring deleted custom fields and relationships, see [Manage Deleted Custom Fields](#).
- When you delete a custom field, all of the field history data is deleted and changes are no longer tracked.
- You can't delete a field if that field is being updated by a background job, such as an update to a roll-up summary field. Wait until the background job finishes, and then try again.
- A background process periodically runs that cleans up metadata associated with deleted custom fields. This process affects the Last Modified Date and Last Modified By fields on page layouts, record types, and custom objects.

See Also

[Find Object Management Settings](#)

Change the Data Type of a Custom Field

You can modify the data type of a custom field. For example, you can change a field from Text to Text Area (Long).

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

Standard Objects are not available in **Database.com**

Salesforce Connect external objects are available in Developer Edition and, for an extra cost, in Enterprise, Performance, and Unlimited Editions.

USER PERMISSIONS NEEDED

To change custom fields:

Customize Application

1. From the management settings for the field's object, go to Fields.

For fields on Salesforce Knowledge article types, from Setup, enter *Knowledge Article Types* in the **Quick Find** box, select **Knowledge Article Types**, and then select an article type.

2. Click **Edit** next to the custom field you want to change.

3. Click **Change Field Type**.
4. Select a new data type and click **Next**.
5. Enter a field label, name, and any other attributes, and then save your changes.

See Also

[Custom Field Types](#)

[Which Standard Fields Can I Encrypt?](#)

[Which Custom Fields Can I Encrypt?](#)

[Find Object Management Settings](#)

Considerations for Converting the Field Type of a Custom Field

Before you convert a custom field from one type to another, review these considerations.

-  **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Considerations for Potential Data Loss from Converting a Custom Field's Type

- To avoid losing data, only convert custom fields that have no data to avoid losing data. Changing the data type of an existing custom field can cause data loss in these situations.
 - Changing to or from type Date or Date/Time
 - Changing to Number from any other type
 - Changing to Percent from any other type
 - Changing to Currency from any other type
 - Changing from Checkbox to any other type
 - Changing from Picklist (Multi-Select) to any other type
 - Changing to Picklist (Multi-Select) from any other type

Currently defined picklist values are retained when you change a picklist to a multi-select picklist. If records contain values that aren't in the picklist definition, those values are deleted from those records when the data type changes.

 - Changing from Auto Number to any other type
 - Changing to Auto Number from any type except Text
 - Changing from Text to Picklist
 - Changing from Text Area (Long) to any type except Email, Phone, Text, Text Area, or URL
- If data is lost, any list view based on the custom field is deleted, and assignment and escalation rules can be affected.

Considerations for Changing a Custom Field's Data Type

- You can't change the data type of any custom field that is mapped for lead conversion.
- If you change the data type of a custom field that's set as an external ID, choosing a data type other than text, number, or email causes the field to no longer act as an external ID.

- The option to change the data type of a custom field isn't available for all data types. For example, an existing custom field can't be converted into an encrypted field nor can an encrypted field be converted into another data type.
- In Salesforce Knowledge article types, the file field type can't be converted into other data types.
- You can't change the data type of a custom field referenced by other items in Setup such as Visualforce pages, Apex code, processes, or flows.
- Before changing a custom field's type, make sure that it isn't the target of a workflow field update or referenced in a field update formula that would be invalidated by the new type.
- If you encrypt a custom field by using Shield Platform Encryption, you can't change the field type.
- If you change a custom field data type via the metadata `deploy()` method or a package upgrade and the deployment fails with an error related to custom field data types, see this [knowledge article](#). This error can occur if a deployment changes the data type of one or more custom fields and the objects that contain the affected custom fields contain a large amount of data. The limit is 85 million custom field type conversions. For example, if you have an object with 30 million records, and you change three text fields to picklists, that's a total of 90 million field type conversions, which is over the limit.
- When you change a custom field data type, the conversion runs in the background. The conversion can take a while depending on the size of the custom field, the number of records affected, and the type of field conversion. In some cases, the conversion can take over 24 hours to complete. These field data type changes take the longest.
 - Changing from Picklist to Text, Picklist (Multi-Select), or Checkbox
 - Changing from Text to Picklist
 - Change from Date/Time to Time
 - Changing from Time to Text
- Changing a custom field type can require changing many records at once. If your request is queued to process these changes, you receive an email notification when the process has completed.

Data Type Restrictions

These data types have other restrictions when you convert them.

Data Type	Description
Auto Number	If you convert an auto-number field into a text field, the data in that field remains unchanged. Also, you can safely convert a text custom field into an auto-number field without losing your data. Converting an auto-number field into any other data type results in data loss. Auto-number fields can contain a maximum of 30 characters. Before converting a text custom field into an auto-number field, change any records that contain more than 30 characters in that field.
Formula	Formula fields are special read-only fields that

Data Type	Description
	can't be converted to any other data type. Likewise, you can't convert any other field type into a formula field.
Picklist	Changing your custom picklists into custom checkboxes is simple. If you select Checkbox as the new data type, you can choose which picklist values to map to checked boxes and unchecked boxes. You can change custom picklists into multi-select picklists without losing any data. Since your records contain only a single value of that picklist, that value is still selected but users can select more values.
Relationships	<ul style="list-style-type: none"> • You can convert relationship fields to nonrelationship fields and vice versa, but only on external objects. • If your org has a large number of records, Salesforce displays a waiting page after you request to change a master-detail into a lookup relationship or a lookup into a master-detail relationship. • Converting a field from a master-detail into a lookup relationship or vice versa can fail if the number of affected records is very large. • After you create a roll-up summary field on an object, you can't convert the object's master-detail relationship into a lookup relationship. • If any records on an object have a null value set for the lookup field, you can't convert the lookup relationship to a master detail relationship. • If you convert a master-detail relationship to a lookup for a custom object on the "detail" side, the org-wide default for the object is automatically updated to Public Read/Write. Similarly, converting a lookup to a master-detail-relationship changes the org-wide default to Controlled by Parent.
Text	By default, an upper bound limit of 4,000 is

Data Type	Description
	<p>applied to inactive unrestricted picklists for each field. If you exceed this limit when converting a text field to a picklist field, you receive an error message. You can't convert a defined unique text field to a picklist or a multi-select picklist.</p> <p>If you deploy an object definition in package.xml and convert a field in that object from a text field to a picklist field, the label of each picklist value is the value's fullName.</p>
Text Area (Long)	<p>When you convert a long text area field to an Email, Phone, Text, Text Area, or URL type field, the record data is truncated to the first 255 characters of the field.</p>
Text Area (Rich)	<p>You can convert rich text area fields into long text area fields only. Any images get deleted the next time you save the long text area field. After converting, markup is hidden in the long text area field, but it isn't removed from the record, so if you change your mind, you can restore the markup before you save the record.</p>

See Also

[Change the Data Type of a Custom Field](#)

[Custom Metadata Type Fields](#)

Manage Deleted Custom Fields

Deleted custom fields and their data are stored until your org permanently deletes them or 15 days have elapsed, whichever happens first. Until that time, you can restore the field and its data.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Page Layouts and Lead Fields aren't available in **Database.com**

USER PERMISSIONS NEEDED

To restore deleted custom fields and relationships: Customize Application

To permanently delete custom fields or relationships: Customize Application

The field counts against the maximum number of custom fields allowed in your org until the hard-delete process permanently deletes it (see [Considerations for Hard Deleting Custom Fields](#)). A deleted field also counts against the applicable limit for its field type. For example, a deleted custom roll-up summary field counts against the maximum number of roll-up summary fields for your Salesforce edition. If you receive the error message “Unable to access page” or “No clean data columns available for custom fields” when trying to create or edit custom fields, you must erase some fields.

Considerations for Restored Custom Fields

- When deleted, the following characters are appended to the end of a custom field's developer name unless a deleted field already has that developer name: “_del”. These characters remain when you restore the custom field.
- Formula fields are restored in a disabled state, which means they don't contain updated data until you edit and save them. While a formula field is disabled, “#Error!” displays in place of the formula value.
- Restored fields don't display in search results immediately after you restore them. It can take a short time before the restored custom field and its data are available in search results.
- Lead fields that are mapped to account, contact, or opportunity fields for lead conversion are still mapped accordingly when restored.
- Auto number fields continue to increment after they're deleted and contain the correct values when restored.
- Field history data for the deleted custom field is restored.

Considerations for Hard Deleting Custom Fields

- The estimated time for a hard-delete process varies, and depends on the system demand to maintain overall performance. However, you can monitor and view the details of a hard delete of a custom field by going to the Background Jobs page. From Setup, enter *Background Jobs* in the Quick Find box, then select **Background Jobs**. A row with the name “Cleanup of custom field data when a custom field definition is hard deleted” indicates that a hard delete job is in progress. The Background Jobs page also shows the details of background jobs, including a percentage estimate of the progress.
- When an org is at or above 75% of its custom field allocation limit for the current object, Salesforce displays a value for the custom fields that are ready for the hard-delete process value. This value increments when you click **Erase** or when a field is beyond the 15-day grace period, until the field is moved into the hard-delete process by Salesforce. The value changes only while Salesforce is actively hard deleting the field.
- Performance and Unlimited Edition orgs can use a Purge button to initiate the hard-delete process immediately.

See Also

- [Manage Deleted Fields in Lightning Experience](#)
- [Manage Deleted Fields in Salesforce Classic](#)
- [Manually Restore Attributes of Deleted Fields](#)
- [Purge Deleted Custom Fields](#)
- [Find Object Management Settings](#)

[Manage Deleted Fields in Lightning Experience](#)

Restore or permanently delete a custom field in Lightning Experience.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Page Layouts and Lead Fields are not available in **Database.com**

USER PERMISSIONS NEEDED

To restore deleted custom fields and relationships: **Customize Application**

To permanently delete custom fields or relationships: **Customize Application**

1. From the Object Manager page, click the name of the custom object.
2. Click **Fields & Relationships**.
3. To see a list of soft deleted fields, click **Deleted Fields** at the top of the Custom Fields & Relationships page.
4. From the list of deleted fields, perform the following actions:
 - a. To permanently remove the custom field and its data, click **Erase**.
 - b. To restore the field and its data, click **Undelete**.If you undelete a custom field, manually restore attributes of the deleted field that weren't automatically restored.

See Also

- [Manually Restore Attributes of Deleted Fields](#)

[Manage Deleted Fields in Salesforce Classic](#)

Restore or permanently delete a custom field in Salesforce Classic.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Page Layouts and Lead Fields are not available in **Database.com**

USER PERMISSIONS NEEDED

To restore deleted custom fields and relationships: Customize Application

To permanently delete custom fields or relationships: Customize Application

1. From the management settings for the field's object, go to Fields.
2. Click **Deleted Fields** at the bottom of the list of custom fields and relationships. The number in parentheses indicates the total number of deleted custom fields for this object. For Performance and Unlimited Editions, the Deleted Fields link always displays. For other editions, this link displays only after you've deleted a custom field.
3. Use the list of deleted fields to perform these actions.
 - a. To view details about a field, click the field label.
 - b. To permanently remove the custom field and its data, click **Erase**.
 - c. To restore the field and its data, click **Undelete**.
If you undelete a custom field, manually restore attributes of the deleted field that weren't automatically restored.

See Also

[Manually Restore Attributes of Deleted Fields](#)

Manually Restore Attributes of Deleted Fields

Custom fields consist of several attributes that provide additional information about the field. Some of these attributes are not restored automatically after a field is undeleted and must be restored manually.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Page Layouts and Lead Fields are not available in **Database.com**

USER PERMISSIONS NEEDED

To restore deleted custom fields and relationships: Customize Application

To permanently delete custom fields or relationships: Customize Application

! **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

1. Add the field to any page layouts that changed during the time the custom field was deleted. If reports and page layouts were not edited, the restored field remains on them.
2. Make the field unique if necessary. Salesforce automatically removes the unique attribute from any deleted custom field.
3. Make the field required if necessary. Salesforce automatically removes the required attribute for any deleted custom field.
4. Add the custom field to any appropriate Salesforce AppExchange packages. Salesforce automatically removes deleted custom fields from packages that contain them.
5. Convert lookup relationships to a master-detail relationship if necessary. Salesforce converts all relationships to lookup relationships when they are deleted. To convert a lookup relationship to a master-detail relationship, populate all the applicable records with the appropriate data.
6. [Redefine any field dependencies](#) that Salesforce removed when the field was deleted.
7. Edit and save any formula fields. Saving prompts a syntax check; if necessary, resolve errors.
8. Set up field history tracking if necessary. If the list of fields enabled for history tracking was modified during the deletion process, the restored field is no longer setup to track field history.

Purge Deleted Custom Fields

Custom fields you delete from an object are removed from the object but remain in the system for 15 days until they're hard deleted automatically by Salesforce. During that time, the deleted custom fields continue to count toward your custom field allocation. When your org reaches 75% of its custom field allocation and it has custom fields that are ready to hard delete, Salesforce displays the Purge button. You can use it to initiate the hard delete process and free up custom field allocation for your org before the hard delete takes place.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To purge deleted fields: Customize Application

USER PERMISSIONS NEEDED

AND

View All Data

1. Go to the object's Deleted Fields detail page to find its recently deleted custom fields.
2. Click **Purge** to free up custom fields faster than the automated process.
3. Confirm the purge action when Salesforce prompts you to do so.

The custom fields that are in the ready for the hard-delete state move to the hard-delete process state. The hard-delete process completely removes the field and the associated system data. If changing the status of the fields takes longer than 30 seconds, Salesforce sends an email with more information to the user who clicked the button.

See Also

[Manage Deleted Custom Fields](#)

Editing Rich Text Area Fields in Records

Use rich text area fields to improve the appearance of text, including adding images and hyperlinks.

REQUIRED EDITIONS

Available in: the Salesforce mobile app, Salesforce Classic, and Lightning Experience

Available in: **Essentials, Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create or change custom fields:	Customize Application
------------------------------------	-----------------------

Rich text area fields use the [rich text editor](#) to format content. The rich text editors for custom fields in Lightning Experience and the Salesforce mobile app have a few differences compared to rich text editors in Salesforce Classic.

In Lightning Experience and the Salesforce mobile app, rich text editors for custom fields use the open-source Quill library. In Salesforce Classic, rich text editors for custom fields use CKEditor. In Lightning Experience, Salesforce Knowledge and emails also use the CKEditor.

Rich Text Editor Toolbar Button for Custom Fields	In Lightning Experience and the Salesforce mobile app	In Salesforce Classic
Color		
Format Font (Bold, Italic,		

Rich Text Editor Toolbar Button for Custom Fields	In Lightning Experience and the Salesforce mobile app	In Salesforce Classic
Underline, Strikethrough)		
Format Body (Bulleted List, Numbered List, Indent, and Outdent)	✓	✓
Align Text (Left Align, Center Align, and Right Align)	✓	✓
Insert Link	✓	✓
Insert Image	✓	✓
Remove Formatting	✓	
Undo and Redo Last Action		✓

 **Note** We recommend using the toolbar to format your content. The rich text editor provides only a WYSIWYG interface. You can't edit HTML tags. When you copy content from a web page or another source and paste it into the editor, unsupported tags are removed. Text enclosed in unsupported tags is preserved as plain text. HTML markup counts against the character limit of the field. For more information, see [Rich Text Area Field Considerations](#).

Note these differences across rich text editors in Lightning Experience and the Salesforce mobile app compared to Salesforce Classic.

In Lightning Experience and the Salesforce mobile app:

- Spaces are considered non-empty values.
- The default font family is the native sans-serif font on your system or device.
- The **Insert Link** button enables you to enter a URL with the `_blank` target value by default. This button appears disabled when the record edit page is loaded, and is enabled after you activate the editor with your keyboard or mouse.
- The **Insert Image** button enables you to insert an image. You can upload an image using the system file picker only. The image upload can vary across rich text editors in different feature areas. For example, in the Chatter publisher on a record page, you can also select an image that has been previously uploaded to your org.
- The **Left Align** button inserts a `<p>` element around the content.
- Use keyboard shortcuts to undo and redo content formatting. In Windows, undo your last action by pressing Ctrl+Z, and reverse your last undo by pressing Ctrl+Y. On Mac OS, use Cmd+Z and Cmd+Y. Alternatively, use the Edit menu in your browser to undo or redo your changes.

In Salesforce Classic:

- Spaces are considered empty values.

- The default font family is Arial, Verdana, Trebuchet MS, sans-serif.
- Color formatting is preserved when you switch to Salesforce Classic to edit a custom field that's been formatted in Lightning Experience or the Salesforce mobile app. This occurs even though the **Color** button isn't available in Salesforce Classic.
- The **Insert Link** button lets you enter a URL with a selection of protocols and target values. When you switch to Lightning Experience or the Salesforce mobile app and edit the custom field, unsupported protocols or target values aren't preserved.
- The **Insert Image** button lets you insert an image by uploading it or reference one that's hosted on another server.
- The **Left Align** button initially inserts a `<p>` element around the content. If you remove the left alignment formatting and reuse the **Left Align** button on existing text that was previously left-aligned, extra `<div>` tags are added.

Implementation Tips

- Specify the size of the editor box for a rich text field by configuring the **Number of lines displayed** property in the field's setup.
- When you view or print content, Salesforce preserves the formatted version of the rich text.
- Searches of content that contains rich text area fields ignore images and tags.
- Deleting a rich text area field moves it to the Deleted Fields section on the custom object or Salesforce Knowledge article types.
- You can convert rich text area fields into long text area fields only. Any images get deleted the next time you save the long text area field. After converting, markup is hidden in the long text area field but it isn't removed from the record, so if you change your mind, you can restore the markup before you save the record.
- The text part of a rich text area field counts toward data storage for the object that contains the field.

Pasting Content from External Sources

- You can copy and paste text from external sources, such as Microsoft® Word, but sometimes doing so requires reapplying the formatting.
- Text from external sources can include HTML tags and special formatting that you can't see and don't need. The tags are counted against the character limit of a field. We recommend pasting text into a plain text editor such as Notepad on Windows orTextEdit on macOS first. Copy the text from the plain text editor, paste it into a rich text field, and apply formatting using the rich text field's buttons.
- JavaScript and CSS are treated as text. For example, if you're creating an Idea through the API, the JavaScript or CSS code is removed without warning. Salesforce supports a limited number of approved HTML tags.
- When a rich text area field is used in a formula, the HTML tags are stripped out before the formula is run.
- Rich text area fields can be filtered and summarized in reports, but HTML tags and special formatting aren't included in report results. For example, `some <i>text</i>` becomes "some text" instead of `some <i>text</i>` or some text.
- You can use a rich text area field in a mail merge, but the HTML tags are included as text in the

resulting document. Images aren't merged.

Images in Rich Text Area Fields

- The maximum size of an image that can be uploaded in a rich text area field is 1 MB. You can upload only .gif, .jpg, and .png file types.
- To upload many images, use API version 20 or later.



Note When you upload images via the API, the `alt` attribute isn't populated unless you specified it separately.

- Images uploaded into a rich text area field are extracted in your org's weekly export and included in the exported data.
- Images in rich text area fields count toward file storage for the object that contains the field.
- You can't add a hyperlink to an image.
- You can't upload an image to a rich text area using the `file://` protocol in the **URL** field. Instead, use `http://`, `https://`, `data://`, `//`, `/`, or a relative URL.
- You can't resize images in Lightning Experience and the Salesforce mobile app. An exception to this is in Lightning Knowledge when using the Chrome browser.

Formatting Support

The power of the rich text editor is in its WYSIWYG interface. Type in the editor and use the toolbar to format your content as much as you can. When you paste formatted content from another source, you can expect some formatting differences in Lightning Experience and the Salesforce mobile app. Here are formatting considerations to look out for.



Warning If you add a custom rich text area field in Salesforce Classic and edit it in Lightning Experience, you can also expect the formatting differences. Saving your changes in Lightning Experience overwrites the original formatting you had in Salesforce Classic and conversely. Alternatively, you can fix some of the formatting using the toolbar or switch to Salesforce Classic to perform your edits.

Colors

The **Color** button is available in Lightning Experience only. Color formatting is preserved when you edit the rich text field in Salesforce Classic. The rich text field in Lightning Experience handles color formatting in RGB format, which adds the `` tag around your text. The rich text field in Salesforce Classic handles color formatting in hexadecimal format, which adds the `` tag around your text. The RGB and hexadecimal value depends on the color you choose.

Definition lists

Formatting for definition lists is preserved, but the styling appears different in Lightning

Experience and the Salesforce mobile app. Definition terms are not bold.

Fonts

Pasting text with a predefined font face, color, or size converts the `font` tag into a `span` tag with the `style` attribute.

Headings

Heading styles are different in Lightning Experience and Salesforce Classic. Headings in Salesforce Classic are bold and become smaller in size as the header level increases in number. Headings in Lightning Experience have the following font size and weight.

- `h1` : 24 px (not bold)
- `h2` : 18 px (bold)
- `h3` : 18 px (not bold)
- `h4` : 14 px (bold)
- `h5` : 14 px (not bold)
- `h6` : 12 px (bold)

Inline styles on `h1`, `h2`, `h3`, `h4`, `h5`, and `h6` tags are not supported and are ignored.

Hyperlinks

Hyperlinks always open in a new window or tab in Lightning Experience and the Salesforce mobile app using `target="_blank"`. In Salesforce Classic, you can use different `target` values in hyperlinks in the rich text field. If you edit a rich text field with such a link in Lightning Experience or the app, the `target` value is converted to `_blank`.

Inline styles

In Salesforce Classic, inline styles are supported on `div`, `span`, `p`, `br`, and `hr` tags. In Lightning Experience and the Salesforce mobile app, inline styles are supported only on `span` tags.

Lists

Nested ordered lists in rich text fields are numbered differently in Salesforce Classic than they are in Lightning Experience and the Salesforce mobile app. In Salesforce Classic, nested ordered lists are numbered with this pattern: 1, 1, 1. In Lightning Experience and the Salesforce mobile app, nested ordered lists are numbered with this pattern: 1, a, i.

In Lightning Experience and the Salesforce mobile app, you can't nest a bulleted list within a numbered list. The nested bulleted list is converted to a numbered list when you paste it into the editor.

You also can't nest a numbered list within a bulleted list. The nested numbered list is converted to a bulleted list when you paste it into the editor.



Warning Nesting lists of different types is supported in Salesforce Classic only. Don't edit an existing rich text area field containing a nested list of a different type in Lightning Experience and the Salesforce mobile app. If you do, the nested list is converted to the same type as its parent list, even if you don't edit the list itself.

Nested lists of the same type are supported. However, pasting a nested list into the editor flattens the list into one list in Lightning Experience and the Salesforce mobile app. For example, a nested bulleted list within a bulleted list becomes a single bulleted list when pasted. For nested lists of the same type, use the toolbar to adjust your list formatting. You can also press the Tab key or Shift+Tab to create a nested list item or remove a nested list item.



Note In Lightning Experience and the Salesforce mobile app, pasting lists that are copied from Microsoft® Word is not supported and results in list items getting converted into paragraphs. Although they look visually like lists, they are pasted as `p` tags.

Paragraphs and line breaks

Pressing the Enter key creates a paragraph in Lightning Experience and the Salesforce mobile app. However, pressing the Enter key in Salesforce Classic adds a `
` element inside the current paragraph. The visual difference is minor.

Markup for blank lines is `<p>
</p>` in Lightning Experience and the Salesforce mobile app, and `
` in Salesforce Classic. If you create text in a rich text field in Lightning Experience or the Salesforce mobile app and later edit it in Salesforce Classic, markup for the blank lines is converted to `<p> </p>`. Subsequent edits in Lightning Experience or the Salesforce mobile app don't change blank lines back to their original markup.

Paragraphs in rich text editors add extra characters to the text when saved in Lightning Experience and the Salesforce mobile app. Each paragraph contributes seven characters `<p></p>` to the character count. For example, if your admin specifies a 5000 character limit, you can enter only 4993 characters in the rich text editor of a custom field in Lightning Experience. Breaking the text into multiple paragraphs further reduces the number of characters you can enter. Blank lines contribute no visible characters, but contribute several characters with their markup.



Note RichText field values are returned in HTML format. That means it might contain block elements. Nesting block elements as merge fields in inline elements can cause formatting issues in HTML. If you're using the RichText custom field as a merge field, consider applying HTML customizations, like fonts, to the custom field itself. If you want only text data in the custom field, use TextField instead.

Special Characters

Since rich text field values are returned in HTML format, some characters are escaped when the

custom field value is retrieved through the API.

- Ampersand character & is returned as &
- Greater than character > is returned as >
- Less than character < is returned as <
- Quote character " is returned as "
- Single quote character ' is returned as '
- Some special characters, such as emoji, are rendered in inconsistent formats. The return value depends on your platform and the receiving end of the API response.

Tables

In Salesforce Classic, Lightning Experience, and the Salesforce mobile app, pasting tables is supported, but you can edit only the content within the tables.

Text-level markup

- Addresses with `address` tags cause enclosing list formatting to be removed.
- Nested quotes using `q` tags are not supported.
- Strikethrough text uses `strike` tags.
- Teletype text within `tt` tags is converted to use `code` tags.

This table lists supported HTML tags and formatting considerations in Lightning Experience and the Salesforce mobile app. When you edit a rich text field or paste text with unsupported tags in Lightning Experience and the Salesforce mobile app, those tags are removed. Text that was enclosed in unsupported tags is preserved as plain text.

HTML Tags Supported in Salesforce Classic	Supported in Custom Fields in Lightning Experience and the Salesforce Mobile App?	Comments for Lightning Experience and the Salesforce Mobile App
<code>a</code>	✓	The <code>target</code> attribute is always set to <code>_blank</code> . This tag is removed if <code>href</code> has no value.
<code>abbr</code>	✓	
<code>acronym</code>		This tag is converted into an <code>abbr</code> tag.
<code>address</code>	✓	When text with this tag is nested in a list and pasted in the editor, the list tags, <code>ol</code> , <code>li</code> , and <code>ul</code> , are removed. Some tags before the <code>address</code> tag get nested in <code>address</code> .
<code>b</code>	✓	
<code>bdo</code>	✓	
<code>big</code>	✓	
<code>blockquote</code>	✓	Consecutive block quotes are merged.

HTML Tags Supported in Salesforce Classic	Supported in Custom Fields in Lightning Experience and the Salesforce Mobile App?	Comments for Lightning Experience and the Salesforce Mobile App
<code>br</code>	✓	Line breaks are nested in <code>p</code> tags.
<code>caption</code>	✓	
<code>cite</code>	✓	
<code>code</code>	✓	
<code>col</code>	✓	
<code>colgroup</code>	✓	
<code>dd</code>	✓	
<code>del</code>	✓	
<code>dfn</code>	✓	
<code>div</code>	✓	
<code>dl</code>	✓	
<code>dt</code>	✓	Definition terms are not bold.
<code>em</code>	✓	
<code>font</code>		Pasting text with a predefined font face, color, or size converts the <code>font</code> tag into a <code>span</code> tag with the <code>style</code> attribute.
<code>h1</code>	✓	Inline styles on headings are not supported.
<code>h2</code>	✓	Inline styles on headings are not supported.
<code>h3</code>	✓	Inline styles on headings are not supported.
<code>h4</code>	✓	Inline styles on headings are not supported.
<code>h5</code>	✓	Inline styles on headings are not supported.
<code>h6</code>	✓	Inline styles on headings are not supported.
<code>hr</code>	✓	
<code>i</code>	✓	
<code>img</code>	✓	
<code>ins</code>	✓	

HTML Tags Supported in Salesforce Classic	Supported in Custom Fields in Lightning Experience and the Salesforce Mobile App?	Comments for Lightning Experience and the Salesforce Mobile App
<code>kbd</code>	✓	
<code>li</code>	✓	Pasting a nested list of the same type flattens the list into one list, for example, a nested bulleted list within a bulleted list. Use the toolbar to adjust your list formatting.
<code>ol</code>	✓	Nesting an ordered list (numbered) in an unordered list (bulleted) converts the ordered list into an unordered list.
<code>p</code>	✓	Pressing Enter creates a new <code>p</code> .
<code>pre</code>	✓	
<code>q</code>	✓	Nested quotes are not supported.
<code>s</code>		This tag is converted into a <code>strike</code> tag.
<code>samp</code>	✓	
<code>small</code>	✓	
<code>span</code>	✓	This tag is nested in a <code>p</code> tag, which adds extra padding around the text.
<code>strike</code>	✓	
<code>strong</code>	✓	
<code>sub</code>	✓	Nested <code>sub</code> tags are merged.
<code>sup</code>	✓	Nested <code>sup</code> tags are merged.
<code>table</code>	✓	
<code>tbody</code>	✓	
<code>td</code>	✓	
<code>tfoot</code>	✓	
<code>th</code>	✓	
<code>thead</code>	✓	
<code>tr</code>	✓	
<code>tt</code>		This tag is converted into a <code>code</code> tag.

HTML Tags Supported in Salesforce Classic	Supported in Custom Fields in Lightning Experience and the Salesforce Mobile App?	Comments for Lightning Experience and the Salesforce Mobile App
<code>u</code>	✓	
<code>ul</code>	✓	Nesting an unordered list (bulleted) in an ordered list (numbered) converts the unordered list into an ordered list.
<code>var</code>	✓	

See Also

[Create a Custom Field](#)

[Rich Text Fields in Knowledge Articles](#)

Rich Text Area Field Considerations

Keep these considerations in mind when working with rich text area fields.

REQUIRED EDITIONS

Available in: the Salesforce mobile app, Salesforce Classic, and Lightning Experience

Available in: **Essentials, Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

General

- Rich text area fields aren't available in self-service portals and they aren't supported for external objects.
- When a rich text area field containing an image larger than 50x50 pixels is placed as the first field on a compact layout or in the Highlights Panel component on a Lightning page, the image is cropped. In any other position, the image appears as expected.

Character Limits

- Salesforce supports up to 131,072 characters for each rich text area field, including the HTML tags. You can also set a lower limit.
- An object can contain unlimited rich text area and long text area fields, although your edition's allocation for total custom fields allowed on an object, regardless of field type, applies. Each object can contain 1,638,400 characters across long text area and rich text area fields. When you create a long text area or rich text area field, you set a character limit for the field—the maximum length of the text that can be entered. The default character limit for long text area and rich text area fields is 32,768 (32 KB). The minimum character limit is 256.

- You can't paste special characters, such as bullets or curly quotes, into a rich text field from another application. It's best to type or paste in plain text and use the rich text editor to format it.
- HTML code isn't supported in the [Salesforce HTML editor](#). HTML code is treated as text.
- The character count includes HTML markup that isn't visible in the editor. The HTML markup is returned through the API. For example, bold formatting includes the `` tag around your text and adds up to 7 more characters. Special characters like `&` are encoded as `&` which adds up to 5 more characters. Also, the rich text field in Lightning Experience and Salesforce Classic can vary in HTML markup, such as with using RGB or hexadecimal values when color formatting is applied. Paragraph and line breaks also insert the `<p></p>` and `
` tags, counting against the character limit. See [Editing Rich Text Area Fields in Records](#).

Formatting and Toolbar Differences Between Lightning Experience and Salesforce Classic

The rich text editors for custom fields in Lightning Experience and the Salesforce mobile app come with a few differences, as compared to rich text editors in Salesforce Classic. See [Editing Rich Text Area Fields in Records](#).

Images and Hyperlinks

- You can't disable specific rich text area features. For example, you can't disable support for hyperlinks or images on certain fields.
- You can't upload an image to a rich text area using the `file://` protocol in the URL field. Instead, use `http://`, `https://`, `data://`, `//`, `/`, or a relative URL.
- When uploading an image to a rich text area, you must enter a URL that is valid and well-formed.

Reports

- Only the first 254 characters in a rich text area or a long text area are supported with the “contains” operator in a report filter.
- You can't use a filter on a rich text area for locales and languages that use multibyte characters.
- The first 999 characters in a standard rich text area or a long text area are displayed in a report. For custom fields, only the first 254 characters are displayed. If you download the report as Details Only, the entire field is available.

Classic Encryption for Custom Fields

Restrict other Salesforce users from seeing custom text fields that you want to keep private. Only users with the View Encrypted Data permission can see data in encrypted custom text fields.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, Unlimited, and Database.com** Editions

Before you begin working with encrypted custom fields, review these implementation notes, restrictions, and best practices.

 **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Implementation Notes

- Encrypted fields are encrypted with a 128-bit data encryption key and use the Advanced Encryption Standard (AES) algorithm. You can archive, delete, and import your data encryption key. To enable encryption key management, contact Salesforce.
- You can use encrypted fields in email templates, but the value is always masked regardless of whether you have the View Encrypted Data permission.
- If you have the View Encrypted Data permission and you grant login access to another user, the user can see encrypted fields in plain text.
- Only users with the View Encrypted Data permission can clone the value of an encrypted field when cloning that record.
- Only the `<apex:outputField>` component supports presenting encrypted fields in Visualforce pages.
- When you use Visualforce email templates or call Visualforce pages with `getContent` or `getContentAsPDF` requests, encrypted field values are always masked regardless of whether you have the View Encrypted Data permission. Masking is present during Apex execution and on the resulting Visualforce markup.

Restrictions

Encrypted Text Fields:

- Can't be unique, have an external ID, or have default values.
- Aren't available for mapping leads to other objects.
- Are limited to 175 characters because of the encryption algorithm.
- Aren't available for use in filters such as list views, reports, roll-up summary fields, and rule filters.
- Can't be used to define report criteria, but they can be included in report results.
- Aren't searchable, but they can be included in search results.
- Aren't available for Connect Offline, Salesforce for Outlook, lead conversion, the Assignment element in a flow, workflow rule criteria or formulas, formula fields, outbound messages, default values, and Web-to-Lead and Web-to-Case forms.
- Aren't available to users with these License types: Partner, PowerPartner, CspLitePortal, CustomerSuccess, or PowerCustomerSuccess. Even if users have the View Encrypted Data permission, encrypted text fields aren't available.

Encrypted Data Files:

- Aren't available for date and time fields.

Best Practices

- Encrypted fields are editable regardless of whether the user has the View Encrypted Data permission. Use validation rules, field-level security settings, or page layout settings to prevent users from editing encrypted fields.
- You can still validate the values of encrypted fields by using validation rules or Apex. Both work regardless of whether the user has the View Encrypted Data permission.
- To view encrypted data unmasked in the debug log, the user must also have the View Encrypted Data permission in the service that Apex requests originate from. These requests can include Apex Web services, triggers, workflows, inline Visualforce pages (a page embedded in a page layout), and Visualforce email templates.
- Existing custom fields can't be converted into encrypted fields nor can encrypted fields be converted into another data type. To encrypt the values of an existing (unencrypted) field, export the data, create an encrypted custom field to store that data, and import that data into the new encrypted field.
- Mask Type isn't an input mask that ensures the data matches the Mask Type. Use validation rules to ensure that the data entered matches the mask type selected.
- Use encrypted custom fields only when government regulations require it because they involve more processing and have search-related limitations.

 **Note** This page is about Classic Encryption, not Shield Platform Encryption. [What's the difference?](#)

See Also

[Create a Custom Field](#)

Add or Edit Picklist Values

Add new values or rename existing values in a custom picklist from the fields area of an object. If the picklist uses a global picklist value set, you can change its values only by editing the global value set. Your changes affect all picklists that inherit their values from that global value set.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To change picklists:

Customize Application

 **Note** You can view changes to picklist values in Setup Audit Trail. To see the audit history: From Setup, in the Quick Find box, enter *View Setup Audit Trail*, and then select **View Setup Audit Trail**. To modify Campaign Member Status picklist values, see the list of considerations at the end of

this topic.

1. For Lightning Experience:
 - a. From Setup, click the Object Manager tab. If you don't see the Object Manager tab, in the Quick Find box, enter *Objects*, and then select **Objects**.
 - b. Click the object name.
 - c. Click **Fields & Relationships**.
- For Salesforce Classic:
 - a. From Setup, in the Quick Find box, enter the name of the object, and then select the object from the menu.
 - b. On the object page, select **Add a custom field in *objectname***.
2. Click the name of the picklist field to update.
3. In the Values section, next to a value, click **Edit**.

Action	Values	API Name	Default	Chart Colors	Modified By
<input type="checkbox"/> Edit Del Deactivate	Africa	Africa	<input type="checkbox"/>	Assigned dynamically	Admin User, 7/25/2023, 10:08 AM
<input type="checkbox"/> Edit Del Deactivate	Asia	Asia	<input type="checkbox"/>	Assigned dynamically	Admin User, 7/25/2023, 10:08 AM
<input type="checkbox"/> Edit Del Deactivate	Caribbean	Caribbean	<input type="checkbox"/>	Assigned dynamically	Admin User, 7/25/2023, 10:08 AM
<input type="checkbox"/> Edit Del Deactivate	Central America	Central_America	<input type="checkbox"/>	Assigned dynamically	Admin User, 7/25/2023, 10:09 AM
<input type="checkbox"/> Edit Del Deactivate	Europe	Europe	<input type="checkbox"/>	Assigned dynamically	Admin User, 7/25/2023, 10:09 AM
<input type="checkbox"/> Edit Del Deactivate	North America	North_America	<input type="checkbox"/>	Assigned dynamically	Admin User, 7/25/2023, 10:09 AM
<input type="checkbox"/> Edit Del Deactivate	Oceania	Oceania	<input type="checkbox"/>	Assigned dynamically	Admin User, 7/25/2023, 10:09 AM
<input type="checkbox"/> Edit Del Deactivate	South America	South America	<input type="checkbox"/>	Assigned dynamically	Admin User, 7/25/2023, 10:10 AM

4. Change the value's name, and optionally make the value the default for the master picklist.
5. Assign a color to use in charts by clicking and then select how to assign colors to values.
 - **Assign fixed colors to all values** assigns a fixed color to each value from the standard set of chart colors. The Chart Colors column shows the assigned colors. For example, if you want Closed Lost values always to be red in charts grouped by Opportunity Stage, assign red to that picklist value.
 - **Assign colors to values dynamically** assigns colors when you generate a chart. For example, manually assign colors only to the picklist values that need a fixed color. The remaining values are dynamically assigned.

Chart colors aren't available for multi-select picklists, currency picklists, or Task Subject, Event Subject, Knowledge Validation Status, and Opportunity Competitor picklists.
6. Click **Save**. New or edited picklist values don't sort automatically. Reorder the picklist values yourself.
7. To require a picklist, click **Edit** at the top of the detail page, and then select **Always require a value in this field in order to save a record**.
8. To change the picklist from unrestricted to restricted or vice versa, adjust the **Restrict picklist to the values defined in the value set** setting.
9. To open an easy-to-print list of your picklist values, click **Printable View**.

-  **Note** Not seeing your new or modified picklist values on records? See [Troubleshoot Missing Picklist Values](#).

Keep these considerations in mind when you add or edit picklist values.

- If you use record types, see [Considerations for Creating and Updating Record Types and Picklists](#).
- For the Campaign Member Status picklist, edit values directly from campaign records rather than in Setup. See [Campaign Member Statuses](#).
- For Ideas, setting the default value of the Categories or Status picklists doesn't affect the default value on the Ideas pages.
- Renaming a picklist value removes it from any associated filter criteria. For example, if your report contains a filter where *Lead Source equals Email or Web* and you change the picklist value *Web* to *Referral*, your report filter changes to *Lead Source equals Email*. If the changed value was the only filter criterion, the filter remains but displays an error.
- If dashboard widget grouping relies on a picklist field that includes a fixed color value, dashboard themes don't apply to the widget. Picklist values with fixed colors retain their assigned colors, while values with dynamically assigned colors use the default color palette.
- If your org uses the Translation Workbench and you change picklist values, let your translators know about the impact to translated values.

See Also

- [Picklist Limitations](#)
- [Dependent Picklists](#)
- [Define Default Field Values](#)
- [Manage Inactive Picklist Values](#)
- [Replace a Picklist Value](#)
- [Deactivate, Reactivate, or Remove a Single Picklist Value](#)
- [Delete, Deactivate, Replace, or Activate Multiple Picklist Values](#)
- [Remove a Picklist Value](#)
- [Considerations for Creating and Updating Record Types and Picklists](#)
- [Protect Picklist API Names for Formulas and Integrations](#)

Picklist Limitations

The maximum number of characters that you can have in a picklist depends on the type of picklist. Each picklist value includes a line break and a return character that aren't visible but that count in the character limit.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

Standard Picklists aren't available in **Database.com**

USER PERMISSIONS NEEDED

To change picklists:	Customize Application
----------------------	-----------------------

When you select picklist values for a list view filter, the combined size of the selected picklist values must be fewer than 240 characters.

Other limits apply to standard and custom picklists.

Other Limits for Standard Picklists

For standard picklists, each value can have up to 255 characters, not including line breaks and returns. This limit applies to single-select and multi-select picklists.

Picklist Field	Maximum Number of Values
Lead Status	100
Task Status	100
Task Priority	50
Case Status	100
Case Priority	50
Opportunity Stage	100

Limits for Custom Picklists

Inactive and active picklist value limits:

- Restricted picklists have a combined active and inactive limit of 1,000 values.
- Unrestricted picklists have a limit of 1,000 active values, and bound unrestricted picklists have a limit of 4,000 inactive values.
- All newly created unrestricted picklists are bound to the 4,000 inactive value limit by default.

Custom single-select picklist limits:

- Up to 1,000 values.
- Up to 255 characters per value.

Custom multi-select picklist limits:

- Up to 500 values.
- Up to 255 characters per value.
- Users can select up to 100 values at a time on a record.

Global picklist value set limits:

- Global picklist value sets have a combined active and inactive value limit of 1,000.
- You can have up to 500 picklist global value sets in an org.
- There's no limit on the number of custom picklists that use global picklist value sets.
- If you apply a global picklist value set to more than 13 different objects, you can deactivate values from the picklist value set, but you can't replace any picklist values or delete values from the set.

Functional Limitations for Custom Picklists

- You can make a custom single-select picklist field into a restricted picklist only if the picklist has fewer than 1,000 values (or entries). You can make a custom multi-select picklist field into a restricted picklist only if the picklist has fewer than 500 values (or entries). A restricted picklist's values are limited to only those values defined by a Salesforce admin, which prevents users from loading redundant or erroneous values through the API.
- Global picklist value sets are always restricted picklists, which preserves data integrity. Global value sets are shared across objects. Reuse the value set for any custom picklist field.
- For custom picklist fields that use a global picklist value set, you can change from a single-select to multi-select picklist and vice versa. But you can't change the picklist to a different field type, such as checkbox, currency, or text.
- You can't undo a custom picklist field's association with a global value set. To use a different global value set or different individual values in a picklist, delete the custom picklist field and replace it with a new picklist.

Functional Limitations for Multi-Select Custom Picklists

Multi-select picklists have implications when you work with that picklist data in other areas of Salesforce. If possible, use dependent picklists or another field type.

- Data—Data is stored in the object as a semicolon-separated list of values. For example, if you have a Regions multi-select picklist, the selected values appear like this in the UI:
Midwest;Southwest;Northwest;West. The way the values are stored can pose challenges when you want to update data via the API or import data. For example, let's say some distributors now handle the Northeast region, and you want to update those distributors via data import. If the data that you import only includes the Northeast region, then that's the only region that's updated in the records. Instead, your import data must include the regions that the distributor currently supports in addition to the new region.

When you use the UI to add a multi-select picklist value, it validates that the value doesn't exist. When you use the API to add a multi-select picklist value, the system doesn't perform this validation.

Therefore, the same picklist value can be stored multiple times in a multi-select picklist if that value is added from both the UI and from the API. If you add picklist values via the API, be sure to add logic to check for duplicate values in the picklist.

- Automation and Integration—To use multi-select picklist values in an Apex trigger, you must write code to parse the semicolon-delimited string. In a flow, you can't update a multi-select picklist by using the field update element. However, you can use a multi-select picklist as part of a flow, but similar to Apex triggers, you must parse the values from the semicolon-delimited string of values.
- Formulas—Only certain formula fields support multi-select picklists, but workarounds are available. For

- example, to copy a multi-select picklist field value from one record to another, use the INCLUDES function. See [Tips for Working with Picklist and Multi-Select Picklist Formula Fields](#).
- Reports—When you include a multi-select picklist field in a report, the data appears as a semicolon-delimited string like it does in the UI. Using picklist values to group data in a report also has some limitations. For example, let's say you want to group by region and list all the distributors in each region. If you group on the Region field, the groups would look something like:

```
Midwest;Southwest;Northwest;West  
Northeast;Mid-Atlantic  
West
```

View Active and Inactive Picklist Values

You can view the number of active and inactive picklist values and the maximum number allowed from the picklist field detail page.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: all editions

USER PERMISSIONS NEEDED

To create or change custom fields: Customize Application

1. Navigate to the picklist.
 - For a picklist on an object, go to the fields section of the object.
 - For a global picklist, from **Setup**, in the Quick Find box, enter *picklist*, and then select **Picklist Value Sets**.
2. To go to the picklist's detail page, click the picklist name.

The Picklist Values Used section shows the active and inactive number of picklist values and the maximum number allowed.

Keep these picklist value limits in mind.

- Restricted picklists have a combined active and inactive limit of 1,000.
- Bound unrestricted picklists have a limit of 1,000 active picklist values and a limit of 4,000 inactive picklist values.
- Global picklist value sets have a combined active and inactive limit of 1,000.
- Existing unbound picklists have no limit. By default, all newly created picklists are bound to the 4,000 inactive limit.

See Also

[Deactivate, Reactivate, or Remove a Single Picklist Value](#)

[Delete, Deactivate, Replace, or Activate Multiple Picklist Values](#)

Replace a Picklist Value

As your business changes, replace picklist values with more relevant alternatives. Replacing a value globally replaces that field value on all existing records.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

USER PERMISSIONS NEEDED

To replace picklist values: Customize Application

For example, let's say you have a status picklist with three different closed values, Closed-red, Closed-yellow, and Closed-green, and you want to simplify those three values to just one. Replace them with the new value Closed.

Important If you replace a parent value in a controlling picklist, the picklist dependency on that value is lost. After replacing the parent value, [re-create the dependency](#) using the new parent value.

1. If necessary, create the replacement value in the picklist edit page. See [Add or Edit Picklist Values](#).
2. Navigate to the picklist.
 - For a global picklist value set: From Setup, enter *picklist* in the **Quick Find** box, then select **Picklist Value Sets**.
 - For a picklist on an object, go to the fields area of the object. For example, for an Account picklist: From Setup, enter *Account* in the **Quick Find** box, then select **Fields** under Accounts.
3. Replace the picklist value.
 - For a global picklist value set: Go to the Global Value Set Detail page by clicking the picklist name. In the Values section, click **Replace**.
 - For all other picklists: Click **Replace** next to the picklist name.
4. Enter the value you want to change, and select a new replacement value.

Replacing an existing picklist value also changes the **Modified By** date and time for the record.

5. To use the new value in records where this field is empty, select **Replace all blank values**.
6. To update all occurrences of the value in your org's records with the new value, click **Replace**. Occurrences in the Recycle Bin are also updated.

The replace job is queued. To check the job's status, from Setup, enter *Background Jobs* in the **Quick Find** box, then select **Background Jobs**. You receive an email when the job is complete. If you replace the Stage picklist for opportunities, the **Probability**, **Forecast Category**, and **Expected Revenue** fields are also updated with the corresponding values.

See Also

[Create a Global Picklist Value Set](#)

- [Protect Picklist API Names for Formulas and Integrations](#)
- [Manage Inactive Picklist Values](#)

Deactivate, Reactivate, or Remove a Single Picklist Value

Deactivate, reactivate, or remove a value from a restricted or unrestricted custom picklist. If it's a global picklist value set, these actions simultaneously update all custom picklists that inherit its value set.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To create or change custom fields: Customize Application

Only a Salesforce admin can modify the values of a restricted picklist. Users can't add unapproved values, even through the API.

1. Go to the picklist you want to modify.
 - For a picklist on an object, go to the fields section of the object.
 - For a global picklist, from **Setup**, in the Quick Find box, enter *picklist*, then select **Picklist Value Sets**.
2. Go to the picklist's detail page by clicking the picklist's name.
3. In the Values section, modify the picklist value. These actions simultaneously update all custom picklists that inherit its value set.
 - To deactivate a value, which removes it from the picklist but keeps it on existing records, click **Deactivate** next to the value's name. The value moves to the Inactive Values section. If you need the value again later, click **Activate** next to its name.
 - To remove a value from the picklist and all records that use it, click **Del** next to the value's name, and select **Replace value on records with blank value**. Then save the changes.
 - To remove a value from a global picklist and any custom picklists that share its value set, click **Del** next to the value's name, and then click **OK**. If any custom picklists use this global picklist value set, you're prompted to replace the value with a blank value or an existing, active value. If no custom picklists use the value set, the value is deleted from the global picklist.

Keep these considerations in mind when you deactivate, reactivate, or delete a picklist value.

- The unrestricted picklist values for the `Task.Subject` and `Event.Subject` fields can't be deactivated. The picklist values are deleted instead of deactivated.
- If a global picklist value is used in historical trending, it's deactivated but not deleted.
- Deleting a value in a global picklist value set or on non-global, restricted picklists goes to the background jobs queue. When the job completes, your picklist is updated and you're notified by email.

- Changes to picklist values are logged in Setup Audit Trail. To view the audit history, from Setup, in the Quick Find box, enter *View Setup Audit Trail*, then select **View Setup Audit Trail**.
- Using the option to replace a picklist value while deleting the current value doesn't trigger workflow rules.

Delete, Deactivate, Replace, or Activate Multiple Picklist Values

Save time by managing all your picklist values at once.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: all editions

USER PERMISSIONS NEEDED

To create or change custom fields: Customize Application

Follow these steps to manage multiple picklist values.

1. Navigate to the custom field definition page for the picklist field.
2. In the Values and Inactive Values sections, there's a checkbox next to each picklist value. Select multiple values, and use one of the new buttons: **Delete Selected**, **Deactivate Selected**, **Replace Selected**, or **Activate Selected**.

This feature is available only for custom picklists with predefined values.

See Also

[Replace a Picklist Value](#)

[Remove a Picklist Value](#)

Sort Picklists

You can arrange picklist values in a specific order or sort them alphabetically.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To sort picklists: Customize Application

1. From the management settings for the picklist field's object, go to Fields.

- For custom task or event fields, go to the object management settings for Activities.
 - For standard task fields, go to the object management settings for Tasks. For standard event fields, go to the object management settings for Events.
 - For Knowledge validation status picklists, from Setup, in the Quick Find box, enter *Validation Statuses*, and then select **Validation Statuses**.
2. Click the name of the picklist to update.
 3. Click **Reorder**.
 4. Use the arrows to arrange the field in the proper sequence.
 5. Select a default value if desired.
 6. To alphabetize the entries for users on edit pages, check **Sort values alphabetically....**
 7. Save your changes.

On record edit and detail pages and in reports, picklist and multi-select picklist fields can include inactive values. These inactive values are sorted last, unless you choose alphabetical sorting. In that case, all values are sorted alphabetically.

 **Note** For English locale users, hyphens and spaces in picklist values are ignored when picklists are sorted alphabetically. To view values in a different order, use manual sorting.

See Also

[Find Object Management Settings](#)

Remove a Picklist Value

Remove or replace custom picklist values to update the records that have the picklist field.

You have several ways to remove existing picklist field values. For picklist values that are part of a global picklist value set, the steps are a little different. When you remove a picklist value, you can also replace the value in records with another value. For more information on replacing an existing value, see [Replace a Picklist Value](#). These steps remove existing values.

1. Go to the picklist.
 - For a global picklist value set: From Setup, in the Quick Find box, enter *picklist*, and then select **Picklist Value Sets**.
 - For a picklist on an object, go to the object's fields area. For example, for an Account picklist: From Setup, in the Quick Find box, enter *Account*, and then under Accounts, select **Fields**.
2. Click the picklist name.
3. To remove a value from the picklist, click **Del** next to the value's name.

Decide whether to replace the value or leave it blank. If you replace the value with a blank value, existing records don't display the value. To keep the value on existing records, select **Deactivate** instead of **Del**.

A picklist must have at least one active value. If you have one value in a picklist, there's no option to delete that value.

Keep these considerations in mind when you replace a picklist value.

- For orgs using record types, include some or all values from the master picklist in different record types to offer a subset of values to users based on their profile.
- Some **special picklists**, such as opportunity **Stage**, **Task Priority**, **Task Status**, **Lead Status**, and **Case Status**, prompt you to map the deleted value to another existing value in all your org's records. You can map the values or leave your existing data unchanged.
- Using the option to replace a picklist value while deleting the current value doesn't trigger workflow rules.

See Also

[Replace a Picklist Value](#)

Picklists with Additional Information

These standard picklist fields have additional information that you can edit.

Picklist	Description
Partner Role (for accounts)	<p>Roles of account partners, for example, Consultant, Supplier. These options are available when you add an account to the Partners related list of an account or opportunity.</p> <p>To edit, from Setup, enter <i>Partner Roles</i> in the Quick Find box, then select Partner Roles.</p> <p>Enter the name of the partner role in the “Role” column. In the “Reverse Role” column, enter the corresponding reverse partner role. Assigning a partner role to an account creates a reverse partner relationship so that both accounts list the other as a partner.</p> <p>Each role and reverse role value can have up to 40 characters.</p>
Priority (for cases)	<p>Urgency of case, for example, Low, High.</p> <p>If you delete a value, you have the option to map the deleted value to another existing value in all your org's cases.</p> <p>Each picklist value can have up to 40 characters.</p>
Status (for campaign members)	<p>State of a campaign member, for example, Sent or</p>

Picklist	Description
	<p>Responded.</p> <p>If you delete a Status value, you have the option to map the deleted value to another existing value. The new replacement value is automatically added to the member status for campaigns that contained the deleted value.</p> <p>If the deleted value is the default member status for a campaign, the new replacement value becomes the default status for that campaign.</p>
Status (for cases)	<p>State of the case, for example, New, On Hold.</p> <p>If you delete a value, you have the option to map the deleted value to another existing value in all your org's cases.</p> <p>Each picklist value can have up to 40 characters.</p>
Status (for contracts)	<p>State of the contract in the contract business process. You can add values to this picklist and organize each value into one of several categories, for example, "Draft", "In Approval Process", or "Activated". Then sort your contracts using these categories in reports and views.</p>
Contact Role (for contracts)	<p>Role of a contact on a contract, for example, Business User, Decision Maker. These options are available when you add a contact to the Contact Roles related list of a contract.</p> <p>To edit, from Setup, enter <i>Contact Roles</i> in the Quick Find box, then select Contact Roles on Contracts.</p> <p>Each picklist value can have up to 40 characters.</p>
Lead Status (for leads)	<p>State of the lead, for example, Open, Qualified.</p> <p>Select one value as the "Default Status" assigned to all new leads created manually, via imports, or via your website. Select one or more values as the</p>

Picklist	Description
	<p>“Converted Status” assigned to converted leads. When you convert qualified leads into an account, contact, and opportunity, you can select one of the “Converted” statuses to assign to the lead. Leads with a “Converted” status type are no longer available in the Leads tab, although you can include them in reports.</p> <p>If you delete a value, you have the option to map the deleted value to another existing value in all your org’s leads.</p> <p>Each value can have up to 20 characters.</p>
Contact Role (for opportunities)	<p>Role of a contact for an opportunity, for example, Business User, Decision Maker. These options are available when you add a contact to the Contact Roles related list of an opportunity.</p> <p>To edit, from Setup, enter <i>Contact Roles</i> in the Quick Find box, then select Contact Roles on Opportunities.</p> <p>Each picklist value can have up to 40 characters.</p>
Stage (for opportunities)	<p>Sales process stages, for example, Prospect, Proposal. This picklist also affects the Type and Forecast Category values of an opportunity. Specifically, changing the Type or Forecast Category for a Stage picklist value updates all opportunities that have that stage value.</p> <p>To edit, from the management settings for opportunities, go to Fields, and then click Edit next to Stage.</p> <p>To deactivate an active stage, click Del next to the stage. On the mapping page, don’t replace the stage with another existing value; just click Save. The stage now appears in the Inactive Stage Picklist Values related list. The stage is no longer in use but can exist in older opportunity records.</p>
Status (for solutions)	Status of a solution, for example, Draft, Reviewed.

Picklist	Description
	<p>Mark one or more values as “Reviewed”. When users solve cases using solutions, they can view which solutions have been reviewed and which haven’t. Each picklist value can have up to 40 characters.</p>
Priority (for tasks)	<p>Importance of the task, for example, High, Normal, Low. Set one value as the default priority of all new tasks, and one value as the highest priority.</p> <p>If you delete a value, you have the option to map the deleted value to another existing value in all your org’s tasks.</p> <p>Each picklist value can have up to 20 characters.</p>
Status (for tasks)	<p>State of a task, for example, Not Started, Completed. Choose at least one value as the “Closed” status and one value as the “Default” status for all new tasks.</p> <p>If you delete a value, you have the option to map the deleted value to another existing value in all your org’s tasks.</p> <p>Each picklist value can have up to 40 characters.</p>
Task Type (for tasks)	<p>Send Email Default specifies the default task type assigned when the task is sending email or mass email, and when tasks are created via inbound email, such as Email to Salesforce. Default specifies the default picklist value when creating tasks.</p> <p>To edit, from the management settings for tasks, go to Fields, and then click Edit next to the picklist value that you want to specify as the default.</p>

See Also

[Picklist Limitations](#)

Manage Inactive Picklist Values

Establish upper bound limits, deactivate, reactivate, or remove a value from a restricted or unrestricted picklist.

REQUIRED EDITIONS

Inactive picklist values can be a result of an import or intentionally removing the value from the list of available values for future use. In certain cases, importing text fields can create numerous unwanted picklist fields. Managing your inactive picklist values and enforcing limits on inactive values for unrestricted picklists can improve performance and improve your Salesforce org's overall health.

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

Get a List of Custom Picklist Fields with Inactive Values

Get email notifications when custom picklist fields have more than 4,000 inactive values and then use this information to bulk delete the inactive unused values. This feature is available only for custom picklists with predefined values.

Bulk Delete Inactive Picklist Values

Managing your inactive picklist values and enforcing limits on the number of inactive values for custom picklists can improve performance and the overall health of your Salesforce org. If you have a picklist with a large number of inactive unused values, you can bulk delete them.

Establish a Picklist Upper Bound Limit

You can enforce an upper bound limit of inactive values on existing picklists. The limit is enforced on the inactive values of unrestricted, custom picklists whose numbers don't exceed the allowable limit. By default, all new unrestricted picklists have an enforced limit of 4,000 inactive values.

Convert a Text Field to a Picklist Field

By default, an upper bound limit of 4,000 is applied to inactive unrestricted picklists for each field. If you exceed this limit when converting a text field to a picklist field, you get an error message.

Get a List of Custom Picklist Fields with Inactive Values

Get email notifications when custom picklist fields have more than 4,000 inactive values and then use this information to bulk delete the inactive unused values. This feature is available only for custom picklists with predefined values.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To create or change custom fields:

[View Setup and Configuration](#)

Follow these steps to get an email notification for custom picklists that have more than 4,000 values.

1. From Setup, in the Quick Find box, enter *Settings*, and then select **Picklist Settings**.
2. In the Picklist Settings window, click **Email Me** to receive an email with the picklist fields and the number of inactive values.

Get a list of custom picklist fields with inactive values

Get an email that tells you which custom picklist fields have more than 4,000 inactive values. Use this information to delete inactive unused values in the identified picklists. When you remove inactive values, it helps to ensure the stability and performance of your picklist fields.

[Email Me](#)



In the email, click the picklist name to go to the picklist field page and delete any inactive unused values.

See Also

[Bulk Delete Inactive Picklist Values](#)

[Picklist Limitations](#)

Bulk Delete Inactive Picklist Values

Managing your inactive picklist values and enforcing limits on the number of inactive values for custom picklists can improve performance and the overall health of your Salesforce org. If you have a picklist with a large number of inactive unused values, you can bulk delete them.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: all editions

USER PERMISSIONS NEEDED

To create or change custom fields:

[Customize Application](#)

These rules apply when bulk deleting picklist values.

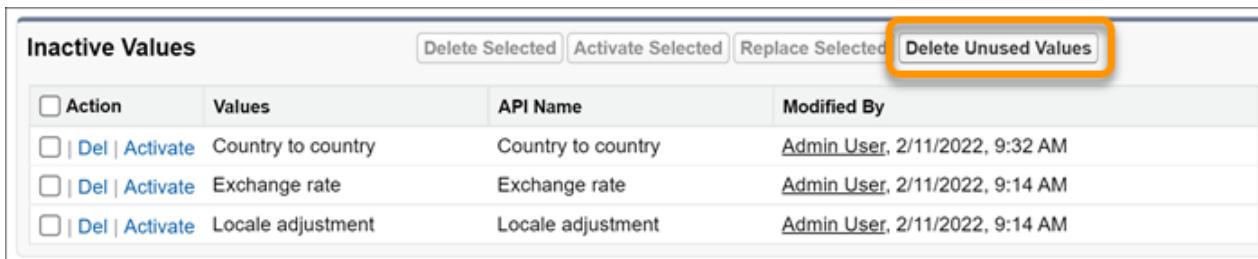
- This feature is available only for custom picklist fields on an object. It's not available for standard or multi-select picklist fields, or for picklists on external objects.
- This feature is only for picklist fields that have defined values. It's not available for picklist fields that use a global value set.
- This feature deletes inactive, unused picklist values. If a picklist value is inactive but still referenced in a record, the value isn't deleted.
- After you delete unused values, if a picklist is unbound and the total number of inactive values falls

below the limit, the field is set to a bound picklist.

- While the job that deletes inactive values runs, these actions are unavailable on that picklist field: Activate, Activate Selected, Deactivate, Deactivate Selected, Delete, Delete Selected, Replace, and Replace Selected. The buttons are enabled, but an error message is returned when you click them.
- While the job that deletes inactive values runs, changes to the picklist field via Metadata API are unavailable. You can retrieve data from the field, but you can't create a picklist value, update the field or its values, delete the field or its values, or deploy the field.

Follow these steps to delete all inactive unused values from a picklist.

1. From Setup, in the Quick Find box, enter *Object Manager*, and then select your object.
2. Click **Fields & Relationships**, and then click the picklist field. In the Picklist Values Used section, you can see how many inactive picklist values the field has.
3. From the Inactive Values section of the picklist field, click **Delete Unused Values**.
4. Click **OK**. The job that deletes the inactive values runs in the background and the button becomes disabled.



Inactive Values			
Action	Values	API Name	Modified By
<input type="checkbox"/> Del Activate	Country to country	Country to country	Admin User, 2/11/2022, 9:32 AM
<input type="checkbox"/> Del Activate	Exchange rate	Exchange rate	Admin User, 2/11/2022, 9:14 AM
<input type="checkbox"/> Del Activate	Locale adjustment	Locale adjustment	Admin User, 2/11/2022, 9:14 AM

The job that deletes inactive values runs in the background. It can take some time to finish depending on the number of inactive values. A record of the change is added in the Setup Audit Trail.

To see the status of the job tasks, use the Background Jobs page. Deleting inactive picklist values involves three job tasks.

- Unused Picklist Values Delete Task (Final Delete)
- Unused Picklist Values Delete Task (Filter)
- Unused Picklist Values Delete Task (Prepass)

The job has completed successfully when the status of Unused Picklist Values Delete Task (Final Delete) is Completed.

After all the picklist values are deleted, you receive an email that tells you whether the deletion was successful or not. The email is sent to the email address associated with the user who initiated the deletion.

If you receive a success email and still see inactive picklist values in the Inactive Values section, those picklist values likely are referenced in a record. You can delete only picklist values that aren't used in any records. To identify which records use the inactive picklist values that you want to delete, you can run a SOQL query or create a report.

See Also

[Get a List of Custom Picklist Fields with Inactive Values](#)

Picklist Limitations

Establish a Picklist Upper Bound Limit

You can enforce an upper bound limit of inactive values on existing picklists. The limit is enforced on the inactive values of unrestricted, custom picklists whose numbers don't exceed the allowable limit. By default, all new unrestricted picklists have an enforced limit of 4,000 inactive values.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To create or change custom fields: Customize Application

1. From **Setup**, in the Quick Find box, enter *Picklist Settings*, and then select **Picklist Settings**.
2. On the Picklist Settings page, click **Establish upper bound on existing picklists**.

A query runs to identify picklists that have fewer than 4,000 inactive values. The limit is then enforced only on those picklists. A message appears to show the number of picklists that were bound to the limit, or that there are no unbound picklists that meet the limit.

You can run the upper bound on existing picklists multiple times. For example, if you delete picklist values, you can potentially reduce the total number of values to be less than the limit. Admins or Salesforce can't configure the limit of 4,000 inactive values.

See Also

[Convert a Text Field to a Picklist Field](#)

Convert a Text Field to a Picklist Field

By default, an upper bound limit of 4,000 is applied to inactive unrestricted picklists for each field. If you exceed this limit when converting a text field to a picklist field, you get an error message.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

1. From the management settings for the field's object, go to Fields.
2. Click **Edit** next to the custom field that you want to change.
3. Click **Change Field Type**.
4. Select a **Picklist** and click **Next**.

5. Enter a field label, name, and any other attributes, and then save your changes.

Keep these considerations in mind when you convert a field to a picklist.

- The limit isn't case-sensitive.
- If there are more than 4,000 unique values, the conversion fails. This conversion limit includes active picklists.
- Soft-deleted values count toward the 4,000 unique value limit. To avoid values that count toward the limit, permanently delete any soft deleted records and field values by emptying the recycle bin.
- Keep this setting: Remove upper bound on inactive picklist values.
- You can't convert a defined unique text field to a picklist or a multi-select picklist.

See Also

[Change the Data Type of a Custom Field](#)

[Establish a Picklist Upper Bound Limit](#)

[Considerations for Converting the Field Type of a Custom Field](#)

Troubleshoot Missing Picklist Values

If you add a new picklist value but it doesn't appear when users view or edit records, there are several configuration issues to consider. Review the most common causes and solutions for this type of problem. This topic covers inactive values, record type or business process assignments, field dependencies, and field-level security.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

Verify the Picklist Value Is Active

Picklist values can be active or inactive. Inactive values remain visible on existing records and in reports. However, they don't appear in the picklist list when users create records or update existing ones. If you added a value in an inactive state or accidentally deactivated a value, users don't see it as a selectable option.

To check if the picklist value you added is inactive, see [View Active and Inactive Picklist Values](#) in Salesforce Help.

Add the Picklist Value to All Relevant Record Types and Business Processes

When you add a picklist value to a picklist, it doesn't automatically appear in existing record types. Each record type maintains its own list of available values, independent of the picklist associated with the record type's object. If you don't add the new value to each record type, users assigned to those record

types can't see it.

-  **Note** This behavior also applies to business processes for Leads, Sales, Support, and so forth. For missing Case Status, Lead Status, or Opportunity Stage values, add them to the object's Processes page in Setup.

To confirm that your new picklist value is available for all appropriate record types, see [Edit Picklists for Record Types and Business Processes](#) in Salesforce Help.

Update Field Dependencies for Dependent Picklists

For dependent picklists, verify that the new picklist value exists in the field dependency matrix for the appropriate controlling field. If you don't create these associations, the new value doesn't appear when users select certain controlling field values.

To make sure that these associations are in place, see [Edit Dependent Picklists](#) and [Use the Field Dependency Matrix](#) in Salesforce Help.

Verify Field-Level Security Settings

Field-level security controls whether users can view, edit, or access a field. If a user's profile or permission set hides a picklist field via field-level security, the user can't see the picklist options.

To confirm users have access to the picklist, see [Verify Access for a Particular Field](#) and [Modify Field Access Settings](#) in Salesforce Help.

See Also

- [Considerations for Creating and Updating Record Types and Picklists](#)
[Knowledge Article: Newly Created Picklist Value Not Appearing on Record](#)

Protect Picklist API Names for Formulas and Integrations

When you create a picklist value, an API Name that matches the Label is assigned. By default, the API Name can be changed at any time, but you can choose to protect the API Name.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To customize picklist settings: [Customize Application](#)

A picklist value is identified by either the displayed value or the API Name. Formulas reference the API

Name so that the formula continues to work even if the displayed value changes. An admin can prevent changes to the API Name to protect the references to the fields in formulas or during integrations, such as data import.

1. From Setup, enter *picklist* in the Quick Find box, then select **Picklist Settings**.
 2. To prevent changes to the API names for any of the values, select **Disable editing picklist values' API names**.
- If you need to change an API Name later, you can deselect this option.

See Also

[Create a Custom Picklist Field](#)

[Replace a Picklist Value](#)

[Add or Edit Picklist Values](#)

Dependent Picklists

Use a dependent picklist to help your users enter accurate and consistent data. A *dependent picklist* is a custom or multi-select picklist for which the valid values depend on the value of another field, called the *controlling field*. A controlling field can be any picklist (with at least one and fewer than 300 values) or checkbox field on the same record.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

 **Example** For example, you can define a Reason custom picklist on opportunities and make its valid values depend on the value of the Stage picklist like this:

- If Stage is *Closed Won*, the valid values for Reason are `Superior features` or `Lower price`.
- If Stage is *Closed Lost*, the valid values for Reason are `Inferior features`, `Higher price`, or `Company viability`.

Define a Dependent Picklist

Specify a dependent picklist and its values for your org.

Use the Field Dependency Matrix

The field dependency matrix lets you specify the dependent picklist values that are available when a user selects each controlling field value. The top row of the matrix contains the controlling field values, while the columns list the dependent field values. Use this matrix to include or exclude values.

Included values are available in the dependent picklist when a value in the controlling field is selected.

Excluded fields aren't available in the dependent picklist for the selected controlling field value.

Edit Dependent Picklists

Modify field dependencies in picklists.

Delete Picklist Dependencies

If you no longer want the values of a dependent picklist to depend on a controlling field, delete its dependency. Deleting the dependency removes the logic that defines how the values of the picklist depend on the controlling field, but doesn't delete the fields or affect their data.

Dependent Picklist Considerations

When defining dependent picklists, review these considerations.

Define a Dependent Picklist

Specify a dependent picklist and its values for your org.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: All Editions

Standard Objects aren't available in **Database.com**

USER PERMISSIONS NEEDED

To define dependent picklists:

Customize Application

If your org uses record types, make sure that your controlling and dependent picklist values are available in the appropriate record types before you define a dependent picklist.

- From the management settings for the object you want to add a field to, go to Fields. Custom task and event fields are accessible from the object management settings for Activities.
 - Click Field Dependencies.**
 - Click New.**
 - Choose a controlling field and dependent field.
Some picklist and checkbox fields aren't available as controlling fields.
 - Click Continue.**
 - Use the [field dependency matrix](#) to specify the dependent picklist values that are available when a user selects each controlling field value.
 - Optionally, click **Preview** to test your selections. If your org uses record types, choose a record type to test how it affects your controlling and dependent picklist values. The record type controls what values are available in the controlling field. The record type and the controlling field together determine what values are available in the dependent picklist. For example, a dependent value is only available if it's available in the selected record type and the selected controlling value.



Note The **Filter by Record Type** option doesn't appear in the Preview window for activity custom fields.

- #### **8. Click Save.**

Use the Field Dependency Matrix

The field dependency matrix lets you specify the dependent picklist values that are available when a user selects each controlling field value. The top row of the matrix contains the controlling field values, while the columns list the dependent field values. Use this matrix to include or exclude values. Included values are available in the dependent picklist when a value in the controlling field is selected. Excluded fields aren't available in the dependent picklist for the selected controlling field value.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To define picklist dependencies: Customize Application

To include or exclude values:

1. Double-click values to include them. Included values are indicated with highlighting. Double-click any highlighted values to exclude them.
2. To select a range of adjacent values, click a value and use SHIFT+click another value. Then click **Include Values** to make the values available, or **Exclude Values** to remove them from the list of available values.
3. Click a value and use CTRL+click to select multiple values. Then click **Include Values** to make the values available, or **Exclude Values** to remove them from the list of available values.
4. To select all the values in that column, click a column header. Then click **Include Values** to make the values available, or **Exclude Values** to remove them from the list of available values.

You can change the values in your view.

- To view all available values at once, click **View All**.
- To view all the dependent values in that column, click **Go To** and choose a controlling value.
- To view the values in columns that are on the previous or next page, click **Previous** or **Next**.
- To view 5 columns at a time, click **View sets of 5**.

Edit Dependent Picklists

Modify field dependencies in picklists.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To edit field dependencies: Customize Application

1. From the management settings for the picklist's object, go to Fields.
2. Click **Field Dependencies**.
3. Click **Edit** next to the field dependency relationship you want to change.
4. Use the [field dependency matrix](#) to specify the dependent picklist values that are available when a user selects each controlling field value.
5. Optionally, click **Preview** to test your selections.
6. Save your changes.

See Also

[Find Object Management Settings](#)

Delete Picklist Dependencies

If you no longer want the values of a dependent picklist to depend on a controlling field, delete its dependency. Deleting the dependency removes the logic that defines how the values of the picklist depend on the controlling field, but doesn't delete the fields or affect their data.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To delete picklist dependencies: Customize Application

1. From the management settings for the picklist's object, go to Fields.
2. Click **Field Dependencies**.
3. Click **Del** next to the field dependency relationship you want to delete.
4. Click **OK** to confirm.

See Also

[Find Object Management Settings](#)

Dependent Picklist Considerations

When defining dependent picklists, review these considerations.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To define and edit dependent picklists: Customize Application

Checkboxes	Checkbox fields can be controlling fields but not dependent fields.
Connect Offline	While controlling fields and dependent picklists are available in Connect Offline, the logic between them isn't.
Converting fields	When you convert existing fields to dependent picklists or controlling fields, it doesn't affect the existing values in your records. After conversion, the dependency rules apply to new records and to any changes to existing records.
Default values	<ul style="list-style-type: none"> You can prepopulate a record event with default values in dependent picklists. Default values on dependent picklists are available on accounts, opportunities, contacts, leads, and custom objects. If the Automated Account Field setting is enabled, you can't prepopulate dependent picklists with default values.
Dependency limitations	<ul style="list-style-type: none"> Before defining a dependency, make sure that your picklist has at least one value. Standard fields such as Product Family don't contain values until you add them. If a standard controlling field relies on functionality that you disable, then the dependency rules for the picklist no longer apply. For example, if you disable the Self-Service portal, and Closed by Self-Service User is a controlling field, its dependent picklist shows all available values. If you replace a parent value in a controlling picklist, the picklist dependency is lost. After replacing the parent value, re-create the dependency by using the new parent value. You can update the field dependency of a managed controller picklist field only from the UI and not by using the Metadata API. Picklist dependencies are created based on the API names of individual picklist values. If these API names are modified after the dependency is configured, the dependent field relationship doesn't work as expected.
Field-level security	Field-level security settings for a controlling field and dependent picklist are independent. Remember to hide a controlling field whenever its correlating dependent picklist is hidden.

Import	The Data Import Wizard doesn't consider field dependencies. You can import any value into a dependent picklist regardless of the value imported for a controlling field.
Lead conversion	<ul style="list-style-type: none"> If you create a dependency for lead fields that map to account, contact, and opportunity fields for lead conversion, create the same dependency on the account, contact, or opportunity. You can map dependent picklists and controlling lead fields to account, contact, or opportunity fields with different dependency rules.
Lightning pages	Dynamic Forms-enabled Lightning pages that contain dependent picklists must also contain their controlling picklist fields, or the dependent picklist values don't appear.
Multi-select picklists	Multi-select picklists can be dependent picklists but not controlling fields.
Page layouts	<ul style="list-style-type: none"> Page layouts that contain dependent picklists must also contain their controlling picklist fields, or the dependent picklist values don't appear. Make sure that the dependent picklist is lower on the page layout than its controlling field. If a dependent picklist is required and no values are available for it based on the controlling field value, users can save the record without entering a value. The record is saved with no value for that field.
Picklist limitations	<ul style="list-style-type: none"> A controlling field can have up to 300 values. If a field is both a controlling field and dependent picklist, it can't contain more than 300 values. No checks are performed for dependent fields when a controlling field is updated. These fields aren't available as controlling fields. <ul style="list-style-type: none"> - Activity Fields <ul style="list-style-type: none"> - Call Type - Create recurring series of events - Show Time As - Subject - Task - Type - Contact Fields

	<ul style="list-style-type: none"> - Salutation - Contact Currency - Custom Object Fields - Currency - Lead Fields - Converted - Unread By Owner
Record types	You can define or change the record type for your dependent picklist only within the Preview dialog when creating or editing the field dependency values. The record type controls what values are available in the controlling field. The record type and the controlling field together determine what values are available in the dependent picklist. For example, a dependent value is only available if it's available in the selected record type and the selected controlling value.
Standard versus custom picklists	<ul style="list-style-type: none"> • Custom picklist fields can be either controlling or dependent fields. • Standard picklist fields can be controlling fields but not dependent fields.

Rich Text Editor

Use the rich text editor to format text in custom fields with the rich text area type. You can also use the rich text editor in many other features, such as Chatter publisher and groups.

Available in: the Salesforce mobile app, Salesforce Classic, and Lightning Experience

Available in: All Editions

You can perform the following operations with the rich text editor's WYSIWYG interface.

- Format text as bold, italicized, underlined, or strikethrough
- Create bullet and numbered lists
- Change paragraph indentation
- Insert a hyperlink
- Insert an image (copying inline images from external sources and pasting them into the editor isn't supported)
- Remove formatting

The availability of toolbar buttons varies across features. For example, the code snippet button is available in Chatter publisher but not in custom fields. Extra functions are supported for features such as

Salesforce Knowledge and Ideas, like the ability to embed multimedia content.

-  **Note** Beginning in Summer '17, custom fields provide a rich text editor that no longer uses CKEditor in Lightning Experience and the Salesforce mobile app. Rich text editors provide a WYSIWYG interface only; you can't edit HTML tags. When you copy content from a web page or another source and paste it into the editor, unsupported tags are removed. Text that was enclosed in unsupported tags is preserved as plain text. You aren't notified when unsupported or potentially malicious HTML tags are removed. You can expect minor formatting or styling differences when editing a rich text field across the interfaces. We recommend using the toolbar to fix formatting or styling differences. Lightning Knowledge continues to use the CKEditor for custom rich text fields. Knowledge article rich text fields provide additional functions, such as the ability to view and edit the source HTML, support for more HTML styles, and smart links between articles.

Some features have rich text editors across Salesforce Classic, Lightning Experience, and the Salesforce mobile app. For example, the Information field in groups provides a rich text editor in all three user interfaces. The Chatter Publisher has a rich text editor only in Salesforce Classic and Lightning Experience.

	Salesforce Classic	Lightning Experience	The Salesforce Mobile App
Groups	✓	✓	✓
Lightning Components (<code>ui:inputRichText</code>) and <code>lightning:inputRichText</code>)	✓	✓	✓
Custom Fields	✓	✓	✓
Chatter Publisher	✓	✓	
Knowledge Article	✓	✓	
Lightning App Builder (Rich Text Component)	✓	✓	
Experience Builder (Rich Content Editor)	✓	✓	
Flow Builder	✓	✓	
Idea Themes	✓		
Send Email Action		✓	
Sales Path		✓	

	Salesforce Classic	Lightning Experience	The Salesforce Mobile App
Notes		✓	✓

Rich Text Editor Considerations

When you use the editor, note the following.

- The maximum number of characters you can enter in the rich text editor is equal to the field length specified when creating or editing the field. The default is 32,768 characters, which include characters used by HTML tags that apply formatting.
- Clicked hyperlinks open in a new browser window. The rich text editor supports HTTP, HTTPS, and mailto hyperlinks.
- The rich text editor doesn't validate hyperlinks to web pages. Confirm that you're specifying a valid URL.
- To insert an image, click and either select:
 - **Web Address** and enter the URL of the image. This option isn't available in Lightning Experience.
 - **Upload Image** and select an image from your local host. You can upload only JPEG, PNG, or GIF files. The file can't exceed 1 MB. You can't add a hyperlink to an image. Copying inline images from external sources to the text editor isn't supported.

Optionally, enter a description that appears when you hover over the image. The image must have a URL that Salesforce can access.

- The rich text editor supports all languages that Salesforce Knowledge supports.
- The rich text editor doesn't support JavaScript or Cascading Style Sheets (CSS).
- The rich text editor is disabled for users who have accessibility mode enabled. It's replaced with a text field.

Supported Formatting

We recommend that you use the toolbar to format your content. When you copy content from a web page or another source and paste it into the editor, unsupported tags are removed. Text that was enclosed in unsupported tags is preserved as plain text. The rich text editor supports the tags listed in the table.

<a>	<dt>	<q>
<abbr>		<samp>
<acronym>		<small>
<address>	<h1>	
	<h2>	<strike>
<bdo>	<h3>	

<big>	<h4>	<sub>
<blockquote>	<h5>	<sup>
 	<h6>	<table>
<caption>	<hr>	<tbody>
<cite>	<i>	<td>
<code>		<tfoot>
<col>	<ins>	<th>
<colgroup>	<kbd>	<thead>
<dd>		<tr>
		<tt>
<dfn>	<p>	<u>
<div>	<pre>	
<dl>	<var>	

-  **Note** Rich text editors in custom rich text area fields have minor formatting differences in Lightning Experience and the Salesforce mobile app, as compared to Salesforce Classic. For more information, see [Editing Rich Text Area Fields in Records](#).

The tags can include the following attributes.

alt	face	size
background	height	src
border	href	style
class	name	target
colspan	rowspan	width

The following URL protocols are supported.

- http:
- https:
- file:
- ftp:
- mailto:
- #
- / for relative links

Rich Text Editors Using CKEditor

Rich text editors in Salesforce Classic continue to use CKEditor. Beginning in Summer '24, Salesforce uses CKEditor version 4.24.0.

In Salesforce Classic, CKEditor is used by:

- Chatter Publisher
- Custom Fields
- Email
- Flow Builder
- Groups
- Idea Themes
- Knowledge Article

In Lightning Experience and the Salesforce mobile app, the following features have rich text editors that use CKEditor.

- Email Composer, using the Send Email action
- Lightning Knowledge

These features load the rich text editor in an iframe. To load the editor correctly in an iframe, your browser must use appropriate security settings. Refer to your browser's instructions to configure the settings. For example:

- Internet Explorer—Make sure that the **Launching programs and files in an IFRA**ME security setting is set to **Enable or Prompt**.
- Safari—We recommend changing your Privacy settings for cookies to **Allow from websites I visit**.
- Chrome—Make sure that **Block third-party cookies** isn't selected in the Privacy and security settings.

See Also

[Editing Rich Text Area Fields in Records](#)

[Rich Text Fields in Knowledge Articles](#)

Add Videos to a Custom Field Using the HTML Editor

To embed a video, use the HTML editor to paste the element as an <iframe> from a video site.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create or change custom fields:

Customize Application

1. Copy the `<iframe>` element from the site where your video is hosted.
2. Paste your code into the HTML editor by clicking one of the buttons.

Button	Description
	Lets you paste the <code><iframe></code> element into a text box on the Embed Multimedia Content dialog box. The frame and its contents are added to the editor window.
	Lets you paste the <code><iframe></code> element directly into the HTML code.

3. Click **Save**.

Keep these considerations in mind when you embed a video.

- If you run into problems embedding videos, check your browser security settings. Some browsers block iframe elements.
- Embedded videos are removed from emails when you insert Knowledge articles into case emails.
- You can't send embedded videos via email.

See Also

[Rich Text Editor](#)

Define Default Field Values

Define a default value for a field. Use a formula to generate dynamic values or constants for static values.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To view default field values:	View Setup and Configuration
-------------------------------	------------------------------

To define or change default field values:	Customize Application
---	-----------------------

1. Create a custom field. See [Create Custom Fields](#). You can also define a default value for an existing custom field. See [Edit Custom Fields](#).
2. Choose the type of field and click **Next**. For a list of the types available for default values, see [Default Field Values](#).
3. Enter the attributes for the field.
4. Enter a default value or [define a formula](#) to calculate the default value.



Note You can define a formula for default values only where appropriate. For example, the default value options for a checkbox field are limited to the options available for those types of fields, such as Checked or Unchecked. For picklists, a valid formula result is either a constant or the API name of an entry in the Values list. The formula result has higher precedence than the default assigned in the Values list. If the formula doesn't generate a valid result, the default assigned in the Values list is entered in the field. If a default isn't assigned to the Values list, no value is entered in the picklist field.

5. Click **Next**.
6. Set the field-level security to determine whether the field is visible for specific profiles, and click **Next**.
7. Choose the page layouts that display the field. The field is added as the last field in the first two-column section on the page layout. For user custom fields, the field is added to the bottom of the user detail page.
8. Click **Save** to finish or **Save & New** to create more custom fields.

You must specify a [default value](#) for required campaign member custom fields.

To avoid uniqueness errors, don't assign default values to fields that are both required and unique.

Default Field Values

Prepopulate field values with defaults to save your users time and improve consistency. Default field values make your users more productive by reducing the number of fields they must fill in manually.

Useful Default Field Value Formulas

How to apply default field values based on maximum discount rate, product language, or tax rate by city.

Default Field Value Considerations

Be aware of the behavior and rules when you assign default values to custom fields.

Default Field Values

Prepopulate field values with defaults to save your users time and improve consistency. Default field values make your users more productive by reducing the number of fields they must fill in manually.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Default field values automatically insert the value of a custom field when a new record is created. You can use a default value on a formula for some types of fields or exact values, such as Checked or Unchecked for checkbox fields.

After you have defined default values:

- The user chooses to create a record.
- The default field value is executed.
- Salesforce displays the edit page with the default field value prepopulated.
- The user enters the fields for the new record.
- The user saves the new record.

The user can change the field's value but the initial default field value is only executed one time, during record creation. For example, you can set the default field value on a custom lead field to seven days after the creation date to signify when to contact the lead again. You can change this value later, but you can't automatically restore the value that was seven days after the creation date.

Set up default field values for the following types of custom fields:

- Checkbox
- Currency
- Date
- Date/Time
- Email
- Number
- Percent
- Phone
- Picklist
- Text
- Text Area
- Time
- URL

For a description of these types, see [Custom Field Types](#).

See Also

- [Useful Default Field Value Formulas](#)
- [Default Field Value Considerations](#)
- [Define Default Field Values](#)
- [Reference Custom Metadata Type Records in Default Values](#)

Useful Default Field Value Formulas

How to apply default field values based on maximum discount rate, product language, or tax rate by city.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To view default field values:	View Setup and Configuration
To define or change default field values:	Customize Application

Maximum Discount Rate

Your organization applies different discount rates to opportunities based on the department of the person creating the opportunity. Use this example to set a default value for a custom field called **Discount Rate** on opportunities.

```
CASE(User.Department, "IT", 0.25, "Field", 0.15, 0)
```

In this example, the formula inserts a discount rate of 25% on any opportunity created by a user in the IT department or 15% on any opportunity created by someone in the Field department. A zero is applied if the creator doesn't belong to either of these departments. This field is a custom percent field on opportunities that uses the standard user field Department.

Product Language

You could associate a product with its language so that your users know the type of documentation or adapter to include. Use this default value formula to automatically set the language of a product based on the country of the user creating the product. In this example, the default value is Japanese if the user's country is Japan and English if the user's country is US. If neither is true, the default value unknown is inserted into the Product Language field.

```
CASE($User.Country , "Japan", "Japanese", "US", "English","unknown")
```

Tax Rate

Use this default value formula to set the tax rate of an asset based on the user's city. Create a custom percent field with this default value.

```
IF($User.City = "Napa", 0.0750,  
IF($User.City = "Paso Robles", 0.0725,  
IF($User.City = "Sutter Creek", 0.0725,  
IF($User.City = "Los Olivos", 0.0750,  
IF($User.City = "Livermore", 0.0875, null  
)  
)  
)
```

```
)  
)
```

In this example, a tax rate of 8.75% is applied to an asset when the user's address is in the city of Livermore. When none of the cities listed applies, the **Tax Rate** field is empty. You can also use the **Tax Rate** field in formulas to automatically calculate taxable amounts and final sales prices.

See Also

[Default Field Values](#)

[Default Field Value Considerations](#)

Default Field Value Considerations

Be aware of the behavior and rules when you assign default values to custom fields.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Default field values automatically insert the value of a custom field when a new record is created. You can use a default value on a formula for some types of fields or exact values, such as Checked or Unchecked for checkbox fields. Review the following considerations before incorporating default field values in your organization.

- If a default value is based on the value of a merge field, Salesforce uses the value of the merge field at the time the default value is executed. If the value of the merge field changes later, the default value is not updated.
- Users can change or remove the default field value on a record.
- Don't assign default values to fields that are both required and unique, because uniqueness errors can result.
- If you make an activity custom field universally required, you must also provide a default value.
- If an activity custom field is unique, you cannot provide a default value.
- Default field values are different from formula fields in the following ways: they are only executed once, at record creation; they are not read only; and the user can change the value but cannot restore the default field value.
- Since the default value is inserted before users enter any values in the new record, you cannot use the fields on the current record to create a default field value. For example, you cannot create a default field value on a contact that uses the first initial and last name because those values are not available when you click **New** to create a contact record. However, you can use the record type because it is selected before the record edit page displays.
- Record type default field values have precedence over an object's default field values.
- To apply a different default value for different record types, use the record type as a merge field in a

CASE function within the default field value setup.

- Fields that are not visible to the user due to field-level security are still available in the formula for a default field value.
- Connect Offline and Salesforce for Outlook do not display default values. However, Salesforce inserts the default values when a user syncs unless the user entered a value.
- Default field values are not available in the Self-Service portal.
- Lead conversion, Web-to-Lead, and Web-to-Case do not execute default field values.
- Visualforce pages don't support default field values.

 **Note** You can define a formula for default values only where appropriate. For example, the default value options for a checkbox field are limited to the options available for those types of fields, such as Checked or Unchecked. For picklists, a valid formula result is either a constant or the API name of an entry in the Values list. The formula result has higher precedence than the default assigned in the Values list. If the formula doesn't generate a valid result, the default assigned in the Values list is entered in the field. If a default isn't assigned to the Values list, no value is entered in the picklist field.

See Also

[Default Field Values](#)

[Useful Default Field Value Formulas](#)

Validation Rules

Improve the quality of your data using validation rules. Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Essentials, Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of "True" or "False". Validation rules also include an error message to display to the user when the rule returns a value of "True" due to an invalid value.

 **Important** We don't recommend using a custom number field that Einstein Prediction Builder manages in a validation rule. If the prediction changes the field value in a way that violates the validation rule, you can't update the record that uses the field. If you get a validation rule error when saving a record that contains an Einstein Prediction Builder field, edit, deactivate, or delete the validation rule.

After you have defined validation rules:

1. The user chooses to create a record or edit an existing record.

2. The user clicks **Save**.
3. All validation rules are verified.
 - If all data is valid, the record is saved.
 - If any data is invalid, the associated error message displays without saving the record.
4. The user makes the necessary changes and clicks **Save** again.

You can specify the error message to display when a record fails validation and where to display it. For example, your error message can be “The close date must occur after today’s date.” You can choose to display it near a field or at the top of the page. Like all other error messages, validation rule errors display in red text and begin with the word “Error”.

 **Note** In Salesforce Classic, users with custom profiles must have Read permission on a standard object to view validation rules for that object.

 **Important** Validation rules apply to new and updated records for an object, even if the fields referenced in the validation rule aren’t included in a page layout or an API call. Validation rules don’t apply if you create records for an object with Quick Create. If your organization has multiple page layouts for the object on which you create a validation rule, verify that the validation rule works on each layout. If your organization has any integrations that use this object, verify that the validation rule works for each integration.

Managing Validation Rules

Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of “True” or “False”. Validation rules also include an error message to display to the user when the rule returns a value of “True” due to an invalid value.

Define Validation Rules

Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of “True” or “False”. Validation rules also include an error message to display to the user when the rule returns a value of “True” due to an invalid value.

Clone Validation Rules

Create new validation rules quickly by using existing rules as your starting point.

Activate Validation Rules

How to activate and deactivate validation rules.

Validation Rules Fields

A list of fields and description for validation rules.

Tips for Writing Validation Rules

Keep these tips in mind when writing validation rules.

Validation Rule Considerations

Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of “True” or “False”. Validation rules also include an error message to display to the user when the rule returns a value of “True” due to an invalid value. Review

these considerations before implementing validation rules in your organization.

See Also

- [Examples of Validation Rules](#)
- [Custom Metadata Type Fields and Validation Rules](#)
- [Custom Metadata Types and Validation Rule Formulas](#)
- [Considerations for AI Prediction Fields](#)
- [Einstein Prediction Builder Editions and Permissions](#)
- [Edit Object Permissions in Profiles](#)

Managing Validation Rules

Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of “True” or “False”. Validation rules also include an error message to display to the user when the rule returns a value of “True” due to an invalid value.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Essentials, Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

From the validation rules page you can:

- [Define a validation rule.](#)
- Click **Edit** next to a rule name to update the [rule fields](#).
- Delete a validation rule.
- Click a validation rule name to view more details or to [clone the rule](#).
- [Activate a validation rule.](#)

See Also

- [Examples of Validation Rules](#)

Define Validation Rules

Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of “True” or “False”. Validation rules also include an error message to display to the user when the rule returns a value of “True” due to an invalid value.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Essentials, Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

USER PERMISSIONS NEEDED

To view field validation rules: View Setup and Configuration

To define or change field validation rules: Customize Application

Before creating validation rules, review the [Validation Rule Considerations](#).

1. From the management settings for the relevant object, go to Validation Rules.
2. In the Validation Rules related list, click **New**.



Note The detail page of a custom activity field does not list associated validation rules. To edit the validation rule for a custom activity field, select the validation rule from Setup by entering *Activities* in the Quick Find box, then selecting **Activities** and choose **Task Validation Rules** or **Event Validation Rules**.

3. Enter the [properties of your validation rule](#).
4. To check your formula for errors, click **Check Syntax**.

See Also

[Validation Rules](#)

[Clone Validation Rules](#)

[Tips for Writing Validation Rules](#)

[Examples of Validation Rules](#)

Clone Validation Rules

Create new validation rules quickly by using existing rules as your starting point.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Essentials, Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

USER PERMISSIONS NEEDED

To view field validation rules: View Setup and Configuration

To define or change field validation rules: Customize Application

1. From the management settings for the relevant object, go to Validation Rules.
2. In the Validation Rules related list, click the name of the validation rule.
3. Click **Clone**.

4. Define the new rule based on the original rule.
5. Click **Save** to finish or **Save & New** to create additional validation rules.



Note The detail page of a custom activity field doesn't list associated validation rules. To edit the validation rule for a custom activity field, select the validation rule from Setup by entering *Activities* in the Quick Find box, then selecting **Activities** and choose **Task Validation Rules** or **Event Validation Rules**.

See Also

[Define Validation Rules](#)
[Validation Rules Fields](#)
[Activate Validation Rules](#)

Activate Validation Rules

How to activate and deactivate validation rules.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Essentials, Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

USER PERMISSIONS NEEDED

To view field validation rules:	View Setup and Configuration
To define or change field validation rules:	Customize Application

1. From the management settings for the relevant object, go to Validation Rules.
2. Click **Edit** next to the rule you want to activate.
3. To activate the rule, select **Active**, and save your changes.
4. To deactivate the rule, deselect **Active**, and save your changes.



Note The detail page of a custom activity field doesn't list associated validation rules.

See Also

[Define Validation Rules](#)
[Validation Rules Fields](#)
[Find Object Management Settings](#)

Validation Rules Fields

A list of fields and description for validation rules.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Essentials, Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Field	Description
Rule Name	Unique identifier of up to 40 characters with no spaces or special characters such as extended characters.
Active	Checkbox that indicates if the rule is enabled.
Description	A 255-character or less description that distinguishes the validation rule from others. For internal purposes only.
Error Condition Formula	The expression used to validate the field. See Build a Formula Field and Formula Operators and Functions .
Error Message	<p>The message that displays to the user when a field fails the validation rule.</p> <p>If your organization uses the Translation Workbench, you can translate the error message into the languages Salesforce supports. See Enable or Disable Translation Workbench.</p>
Error Location	<p>Determines where on the page to display the error. To display the error next to a field, choose Field and select the field. If the error location is a field, the validation rule is also listed on the detail page of that field. If the error location is set to a field that is later deleted, to a field that is read only, or to a field that isn't visible on the page layout, Salesforce automatically changes the location to Top of Page.</p> <p>Error messages can only be displayed at the top of the page in validation rules for case milestones and Ideas.</p> <p>The Top of Page setting is respected in Salesforce Classic only. Validation rules that are set to Top of Page appear at the bottom of the page in Lightning Experience.</p>

See Also

[Define Validation Rules](#)

Tips for Writing Validation Rules

Keep these tips in mind when writing validation rules.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Essentials, Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

- Consider all the settings that can make a record fail validation, including assignment rules, field updates, field-level security, or hidden fields.
- Make sure to test a validation rule before activating it because if rules for the same field conflict, users can't save the record. Use the debug log to monitor the details of your rule implementation.



Tip A poorly designed validation rule can prevent users from saving valid data. Make sure you thoroughly test a validation rule before activating it. You can also use the debug log to monitor the details of your rule implementation.

- When referencing related fields in a validation formula, make sure those objects are deployed.
- Use the **RecordType.Id** merge field in your formula to apply different validations for different record types.
- Boolean error condition expression works. For example:
 - Correct: `CloseDate < TODAY()`
 - Incorrect: `IF(CloseDate < TODAY(), TRUE, FALSE)`
- If a validation rule contains the BEGINS or CONTAINS function, it processes blank fields as valid. For example, a validation rule that tests whether the serial number of an asset begins with 3, all assets with a blank serial number are considered valid.
- When using a validation rule to ensure that a number field contains a specific value, use the ISBLANK function to include fields that don't contain any value. For example, to validate that a custom field contains a value of 1, use this validation rule to display an error if the field is blank or shows any other number.

```
OR (ISBLANK (field__c), field__c<>1)
```

- Avoid using the IsClosed or IsWon opportunity merge fields in validation formulas. Instead, use the ISPICKVAL function to determine if the Stage contains the appropriate value. For example, this validation formula makes a custom Project Start Date field required whenever the Stage is Closed Won.

```
AND(ISPICKVAL(StageName, "Closed Won"),  
ISBLANK(Project_Start_Date__c))
```

- Simplify your validation formulas by using checkbox fields, which don't require an operator because they return true or false. For example, this validation formula checks to be sure an opportunity has opportunity products using the **HasOpportunityLineItem** merge field before users can save a change to it.

```
NOT (OR (ISNEW(), HasOpportunityLineItem))
```

- When creating or updating a validation rule, click **Insert Field** to check if a field is available for an

entity. If the field doesn't exist for an entity, an error appears.



Tips for Writing Validation Rule Error Messages

- Give instructions that tell the user the type of entry that's valid, such as Close Date must be after today.
- Include the field label to identify the field that failed validation, especially if the error message appears at the top of the page.
- When defining validation rules, you can set the error location to **Top of Page or Field**. If the error location is set to a field that's deleted later, read only, or to a field that isn't visible on the page layout, Salesforce automatically changes the location to **Top of Page**.
- To translate error messages, use the Translation Workbench.
- Assign corresponding numbers to validation rules and their error messages to identify the source of the error.

See Also

- [Define Validation Rules](#)
[Validation Rule Considerations](#)

Validation Rule Considerations

Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of “True” or “False”. Validation rules also include an error message to display to the user when the rule returns a value of “True” due to an invalid value. Review these considerations before implementing validation rules in your organization.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Essentials, Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

How Salesforce Processes Validation Rules

Salesforce processes rules in the following order:

- Validation rules
- Assignment rules
- Auto-response rules
- Workflow rules (with immediate actions)
- Escalation rules

In addition,

- When one validation rule fails, Salesforce continues to check other validation rules on that field or other fields on the page and displays all error messages at once.
- If validation rules exist for activities and you create an activity during lead conversion, the lead converts but a task isn't created.
- Validation rules are only enforced during lead conversion if validation and triggers for lead conversion are enabled in your organization.
- Campaign hierarchies ignore validation rules.
- Salesforce runs validation rules before it creates records submitted via Web-to-Lead and Web-to-Case and then creates records that have valid values.
- To give a default value to a division field before the validation rule evaluation, the division field must be on the page layout.
- Validation rules continue to run on individual records if the owner is changed. If the Mass Transfer tool is used to change the ownership of multiple records, however, validation rules don't run on those records.

Validation Rule Field Restrictions

Validation rule formulas don't or can't refer to:

- Compound fields, including addresses, first and last names, and dependent picklists and lookups
 - **Note** However, you can use compound fields in `ISNULL`, `ISBLANK`, and `ISCHANGED` functions.
- Campaign statistic fields, including statistics for individual campaigns and campaign hierarchies
- Merge fields for auto-number or compound address fields, such as Mailing Address
 - **Note** You can use merge fields for individual address fields, such as Billing City, in validation rule formulas.

In relation to other fields and functions in Salesforce, validation rules behave as follows:

- The detail page of a custom activity field doesn't list associated validation rules.
- Workflow rules and some processes can invalidate previously valid fields. Invalidation occurs because updates to records based on workflow rules and also on process scheduled actions don't trigger validation rules.
- Process record updates on immediate actions do fire validation rules.
- You can't create validation rules for relationship group members.
- You can use roll-up summary fields in validation rules because the fields don't display on edit pages. Don't use roll-up summary fields as the error location.
- Validation rules for the Content Version object aren't triggered when files are uploaded from Chatter.
- When uploaded from the Files Tab, validation rules are triggered for Lightning but not for Classic.

Lookup Filters vs. Validation Rules

Validation rules and lookup filters achieve similar ends, but offer different advantages. Use a lookup

filter:

- To improve user efficiency by limiting the number of available options in a lookup search dialog.
- To improve user efficiency by automating filters on lookup search dialogs that your users manually set.

Use a validation rule:

- If you're close to the maximum number of lookup filters allowed.
- To implement a complex business rule that requires you to use a formula. Formulas can reference fields that basic filter criteria can't reference, such as fields on the parent of the source object. Formulas can also use functions. For example, use `ISNEW` to apply the rule only on record creation, or `ISCHANGED` to apply the rule only when a field changes.

See Also

- [Define Validation Rules](#)
- [Activate Validation Rules](#)
- [Examples of Validation Rules](#)

Examples of Validation Rules

Review examples of validation rules for various types of apps that you can use and modify for your own purposes. *Validation rules* verify that the data a user enters in a record meets the standards you specify before the user can save the record.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

USER PERMISSIONS NEEDED

To view field validation rules: View Setup and Configuration

To define or change field validation rules: Customize Application

Use the following samples for validation rules in Salesforce and Salesforce AppExchange apps, including:

[Sample Account Address Validation Rules](#)

Rules to maintain valid account addresses.

[Sample Account Validation Rules](#)

Validation rule examples for numeric account numbers, account number length, and annual revenue range.

[Sample Call Center Validation Rules](#)

Validation rules for requiring a conditional description, preventing cases from being reset, restricting case status, and more.

[Sample Experience Cloud Site Validation Rules](#)

Validation rule examples for various use cases, like preventing offensive language in questions, replies, ideas, and idea comments.

[Sample Contact Validation Rules](#)

Validation rules for various use cases, like requiring fields for mailing address, mailing street, and ZIP code.

[Sample Cross Object Validation Rules](#)

Examples for three validation rules on opportunity products.

[Sample Date Validation Rules](#)

Examples for date validation rules. For example, how to validate that the value of a custom field is a weekday, a Saturday or Sunday, that a custom date field contains a date within the current month and year, and more.

[Sample Number Validation Rules](#)

Examples for how to validate that users can't save a time card record with more than 40 hours in a work week, numbers can't be negative, and even or odd numbers.

[Sample Opportunity Management Validation Rules](#)

Examples for Examples for how to validate custom fields and other fields on opportunities.

[Sample Quote Validation Rules](#)

An example on how to validate a quote.

[Sample User, Role, and Profile Validation Rules](#)

Examples on how to validate custom user, role, and profile fields.

[Miscellaneous Sample Validation Rules](#)

Examples for how to validate certain number formats for credit card numbers or drivers licences.

See Also

[Validation Rules](#)

[Define Validation Rules](#)

Sample Account Address Validation Rules

Rules to maintain valid account addresses.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

For more information on any of the formula functions used in these examples, see [Formula Operators and Functions by Context](#).

Canadian Billing Postal Code

Field	Value
Description:	Validates that the account Billing Zip/Postal Code is in the correct format if Billing Country is Canada.
Formula:	<pre>AND(OR(BillingCountry = "CAN", BillingCountry = "CA", BillingCountry = "Canada"), NOT(REGEX(BillingPostalCode, "((?i)[ABCEGHJKLMNPRSTVXY]\d[A-Z]?\s?\d[A-Z]\d?)")))</pre>
Error Message:	Canadian postal code must be in A9A 9A9 format.
Error Location:	Billing Zip/Postal Code

Billing Zip Code Is in Billing State

Field	Value
Description:	Validates that the account Billing Zip/Postal Code is valid by looking up the first five characters of the value in a custom object called Zip_Code__c that contains a record for every valid ZIP code in the US. If the ZIP code isn't found in the Zip_Code__c object, or the Billing State doesn't match the corresponding State_Code__c in the Zip_Code__c object, an error is displayed.
Formula:	<pre>VLOOKUP(\$ObjectType.Zip_Code__c.Fields.City__c , \$ObjectType.Zip_Code__c.Fields.Name , LEFT(BillingPostalCode,5)) <> BillingCity</pre>
Error Message:	Billing Zip Code doesn't exist in specified Billing State.
Error Location:	Billing Zip/Postal Code

US Billing Zip Code

Field	Value
Description:	Validates that the account Billing Zip/Postal Code is in 99999 or

Field	Value
	99999-9999 format if Billing Country is USA or US.
Formula:	<pre>AND (OR(BillingCountry = "USA", BillingCountry = "US"), NOT(REGEX(BillingPostalCode, "\\\d{5}(-\\\\d{4})?")))</pre>
	This example uses the REGEX function; see Shipping Zip Code if you aren't familiar with regular expressions.
Error Message:	ZIP code must be in 99999 or 99999-9999 format.
Error Location:	Billing Zip/Postal Code

Shipping Zip Code

Field	Value
Description:	Validates that the account Shipping Zip/Postal Code is in 99999 or 99999-9999 format if Shipping Country is USA or blank.
Formula:	<pre>AND (OR(ShippingCountry = "USA", ISBLANK(ShippingCountry)), OR(AND(LEN(ShippingPostalCode) <>5, LEN(ShippingPostalCode) <> 10), NOT(CONTAINS("0123456789", LEFT(ShippingPostalCode, 1))), NOT(CONTAINS("0123456789", MID(ShippingPostalCode , 2, 1))), NOT(CONTAINS("0123456789", MID(ShippingPostalCode , 3, 1))), NOT(CONTAINS("0123456789", MID(ShippingPostalCode , 4, 1))), NOT(CONTAINS("0123456789", MID(ShippingPostalCode , 5, 1))), AND(LEN(ShippingPostalCode) = 10, OR(MID(ShippingPostalCode , 6, 1) <> "-", NOT(CONTAINS("0123456789", MID(ShippingPostalCode , 7, 1))))))))</pre>

Field	Value
	<pre>NOT(CONTAINS("0123456789", MID(ShippingPostalCode , 8, 1))), NOT(CONTAINS("0123456789", MID(ShippingPostalCode , 9, 1))), NOT(CONTAINS("0123456789", MID(ShippingPostalCode , 10, 1)))))))</pre>
	<p>This example interprets a blank country as the US. To use this example with other countries, remove the clause that checks the length of the country field. Also, validation rule criteria are case-sensitive, so this rule is only enforced when the country is blank or “USA” in all capital letters. The rule isn’t enforced when the country is “usa.”</p> <p>You can also validate ZIP codes using a regular expression; for an example of a formula using a regular expression, see REGEX.</p>
Error Message:	ZIP code must be in 99999 or 99999-9999 format.
Error Location:	Shipping Zip/Postal Code

Valid Billing State (US)

Field	Value
Description:	Validates that the account Billing State/Province is a valid two-character abbreviation if Billing Country is US, USA, or blank.
Formula:	<pre>AND (OR(BillingCountry = "US", BillingCountry="USA", ISBL ANK(BillingCountry)), OR(LEN(BillingState) < 2, NOT(CONTAINS ("AL:AK:AZ:AR:CA:CO:CT:DE:DC:FL:GA:HI:ID:" & "IL:IN:IA:KS:KY:LA:ME:MD:MA:MI:MN:MS:MO:MT:NE:NV:N H:" & "NJ:NM:NY:NC:ND:OH:OK:OR:PA:RI:SC:SD:TN:TX:UT:VT:V A:" &</pre>

Field	Value
	<pre>"WA:WV:WI:WY", BillingState))))</pre>
	<p>This example interprets a blank country as the US. To use this example with other countries, remove the clause that checks the length of the country field. Also, validation rule criteria are case-sensitive, so this rule is only enforced when the country is blank or “USA” in all capital letters. The rule isn’t enforced when the country is “usa.”</p>
Error Message:	A valid two-letter state code is required.
Error Location:	Billing State/Province

Valid Billing Province (Canada)

Field	Value
Description:	Validates that the account Billing State/Province is a valid two-character abbreviation if Billing Country is CA or CAN.
Formula:	<pre>AND (OR(BillingCountry = "CA", BillingCountry="CAN"), OR(LEN(BillingState) < 2, NOT (CONTAINS ("AB:BC:MB:NB:NL:NT:NS:NU:ON:PC:QC:SK:YT", BillingState)))))))</pre>
Error Message:	A valid two-letter province code is required.
Error Location:	Billing State/Province

Valid Shipping State

Field	Value
Description:	Validates that the account Shipping State/Province is a valid two-character abbreviation if Shipping Country is US, USA, or blank.
Formula:	<pre>AND (</pre>

Field	Value
	<pre>OR(ShippingCountry = "US", ShippingCountry="USA", ISBLANK(ShippingCountry)), OR(LEN(ShippingState) < 2, NOT (CONTAINS ("AL:AK:AZ:AR:CA:CO:CT:DE:DC:FL:GA:HI:ID:" & "IL:IN:IA:KS:KY:LA:ME:MD:MA:MI:MN:MS:MO:MT:NE:NV:NH:" & "NJ:NM:NY:NC:ND:OH:OK:OR:PA:RI:SC:SD:TN:TX:UT:VT:VA:" & "WA:WV:WI:WY", ShippingState))))</pre>
	<p>This example interprets a blank country as the US. To use this example with other countries, remove the clause that checks the length of the country field. Also, validation rule criteria are case-sensitive, so this rule is only enforced when the country is blank or “USA” in all capital letters. The rule isn’t enforced when the country is “usa.”</p>
Error Message:	A valid two-letter state abbreviation is required.
Error Location:	Shipping State/Province

Valid Shipping Province (Canada)

Field	Value
Description:	Validates that the account Shipping State/Province is a valid two-character abbreviation, if Billing Country is CA or CAN.
Formula:	<pre>AND (OR(ShippingCountry = "CA", ShippingCountry="CAN"), OR(LEN(ShippingState) < 2, NOT (CONTAINS ("AB:BC:MB:NB:NL:NT:NS:NU:ON:PC:QC:SK:YT", ShippingState))))</pre>
Error Message:	A valid two-letter province abbreviation is required.
Error Location:	Shipping State/Province

Valid Billing Country

Field	Value
Description:	Validates that the account Billing Country is a valid ISO 3166 two-letter code.
Formula:	<pre> OR (LEN(BillingCountry) = 1, NOT (CONTAINS ("AF:AX:AL:DZ:AS:AD:AO:AI:AQ:AG:AR:AM:" & "AW:AU:AZ:BS:BH:BD:BB:BY:BE:BZ:BJ:BM:BT:BO:" & "BA:BW:BV:BR:IO:BN:BG:BF:BI:KH:CM:CA:CV:KY:" & "CF:TD:CL:CN:CX:CC:CO:KM:CG:CD:CK:CR:CI:HR:" & "CU:CY:CZ:DK:DJ:DM:DO:EC:EG:SV:GQ:ER:EE:ET:FK:" & "FO:FJ:FI:FR:GF:PF:TF:GA:GM:GE:DE:GH:GI:GR:GL:" & "GD:GP:GU:GT:GG:GN:GW:GY:HT:HM:VA:HN:HK:HU:" & "IS:IN:ID:IR:IQ:IE:IM:IL:IT:JM:JP:JE:JO:KZ:KE:KI:" & "KP:KR:KW:KG:LA:LV:LB:LS:LR:LY:LI:LT:LU:MO:MK:" & "MG:MW:MY:MV:ML:MT:MH:MQ:MR:MU:YT:MX:FM:MD:MC:" & "MC:MN:ME:MS:MA:MZ:MM:MA:NR:NP:NL:AN:NC:NZ:NI:" & "NE:NG:NU:NF:MP:NO:OM:PK:PW:PS:PA:PG:PY:PE:PH:" & "PN:PL:PT:PR:QA:RE:RO:RU:RW:SH:KN:LC:PM:VC:WS:" & "SM:ST:SA:SN:RS:SC:SL:SG:SK:SI:SB:SO:ZA:GS:ES:" & "LK:SD:SR:SJ:SZ:SE:CH:SY:TW:TJ:TZ:TH:TL:TG:TK:" & "TO:TT:TN:TR:TM:TC:TV:UG:UA:AE:GB:US:UM:UY:UZ:" & "VU:VE:VN:VG:VI:WF:EH:YE:ZM:ZW", BillingCountry))) </pre>
Error Message:	A valid two-letter country code is required.
Error Location:	Billing Country

Sample Account Validation Rules

Validation rule examples for numeric account numbers, account number length, and annual revenue range.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

For more information on any of the formula functions used in these examples, see [Formula Operators and Functions by Context](#).

Account Number Is Numeric

Field	Value
Description:	Validates that the Account Number is numeric if not blank.
Formula:	<pre>OR (ISBLANK(AccountNumber) , NOT (ISNUMBER(AccountNumber))))</pre>
Error Message:	Account Number is not numeric.
Error Location:	Account Number

Account Number Length

Field	Value
Description:	Validates that the Account Number is exactly seven digits (if it is not blank). The number seven is simply illustrative. You can change this to any number you like.
Formula:	<pre>AND (ISBLANK(AccountNumber) , LEN(AccountNumber) <> 7)</pre>
Error Message:	Account Number must be seven digits.
Error Location:	Account Number

Annual Revenue Range

Field	Value
Description:	Validates that the account Annual Revenue is not negative and does not exceed \$100 billion. This limit is designed to catch typos.

Field	Value
Formula:	<pre>OR (AnnualRevenue < 0, AnnualRevenue > 100000000000)</pre>
Error Message:	Annual Revenue cannot exceed 100 billion.
Error Location:	Annual Revenue

Sample Call Center Validation Rules

Validation rules for requiring a conditional description, preventing cases from being reset, restricting case status, and more.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Essentials, Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

For more information on any of the formula functions used in these examples, see [Formula Operators and Functions](#).

Conditionally Require Description When Case Reason is “Other”

Field	Value
Description:	Validates that a custom field called Other Reason contains a value if a case has a Case Reason of “Other.”
Formula:	<pre>AND (ISPICKVAL(Reason, "Other") , ISBLANK(Other_Reason__c))</pre>
Error Message:	Description of Other Reason is required.
Error Location:	Other Reason

Prevent Open Cases from Being Reset to New

Field	Value
Description:	If a case is already open, prevents the Status from being changed back to “New.”
Formula:	<pre>AND (ISCHANGED(Status), NOT (ISPICKVAL(PRIORVALUE(Status), "New")), ISPICKVAL(Status, "New"))</pre>
Error Message:	Open case Status cannot be reset to New.
Error Location:	Status

Restrict Status of Re-Opened Cases

Field	Value
Description:	Validates that the case Status is “Re-opened” when a closed case is opened again.
Formula:	<pre>AND (ISCHANGED(Status), OR (ISPICKVAL(PRIORVALUE(Status), "Closed"), ISPICKVAL(PRIORVALUE(Status), "Closed in SSP"), NOT(ISPICKVAL(Status, "Re-Opened"))))</pre>
Error Message:	Closed case can only be changed to “Re-opened.”
Error Location:	Status

Prevent Case Milestone Completion After Cases Are Closed

Field	Value
Description:	Validates that a milestone's Completion Date can't occur after the case's Status is Closed.

Field	Value
Formula:	<pre>Case.IsClosed = true</pre>
Error Message:	You can't complete a milestone after a case is closed.
Error Location:	Top of Page

Prevent Case Milestone Completion Before Case Creation Dates

Field	Value
Description:	Validates that the milestone's Completion Date has occurred after the case's Date/Time Opened .
Formula:	<pre>CompletionDate >= Case.CreatedDate && CompletionDate <= Case.ClosedDate</pre>
Error Message:	The milestone Completion Date must occur after the date the case was created and before the case was closed.
Error Location:	Top of Page

Sample Experience Cloud Site Validation Rules

Validation rule examples for various use cases, like preventing offensive language in questions, replies, ideas, and idea comments.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

For more information on any of the formula functions used in these examples, see [Formula Operators and Functions by Context](#).

Preventing Offensive Language in Questions

Field	Value
Description:	Prevents users from entering offensive language in the Title and

Field	Value
	Description fields when asking a question.
Formula:	OR (CONTAINS (Title, 'darn'), CONTAINS (Body, 'darn'))
Error Message:	Question title or description contains offensive language.

Preventing Offensive Language in Replies

Field	Value
Description:	Prevents users from entering offensive language when replying to a question.
Formula:	OR (CONTAINS (Body, 'darn'), CONTAINS (Body, 'dang'))
Error Message:	Reply contains offensive language.

Preventing Offensive Language in Ideas

Field	Value
Description:	Prevents users from entering offensive language in the Title and Description fields when posting an idea.
Formula:	OR (CONTAINS (Title, 'darn'), CONTAINS (Body, 'darn'))
Error Message:	Idea title or description contains offensive language.

Preventing Offensive Language in Idea Comments

Field	Value
Description:	Prevents users from entering offensive language when posting a comment.
Formula:	OR (CONTAINS (CommentBody, 'darn'), CONTAINS (CommentBody, 'dang'))
Error Message:	Comment contains offensive language.

Sample Contact Validation Rules

Validation rules for various use cases, like requiring fields for mailing address, mailing street, and ZIP code.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

For more information on any of the formula functions used in these examples, see [Formula Operators and Functions](#).

Mailing Address Fields Are Required

Field	Value
Description:	Validates that the contact Mailing Street , Mailing City , and Mailing Country are provided.
Formula:	<pre>OR (ISBLANK(MailingStreet), ISBLANK(MailingCity), ISBLANK(MailingCountry))</pre>
Error Message:	Mailing Street, City, and Country are required.
Error Location:	Top of Page

Mailing Street Is Required

Field	Value
Description:	Validates that the contact Mailing Street is provided.
Formula:	<pre>ISBLANK(MailingStreet)</pre>
Error Message:	Mailing Street is required.
Error Location:	Mailing Street

Mailing Zip Code

Field	Value
Description:	Validates that the contact Mailing Zip/Postal Code is in 99999 or 99999-9999 format if Mailing Country is USA or blank.
Formula:	<pre> AND(OR(MailingCountry = "USA", ISBLANK(MailingCountry)), OR(AND(LEN(MailingPostalCode) <>5, LEN(MailingPostalCode) <> 10), NOT(CONTAINS("0123456789", LEFT(MailingPostalCode, 1))), NOT(CONTAINS("0123456789", MID(MailingPostalCode , 2, 1))), NOT(CONTAINS("0123456789", MID(MailingPostalCode , 3, 1))), NOT(CONTAINS("0123456789", MID(MailingPostalCode , 4, 1))), NOT(CONTAINS("0123456789", MID(MailingPostalCode , 5, 1))), AND(LEN(MailingPostalCode) = 10, OR(MID(MailingPostalCode , 6, 1) <> "-", NOT(CONTAINS("0123456789", MID(MailingPostalCode , 7, 1))), NOT(CONTAINS("0123456789", MID(MailingPostalCode , 8, 1))), NOT(CONTAINS("0123456789", MID(MailingPostalCode , 9, 1))), NOT(CONTAINS("0123456789", MID(MailingPostalCode , 10, 1)))))))))))</pre>

This example interprets a blank country as US. To use this example with other countries, remove the clause that checks the length of the country field. Also, validation rule criteria are case sensitive, so this rule is only enforced when the country is blank or “USA” in all capital letters. The rule is not enforced when the country is “usa.”

Field	Value
	You can also validate zip codes using a regular expression; for an example of a formula using a regular expression, see REGEX .
Error Message:	Zip code must be in 99999 or 99999-9999 format.
Error Location:	Mailing Zip/Postal Code

Phone Number Has International Format

Field	Value
Description:	Validates that the Phone number begins with a plus sign (+) for country code. Note that this validation rule conflicts with the ten-digit rule.
Formula:	<code>LEFT(Phone, 1) <> "+"</code>
Error Message:	Phone number must begin with + (country code).
Error Location:	Phone

US Phone Number Has Ten Digits

Field	Value
Description:	Validates that the Phone number is in (999) 999-9999 format. This works by using the REGEX function to check that the number has ten digits in the (999) 999-9999 format.
Formula:	<code>NOT(REGEX(Phone, "\\\D*?(\\\d\\\\D*?) {10}"))</code>
Error Message:	US phone numbers should be in this format: (999) 999-9999.
Error Location:	Phone

Sample Cross Object Validation Rules

Examples for three validation rules on opportunity products.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

For more information on any of the formula functions used in these examples, see [Formula Operators and Functions by Context](#).

Discounts Must Be Within Range

The examples below work together to help you manage discount amounts for products and require a custom percent field on opportunity products called **Line Discount**. The examples below also require you to use price books and customize the **Product Family** field to include the following values: *Software, Consulting, and Training.*

Here is a validation rule for software discounts.

Field	Value
Description:	Prevents users from saving software products with a discount over 10 percent.
Formula:	<pre>AND(Line_Discount__c > 0.10, ISPICKVAL(Product2.Family, "Software"))</pre>
Error Message:	The discount must be 10% or less for software products.
Error Location:	Line Discount

Here is a validation rule for consulting discounts.

Field	Value
Description:	Prevents users from saving consulting products with a discount over 15 percent.
Formula:	<pre>AND(Line_Discount__c > 0.15, ISPICKVAL(Product2.Family, "Consulting"))</pre>
Error Message:	The discount must be 15% or less for consulting products.
Error Location:	Line Discount

Here is a validation rule for training discounts.

Field	Value
Description:	Prevents users from saving training products with a discount over 20 percent.
Formula:	<pre>AND(Line_Discount__c > 0.20, ISPICKVAL(Product2.Family, "Training"))</pre>
Error Message:	The discount must be 20% or less for training products.
Error Location:	Line Discount

Prevent Changing Opportunity Products on Closed Opportunities

This example consists of two validation rules: one on opportunity products and another on opportunities.

Field	Value
Description:	Prevents users from editing opportunity products after an opportunity is closed. Create the following validation rule example on opportunity products.
Formula:	<pre>OR(ISPICKVAL(Opportunity.StageName, "Closed Won"), ISPICKVAL(Opportunity.StageName, "Closed Lost"))</pre>
Error Message:	Cannot change opportunity products for closed opportunities.
Error Location:	Top of Page

The following validation rule is on opportunities.

Field	Value
Description:	Prevents users from deleting opportunity products after an opportunity is closed. Create the following validation rule example on opportunities. It uses a custom roll-up summary field on opportunities that counts the number of opportunity products on an opportunity.

Field	Value
Formula:	AND (OR (ISPICKVAL(StageName, "Closed Won"), ISPICKVAL(StageName, "Closed Lost")), Number_of_Line_Items__c < P RIORVALUE (Number_of_Line_Items__c))
Error Message:	Cannot delete opportunity products for closed opportunities.
Error Location:	Top of Page

Prevent Saving a Case When Account Does Not Have Support

Field	Value
Description:	Prevents users from saving a case for an account that does not have support. This example assumes you have a custom checkbox field on accounts called Allowed Support that tracks if the account has support.
Formula:	Account.Allowed_Support__c = FALSE
Error Message:	Unable to create cases for this account because it is not signed up for support.
Error Location:	Top of Page

Prevent Saving a Case When Contact is No Longer with the Company

Field	Value
Description:	Prevents users from saving an open case associated with a contact that is no longer with the company. This example uses a custom checkbox field on contacts called No Longer With Company .
Formula:	AND (Contact.Not_Longer_With_Compan y__c, NOT (IsClosed))

Field	Value
Error Message:	Unable to save this case because the related contact is no longer with the company. To continue, choose another contact.
Error Location:	Contact Name

Sample Date Validation Rules

Examples for date validation rules. For example, how to validate that the value of a custom field is a weekday, a Saturday or Sunday, that a custom date field contains a date within the current month and year, and more.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

For more information on any of the formula functions used in these examples, see [Formula Operators and Functions by Context](#).

Date Must Be a Weekday

Field	Value
Description:	Validates that the value of a custom date field is a weekday (not Saturday or Sunday).
Formula:	<pre>CASE(MOD(My_Date__c - DATE(1900, 1, 7), 7), 0, 0, 6, 0, 1) = 0</pre>
Error Message:	Date must be a weekday.
Error Location:	My Date

Date Must Be a Weekend Day

Field	Value
Description:	Validates that the value of a custom date field is a Saturday or Sunday.
Formula:	<pre>CASE(MOD(My_Date__c - DATE(1900, 1, 7), 7), 0, 1, 6, 1, 0) = 0</pre>
Error Message:	Date must be a weekend day.
Error Location:	My Date

Date Must Be in the Current Month

Field	Value
Description:	Validates that a custom date field contains a date within the current month and year.
Formula:	<pre>OR (YEAR(My_Date__c) <> YEAR(TODAY()), MONTH(My_Date__c) <> MONTH(TODAY()))</pre>
Error Message:	Date must be in the current month.
Error Location:	My Date

Date Must Be in the Current Year

Field	Value
Description:	Validates that a custom date field contains a date within the current year.
Formula:	<code>YEAR(My_Date__c) <> YEAR(TODAY())</code>
Error Message:	Date must be in the current year.
Error Location:	My Date

Date Must Be the Last Day of the Month

Field	Value
Description:	Validates whether a custom field called My Date is the last day of the month. To do this, it determines the date of the first day of the next month and then subtracts 1 day. It includes special case logic for December.
Formula:	<pre>DAY(My_Date__c) <> IF(Month(My_Date__c)=12, 31, DAY(DATE(YEAR(My_Date__c),MONTH(My_Date__c)+1,1) - 1))</pre>
Error Message:	Date must be the last day of the month.
Error Location:	My Date

Date Must Be Within One Year of Today

Field	Value
Description:	Validates whether a custom field called Follow-Up Date is within one year of today's date. This example assumes a 365 day year. (It does not handle leap years.)
Formula:	<pre>Followup_Date__c - TODAY() > 365</pre>
Error Message:	Follow-Up Date must be within one year of today.
Error Location:	Follow-Up Date

Day of Month Cannot Be Greater Than 15

Field	Value
Description:	Validates that a custom field called Begin Date contains a date in the first 15 days of the specified month.
Formula:	<pre>DAY(Begin_Date__c) > 15</pre>
Error Message:	Begin Date cannot be after the 15th day of month.
Error Location:	Begin Date

End Date Cannot Be Before Begin Date

Field	Value
Description:	Validates that a custom field called End Date does not come before another custom field called Begin Date .
Formula:	<code>Begin_Date__c > End_Date__c</code>
Error Message:	End Date cannot be before Begin Date.
Error Location:	Begin Date

Expiration Date Cannot Be Before Close Date

Field	Value
Description:	Validates that a custom field called Expiration Date does not come before Close Date .
Formula:	<code>Expiration_Date__c < CloseDate</code>
Error Message:	Expiration Date cannot be before Close Date.
Error Location:	Expiration Date

Sample Number Validation Rules

Examples for how to validate that users can't save a time card record with more than 40 hours in a work week, numbers can't be negative, and even or odd numbers.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

For more information on any of the formula functions used in these examples, see [Formula Operators and Functions](#).

Time Cards Must Total 40 Hours

Field	Value
Description:	Ensures that users cannot save a time card record with more than 40 hours in a work week. This example requires five custom fields on your custom object, one for each day of work.
Formula:	<pre>Monday_Hours__c + Tuesday_Hours__c + Wednesday_Hours__c + Thursday_Hours__c + Friday_Hours__c > 40</pre>
Error Message:	Your total hours cannot exceed 40.
Error Location:	Top of Page

Number Cannot Be Negative

Field	Value
Description:	Validates that a custom field called Hours Worked is not a negative number.
Formula:	<pre>Hours_Worked__c < 0</pre>
Error Message:	Hours Worked cannot be less than zero.
Error Location:	Hours Worked

Number Must Be Even

Field	Value
Description:	Validates that a custom field called Ark Passengers is a non-negative even number.
Formula:	<pre>OR (Ark_Passengers__c < 0, MOD(Ark_Passengers__c, 2) <> 0)</pre>

Field	Value
Error Message:	Ark Passengers must be a positive even number.
Error Location:	Ark Passengers

Number Must Be Odd

Field	Value
Description:	Validates that a custom field called Socks Found is a non-negative odd number.
Formula:	<pre>OR (Socks_Found__c < 0, MOD(Socks_Found__c, 2) = 0)</pre>
Error Message:	Socks Found must be an odd number.
Error Location:	Socks Found

Number Must Be a Multiple of Five

Field	Value
Description:	Validates that a custom field called Multiple of 5 is a multiple of five.
Formula:	<pre>MOD(Multiple_of_5__c, 5) <> 0</pre>
Error Message:	Number must be a multiple of five.
Error Location:	Multiple of 5

Number Must Be an Integer

Field	Value
Description:	Validates that a custom field called My Integer is an integer.
Formula:	<pre>FLOOR(My_Integer__c) <> My_Integer__c</pre>
Error Message:	This field must be an integer.

Field	Value
Error Location:	My Integer

Number Must Be Between -50 and 50

Field	Value
Description:	Validates that a custom field called Volume is between -50 and 50.
Formula:	<code>ABS (Volume__c) > 50</code>
Error Message:	Volume must be between -50 and 50.
Error Location:	Volume

Number Range Validation

Field	Value
Description:	Validates that the range between two custom fields, Salary Min and Salary Max , is no greater than \$20,000.
Formula:	<code>(Salary_Max__c - Salary_Min__c) > 20000</code>
Error Message:	Salary range must be within \$20,000. Adjust the Salary Max or Salary Min values.
Error Location:	Salary Max

Percentage Must Be Between Zero and 100

Field	Value
Description:	Validates that a custom field called Mix Pct is between 0 and 100%. Note that percent fields are expressed divided by 100 in formulas (100% is expressed as 1; 50% is expressed as 0.5).
Formula:	<code>OR (Mix_Pct__c > 1.0, Mix_Pct__c < 0.0)</code>

Field	Value
Error Message:	Mix Pct must be between 0 and 100%.
Error Location:	Mix Pct

Sample Opportunity Management Validation Rules

Examples for Examples for how to validate custom fields and other fields on opportunities.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

For more information on any of the formula functions used in these examples, see [Formula Operators and Functions](#).

Conditionally-Required Field Based on Opportunity Stage

Field	Value
Description:	Validates that a custom field called Delivery Date is provided if an opportunity has advanced to the Closed Won or Negotiation/Review stage.
Formula:	<pre> AND (OR (ISPICKVAL(StageName, "Closed Won"), ISPICKVAL(StageName, "Negotiation/Review")), ISBLANK(Delivery_Date__c)))</pre>
Error Message:	Delivery Date is required for this stage.
Error Location:	Delivery Date

Close Date Cannot Be Prior to Current Month

Field	Value
Description:	Validates that the Close Date of an opportunity is not within a month

Field	Value
	prior to the current month. Note the use of ISNEW and ISCHANGED in this formula to ensure the condition is only checked when the opportunity is being created or the Close Date field is modified subsequently.
Formula:	<pre>AND (OR (ISNEW(), ISCHANGED(CloseDate), CloseDate < DATE(YEAR(TODAY()), MONTH(TODAY()), 1)))</pre>
Error Message:	Close Date cannot be prior to current month.
Error Location:	Close Date

Close Date Must Be a Future Date

Field	Value
Description:	Ensures that users do not change the Close Date of an opportunity to a day in the past.
Formula:	<pre>SampleDate < TODAY()</pre>
Error Message:	Close Date cannot be a day in the past.
Error Location:	Close Date

Discounts on Opportunities

Field	Value
Description:	Validates that a custom discount percent field is between 0 and 40%.
Formula:	<code>OR(Discount_Rate__c < 0, Discount_Rate__c > 0.40)</code>
Error Message:	The Discount Rate must not exceed 40%.
Error Location:	Discount Rate

High-Value Opportunity Must Be Approved Before Closed

Field	Value
Description:	Opportunities with amounts greater than \$50,000 require that a custom checkbox field called Approved is checked in order to change the stage to Closed Won or Closed Lost. To automate this, set field-level security on the Approved checkbox so that it can only be checked via a custom approval process (Enterprise Edition, Unlimited Edition, or Performance Edition).
Formula:	<pre> AND (OR (ISPICKVAL(StageName, "Closed Won"), ISPICKVAL(StageName, "Closed Lost")), (Amount > 50000), NOT(ISPICKVAL(Approval_Status__c , "Approved")))) </pre>
Error Message:	All high-value opportunities must be approved for closure. Click the Request Close button.
Error Location:	Top of Page

Opportunity Amount Cannot Exceed \$10 Million

Field	Value
Description:	Validates that opportunity Amount is positive and no more than \$10 million. This limit is designed to catch typos.
Formula:	<pre> OR (Amount < 0, Amount > 10000000) </pre>
Error Message:	Amount cannot exceed \$10 million.
Error Location:	Amount

Opportunity Check for Products

Field	Value
Description:	Validates that an opportunity has at least one opportunity product before users can save a change to an opportunity.
Formula:	NOT (OR (ISNEW(), HasOpportunityLineItem))
Error Message:	You must add products to this opportunity before saving.
Error Location:	Top of Page

Opportunity Must Have Products if Beyond “Needs Analysis” Stage

Field	Value
Description:	Validates that an opportunity has opportunity products before the Stage can move beyond Needs Analysis.
Formula:	AND (CASE(StageName, "Value Proposition", 1, "Id. Decision Makers", 1, "Perception Analysis", 1, "Proposal/Price Quote", 1, "Negotiation/Review", 1, "Closed Won", 1, 0) = 1, NOT (HasOpportunityLineItem))
Error Message:	Opportunity products are required to advance beyond the Needs Analysis stage.
Error Location:	Top of Page

Opportunity Name Format

Field	Value
Description:	Validates that an opportunity contains a hyphen as a way of enforcing an “[Account] - [Amount]” opportunity naming convention.

Field	Value
Formula:	<pre>FIND(" - ", Name) = 0</pre>
Error Message:	Opportunity Name should use “[Account] - [Amount]” format.
Error Location:	Opportunity Name

Prevent Sales Reps from Moving Opportunity Stage Backwards

Field	Value
Description:	Prevent sales reps from changing opportunity Stage “backwards” to specific values, once they have accepted the opportunity via a custom approval process. The approval process sets the custom Accepted Flag checkbox to True.
Formula:	<pre>AND (Accepted_Flag__c, OR (ISPICKVAL(StageName, "Stage 1"), ISPICKVAL(StageName, "Stage 2")))</pre>
Error Message:	Invalid stage for accepted opportunity.
Error Location:	Stage

Probability Must Be 100% for Won Opportunities

Field	Value
Description:	Validates that the probability of a won opportunity is properly set to 100%. This is useful for data cleanliness and reporting purposes.
Formula:	<pre>AND (ISPICKVAL(StageName, "Closed Won"), Probability <> 1)</pre>
Error Message:	Probability must be 100% for won opportunities.
Error Location:	Probability

Probability Must Be Zero for Lost Opportunities

Field	Value
Description:	Validates that the probability of a lost opportunity is properly set to zero. This is useful for data cleanliness and reporting purposes.
Formula:	<pre>AND (ISPICKVAL(StageName, "Closed Lost"), Probability <> 0)</pre>
Error Message:	Probability must be 0% for lost opportunities.
Error Location:	Probability

Project Start Date

Field	Value
Description:	Validates that a field is conditionally required based on the values of other fields. Use this validation formula to ensure that users include a Project Start Date for an opportunity that is closed/won.
Formula:	<pre>AND (ISPICKVAL(StageName, "Closed Won"), ISNULL(Project_Start_Date__c))</pre>
Error Message:	Project start date is required for won opportunities.
Error Location:	Project Start Date

Sample Quote Validation Rules

An example on how to validate a quote.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

For more information on any of the formula functions used in these examples, see [Formula Operators and Functions by Context](#).

Display Error if Quote Line Item Discount Exceeds 40%

Field	Value
Description:	Shows an error if a quote line item's discount exceeds 40%.
Formula:	Discount > .40
Error Message:	The discount on this quote line item cannot exceed 40%.
Error Location:	Discount on quote

Sample User, Role, and Profile Validation Rules

Examples on how to validate custom user, role, and profile fields.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

For more information on any of the formula functions used in these examples, see [Formula Operators and Functions](#).

Discount Percent Does Not Exceed Role-Based Limit

Field	Value
Description:	Validates that a custom field on opportunities called Discount Percent does not exceed a maximum value that varies depending on the user's role. The default maximum is 15%.
Formula:	<pre>Discount_Percent__c > VLOOKUP(\$ObjectType.Role_Limits__c.Fields.Limit__c, \$ObjectType.Role_Limits__c.Fields.Name, \$UserRole.Name)</pre>
Error Message:	Discount (%) exceeds limit allowed for your role.
Error Location:	Discount Percent

Expense Amount Does Not Exceed User's Max Allowed Expense

Field	Value
Description:	Validates a custom field called Expense Amount against a custom user field called Max Allowed Expense .
Formula:	<pre>Expense_Amount__c > \$User.Max_Allowed_Expense__c</pre>
Error Message:	Amount cannot exceed your maximum allowed expense.
Error Location:	Expense Amount

Only Record Owner Can Change Field

Field	Value
Description:	Ensures that only the record owner can make changes to a custom field called Personal Goal .
Formula:	<pre>AND (ISCHANGED(Personal_Goal__c), Owner <> \$User.Id)</pre>
Error Message:	Only record owner can change Personal Goal.
Error Location:	Personal Goal

Only Record Owner or Administrator Can Change Field

Field	Value
Description:	Ensures that a user can make changes to a custom field called Personal Goal only if the user is the record owner or has a custom profile of "Custom: System Admin."
Formula:	<pre>AND (ISCHANGED(Personal_Goal__c), Owner <> \$User.Id, \$Profile.Name <> "Custom: System Admin")</pre>

Field	Value
	\$Profile merge fields are only available in Enterprise, Unlimited, Performance, and Developer Editions.
Error Message:	Only record owner or administrator can change Personal Goal.
Error Location:	Personal Goal

Opportunity Close Date Can Only Be Back-Dated by Administrator

Field	Value
Description:	Validates that the Close Date of an opportunity does not fall prior to the current month, except for users who have a custom profile called "Custom: System Admin."
Formula:	<pre> AND (OR (ISNEW(), ISCHANGED(CloseDate), CloseDate < DATE(YEAR(TODAY()) , MONTH(TODAY()) , 1), \$Profile.Name <> "Custom: System Admin")) </pre>
	\$Profile merge fields are only available in Enterprise, Unlimited, Performance, and Developer Editions.
Error Message:	Close Date cannot be prior to current month.
Error Location:	Close Date

Miscellaneous Sample Validation Rules

Examples for how to validate certain number formats for credit card numbers or drivers licences.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

For more information on any of the formula functions used in these examples, see [Formula Operators](#)

[and Functions by Context.](#)

Allow Number to Be Increased but Not Decreased

Field	Value
Description:	Allows a custom field called Commit Amount to be increased but not decreased after initial creation. This rule uses the PRIORVALUE() function to compare the updated value of the field to its value prior to update.
Formula:	PRIORVALUE(Commit_Amount__c) > Commit_Amount__c
Error Message:	Commit Amount cannot be decreased.
Error Location:	Commit Amount

California Driver's License

Field	Value
Description:	Ensures that a custom field called Drivers License is in the correct A9999999 format when the Mailing State is "CA".
Formula:	AND(MailingState = "CA", NOT(REGEX(Drivers_License__c, "([A-Z]\\d{7})?")))
Error Message:	Invalid California driver's license format.
Error Location:	Drivers License

Force Users to Check “I Accept Terms” to Enter Certain Values

Field	Value
Description:	Uses a checkbox labeled “I accept terms” to force the user to select a checkbox in order to enter a value called Number of Days that exceeds their Paid Time Off (PTO) balance available.
Formula:	AND(

Field	Value
	<pre>NOT(I_accept_terms__c), Number_of_Days__c > \$User.PTO_Balance__c)</pre>
Error Message:	Request will cause a negative PTO balance. You must accept Negative PTO Balance terms.
Error Location:	I accept terms

Prohibit Changes to a Field After It Has Been Saved

Field	Value
Description:	Prevents users from changing a custom field called Guaranteed Rate after it has been saved initially.
Formula:	<pre>AND(NOT(ISNEW()), ISCHANGED(Guaranteed_Rate__c))</pre>
Error Message:	Guaranteed Rate cannot be changed.
Error Location:	Guaranteed Rate

Social Security Number Format

Field	Value
Description:	<p>Validates that a custom text field called SSN is formatted in 999-99-9999 number format (if it is not blank). The pattern specifies:</p> <ul style="list-style-type: none"> • Three single digits (0-9):\\d{3} • A dash • Two single digits (0-9):\\d{2} • A dash • Four single digits (0-9):\\d{4}
Formula:	<pre>NOT(</pre>

Field	Value
	<pre>OR (ISBLANK(Social_Security_Number__c), REGEX(Social_Security_Number__c , "[0-9]{3}-[0-9]{2}-[0-9]{4}")))</pre>
Error Message:	SSN must be in this format: 999-99-9999.
Error Location:	SSN

Valid Currency

Field	Value
Description:	Validates selected currency against an explicit subset of active currencies in your organization using the Currency picklist. Use this example if you only allow some of the active currencies in your organization to be applied to certain types of records.
Formula:	<pre>CASE(CurrencyIsoCode, "USD", 1, "EUR", 1, "GBP", 1, "JPY", 1, 0) = 0</pre>
Error Message:	Currency must be USD, EUR, GBP, or JPY.
Error Location:	Currency

Valid Credit Card Number

Field	Value
Description:	<p>Validates that a custom text field called Credit_Card_Number is formatted in 9999-9999-9999-9999 or 9999999999999999 number format when it is not blank. The pattern specifies:</p> <ul style="list-style-type: none"> • Four digits (0-9) followed by a dash: \d{4}-

Field	Value
	<ul style="list-style-type: none"> The aforementioned pattern is repeated three times by wrapping it in {3} Four digits (0-9) The OR character () allows an alternative pattern of 16 digits of zero through nine with no dashes: \d{16}
Formula:	NOT(REGEX(Credit_Card_Number__c , "(((\\d{4}-){3}\\d{4}) \\d{16}))?")
Error Message:	Credit Card Number must be in this format: 9999-9999-9999-9999 or 9999999999999999.
Error Location:	Credit Card Number

Valid IP Address

Field	Value
Description:	Ensures that a custom field called IP Address is in the correct format, four 3-digit numbers (0-255) separated by periods.
Formula:	NOT(REGEX(IP_Address__c , "^((25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\\.){3} (25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\\$"))
Error Message:	Error: IP Address must be in form 999.999.999.999 where each part is between 0 and 255.
Error Location:	IP Address

Website Extension

Field	Value
Description:	Validates a custom field called Web Site to ensure its last four characters are in an explicit set of valid website extensions.
Formula:	AND(

Field	Value
	<pre>RIGHT(Web_Site__c, 4) <> ".COM", RIGHT(Web_Site__c, 4) <> ".com", RIGHT(Web_Site__c, 4) <> ".ORG", RIGHT(Web_Site__c, 4) <> ".org", RIGHT(Web_Site__c, 4) <> ".NET", RIGHT(Web_Site__c, 4) <> ".net", RIGHT(Web_Site__c, 6) <> ".CO.UK", RIGHT(Web_Site__c, 6) <> ".co.uk")</pre>
Error Message:	Web Site must have an extension of .com, .org, .net, or .co.uk.
Error Location:	Web Site

Require Field Input to Ensure Data Quality

Improve the quality of data that users enter in Salesforce by creating universally required fields.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Connect Offline, Salesforce for Outlook, the Self-Service portal, Web-to-Lead, and Web-to-Case aren't available in **Database.com**

A universally required field is a custom field. It must have a value whenever a record is saved within Salesforce, the Lightning Platform API, Connect Offline, Salesforce for Outlook, the Self-Service portal, or automated processes, such as Web-to-Lead and Web-to-Case. Making a field required on a page layout or through field-level security ensures that users must enter a value. Making a field required universally gives you a higher level of data quality beyond the presentation level of page layouts.

You can make these types of custom fields universally required:

- Currency
- Date
- Date/Time
- Email

- Master-Detail Relationship (always required)
- Number
- Percent
- Phone
- Picklist
- Text
- Text Area
- URL

To make a custom field universally required, select the **Required** checkbox when defining the custom field.

 **Note** You must specify a default value for required campaign member custom fields. If you make a user field universally required, you must specify a default value for that field.

Relationship group members don't support universally required fields.

Considerations for Universally Required Fields

A universally required field is a custom field. It must have a value whenever a record is saved within Salesforce, the Lightning Platform API, Connect Offline, Salesforce for Outlook, the Self-Service portal, or automated processes such as Web-to-Lead and Web-to-Case. Review the following considerations before making your custom fields universally required.

Considerations for Universally Required Fields

A universally required field is a custom field. It must have a value whenever a record is saved within Salesforce, the Lightning Platform API, Connect Offline, Salesforce for Outlook, the Self-Service portal, or automated processes such as Web-to-Lead and Web-to-Case. Review the following considerations before making your custom fields universally required.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Standard Objects, Page Layouts, Connect Offline, Salesforce for Outlook, the Self-Service portal, Web-to-Lead, and Web-to-Cases are not available in **Database.com**

- Standard fields cannot be universally required.
- Universally required fields are required across all record types.
- Edit pages always display universally required fields, regardless of field-level security.
- When designing your page layouts, universally required fields:
 - Cannot be removed from a page layout

- Are automatically added to the end of the first section of a page layout if not already on it
- Cannot be read only or optional
- Display in bold, indicating they are always visible
- Are disabled on the field properties page because you cannot remove the required setting
- Universally required fields are only enforced during lead conversion if validation and triggers for lead conversion are enabled in your organization.
- Quick Create does not enforce universally required fields.
- If you make an activity custom field universally required, you must also provide a default value.
- You must include universally required fields in your import files or the import will fail.
- Don't assign default values to fields that are both required and unique, because uniqueness errors can result.
- You cannot make a field universally required if it is used by a field update that sets the field to a blank value.
- Required fields may be blank on records that existed before making the field required. When a user updates a record with a blank required field, the user must enter a value in the required field before saving the record.
- Web-to-Lead and Web-to-Case request data is not validated by Salesforce. Invalid data isn't saved when requests are submitted. For example, if your custom field is a currency field and a user enters alphabetic characters such as "Abc" instead of numbers, the request is still submitted but with no value saved in the custom currency field.

See Also

[Require Field Input to Ensure Data Quality](#)

About Field Sets

A field set is a grouping of fields. For example, you could have a field set that contains fields describing a user's first name, middle name, last name, and business title.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

When a field set is added to a Visualforce page, developers can loop over its fields and render them. If the page is added to a managed package, administrators can add, remove, or reorder fields in a field set to modify the fields presented on the Visualforce page without modifying any code. The same Visualforce page can present different sets of information, depending on which fields a subscriber prefers to keep.

Users with the Customize Application permission and delegated administrators can create or edit field sets for their organization or [edit any installed field set](#). Field sets are available on all standard objects that support custom fields, and any organization that supports creating Visualforce pages.

 **Note** Only fields available in the API can be added to field sets.

Fields added to a field set can be in one of two categories:

- If a field is marked as **Available for the Field Set**, it exists in the field set, but the developer hasn't presented it on the packaged Visualforce page. Administrators can display the field after the field set is deployed by moving it from the Available column to the In the Field Set column.
- If a field is marked as **In the Field Set**, the developer has rendered the field on the packaged Visualforce page by default. Administrators can remove the field from the page after the field set is deployed by removing it from the In the Field Set column.

Create and Edit Field Sets

Salesforce has a drag-and-drop WYSIWYG tool for creating and editing field sets. The enhanced field sets editor is enabled by default, and provides all of the functionality of the original editor, as well as additional functionality and an easier-to-use WYSIWYG interface.

Define a Field as Required Using Field Sets

You can define a field as required when you create or edit field sets. You may want to define a field as required to ensure a user enters the necessary information on a field.

See Also

[Create a Custom Field](#)

[Developer's Guide: Visualforce Developer's Guide](#)

Create and Edit Field Sets

Salesforce has a drag-and-drop WYSIWYG tool for creating and editing field sets. The enhanced field sets editor is enabled by default, and provides all of the functionality of the original editor, as well as additional functionality and an easier-to-use WYSIWYG interface.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create, edit, and delete field sets:

Customize Application or be a delegated administrator

 **Important** The total number of cross-object spans within the In the Field Set container can't exceed 25.

1. From the management settings for the appropriate object, go to Field Sets, and then click **New**.
2. Enter a Field Set Label. This label is the name presented to subscribers who install the field through a

managed package.

3. Optionally, enter a name for your field set. This name is used by your Visualforce page to reference the field set.
4. In the Where is this used? area, provide a brief description of which Visualforce pages use the field set, and for what purpose. This information helps a subscriber understand where and how an installed field set is being used, so that they can populate it with their own fields.
5. Save your changes.
6. To add fields to the field set, drag the fields from the object palette and drop them into the Available for the Field Set or the In the Field Set container. The fields in the Available for the Field Set container are not initially visible on the Visualforce page. The fields in the In the Field Set container are visible by default.

In the field set, you can span to fields that reference multiple objects. When you span a field into a field set that references multiple objects, the only field you can span to is the Name object.

You can drag a field from one container to the other. The vertical order of the In the Field Set list indicates the order of how the fields render on Visualforce pages.

7. To remove a field from the field set, drag the element back to the object palette, or click  next to the element.
8. To make a field required, double-click the element or click  next to it and select the **Required** checkbox.
 —Indicates the field is required and must have a value to save the record.
9. Save your changes.

After a field set is deployed in your organization, you can always mark fields that are in the Available for the Field Set list as In the Field Set, or vice versa.

- Find the field set that you want to edit. From Setup enter *Installed Packages* in the Quick Find box, select **Installed Packages**, click an installed package, and then click the field set you want to edit. Alternatively, if you know which object contains the field set you want to edit, go to the object detail page and click **Edit** in the field set related list.
- If you didn't create the field set initially, you're only able to edit the fields within the field set. To move fields between containers, drag a field from one container to the other. To change the order of a rendered field, drag a field up or down the list and drop the field in the order you want it to appear.
- Save your changes.

See Also

[About Field Sets](#)

Define a Field as Required Using Field Sets

You can define a field as required when you create or edit field sets. You may want to define a field as required to ensure a user enters the necessary information on a field.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create, edit, and delete field sets:

Customize Application or be a delegated administrator

The required field is only available in the In the Field Set container. If you define a field as required in the In the Field Set container, and remove the field from the In the Field Set, the required attribute is removed.

 **Note** If you remove fields that were made required by an installed managed package from the In the Field Set container, the required attribute isn't removed from those fields.

To define a field as required in a field set, see [Creating and Editing Field Sets](#).

See Also

[About Field Sets](#)

Roll-Up Summary Field

A roll-up summary field calculates values from related records, such as those in a related list. You can create a roll-up summary field to display a value in a master record based on the values of fields in a detail record. The detail record must be related to the master through a master-detail relationship. For example, you want to display the sum of invoice amounts for all related invoice custom object records in an account's Invoices related list. You can display this total in a custom account field called Total Invoice Amount.

REQUIRED EDITIONS

 **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

You can perform different types of calculations with a roll-up summary field. You can count the number of detail records related to a master record. Or, you can calculate the sum, minimum value, or maximum value of a field in the detail records. See [Create a Roll-Up Summary Field](#).

Before you begin creating roll-up summary fields for your org, review the implementation tips and best practices.

Administration

- Create roll-up summary fields on:
 - Any custom object that is on the master side of a master-detail relationship
 - Any standard object that is on the master side of a master-detail relationship with a custom object
 - Opportunities using the values of opportunity products related to the opportunity
 - Accounts using the values of related opportunities
 - Campaigns using campaign member status or the values of campaign member custom fields
-  **Note** Campaign member custom formula fields that reference fields derived from leads or contacts aren't supported.
- The types of fields you can calculate in a roll-up summary field depend on the type of calculation. For example:
 - Number, currency, and percent fields are available when you select *SUM* as the roll-up type.
 - Number, currency, percent, date, and date/time fields are available when you select *MIN* or *MAX* as the roll-up type.
- Sometimes you can't change the field type of a field that you reference in a roll-up summary field.
- Make sure that the filter for your roll-up summary doesn't encounter a formula field that results in *#Error!*. If one of your filter criteria uses a formula field that results in an error, no matches are returned for that filter criterion. For example, your roll-up summary filter is “Formula Field equals 10”. Two records contain errors, and one contains the value “10” in that field. In this case, your summary includes only the record with the value “10.”
- Salesforce doesn't recalculate the value of campaign roll-up summary fields when a lead or contact is deleted. Select the **Force a mass recalculation of this field** option on the edit page of the roll-up summary field to manually recalculate the value.
- You can't use long text area, multi-select picklist, **Description** fields, system fields like **Last Activity**, cross-object formula fields, and lookup fields in the field column of roll-up summary filters.
- Auto number fields aren't available as the field to aggregate in a roll-up summary field.
- After you create a roll-up summary field on an object, you can't convert the object's master-detail relationship into a lookup relationship.
- Roll-up summary fields aren't available for mapping lead fields of converted leads.

Management

- If a roll-up summary field doesn't contain cross-object field references or functions that derive values on the fly, such as *NOW* or *TODAY*, it can calculate the values of formula fields.
-  **Note** The value of a formula field can result in *#Error!*, which affects the summarized total. If your roll-up summary type is **COUNT**, records are included regardless of whether they contain a formula field with an error. However, when the **Field to Aggregate** is a formula field that results in *#Error!*, calculations of type **MIN**, **MAX**, and **SUM** exclude those formula values.
- Changes to the value of a roll-up summary field can trigger assignment rules to run. If a roll-up

summary field is part of the criteria in an assignment rule, the field's new value is used to evaluate whether to reassign the record.

- These changes cause a mass recalculation of roll-up summary fields. However, when these changes cause a recalculation of roll-up summary values, the recalculation doesn't trigger workflow rules and field validations.
 - Changing the roll-up summary definition, such as the object, function, or field being aggregated
 - Changing the expression of a formula field referenced in a roll-up summary field
 - Replacing picklist values for picklist fields referenced in the roll-up summary filter
 - Changing picklist record type definitions
 - Changing currency conversion rates
 - Changing price book entries
 - Selecting the **Force a mass recalculation of this field** option on the edit page of the roll-up summary field
- Calculating roll-up summary field values can take up to 30 minutes, depending on the number of records affected and other factors.
- You aren't prevented from creating roll-up summary fields that can result in invalid values, such as February 29 in a non-leap year. If a roll-up summary field results in an invalid value, the value isn't recalculated. The field continues to display with an invalid roll-up summary icon () until you change the values being summarized.
- If your org uses multiple currencies, the currency of the master record determines the currency of the roll-up summary field. For example, if the master and detail records are in different currencies, the detail record value is converted into the currency of the master record.
- Changing a conversion rate triggers roll-up summary fields to recalculate. If you're using multiple currencies, we recommend changing the conversion rate from Manage Currencies in Setup, and not from the API. If you change the rate from the API, related jobs that are less than 24 hours old can interfere with your change.
- If your org has advanced currency management enabled, currency roll-up summary fields are invalid if they're on accounts and summarizing opportunity values, on opportunities and summarizing custom object values, or on quotes and summarizing custom object values.
- Salesforce prevents users from saving a record if it invalidates a related record. For example, a master record has a validation rule that requires the roll-up summary field value to be greater than 100. If the user's change to a related child record would put the value over 100, the user can't save the record.
- If a lookup field references a record that was deleted, Salesforce clears the value of the lookup field by default. Alternatively, you can choose to prevent records from being deleted if they're in a lookup relationship.

To be used in a roll-up summary field with a **Roll-Up Type** of **COUNT** or **SUM**, the lookup field must have the **What to do if the lookup record is deleted?** option set to *Don't allow deletion of the lookup record that's part of a lookup relationship. If the option Clear the value of this field. You can't choose this option if you make the field required* is selected, you can't create a **COUNT** or **SUM** roll-up summary field that pulls data from your lookup field.

- When multiple currencies are enabled in an org and corporate currency is different from the currency set on the record, the UI and the database for roll-up summary field values can display different decimal values. Values in the UI are displayed as two decimal places, whereas the database displays the exact value, which can be several decimal places. This behavior is due to the way values are stored

in the database. The UI precision doesn't affect the precision of the database, which is a floating-point value.

In some cases, there can be small numerical remainders after deletion or filtering of records when you use SUM as the roll-up type. To correct the value, recalculate the value manually by selecting the **Force a mass recalculation of this field** option on the edit page of the roll-up summary field.

- When you delete a roll-up summary field using Metadata API, the field isn't saved in the Recycle Bin. The field is purged even if you don't set the `purgeonDelete` deployment option to `true`.

Best Practices

- Apply field-level security to your roll-up summary fields if they calculate values that you don't want visible to users. Fields that your users can't see due to field-level security settings on the detail record are still calculated in a roll-up summary field.
- If you have validation rules, consider how they affect roll-up summary fields. The value in a roll-up summary field changes when the values in the detail records change. So, validation errors can display when saving either the detail or master record.
- Because roll-up summary fields aren't displayed on edit pages, you can use them in validation rules but not as the error location for your validation.
- Avoid referencing a roll-up summary field from a child record. The roll-up summary fields referenced from a child record can have outdated values because the parent record isn't updated. Instead, reference roll-up summary fields from a parent record. Your roll-up summary fields always have updated values because that rule runs after the parent value is updated.

If you're trying to enforce a record limit of 25 on the parent roll-up summary field, create validation rules on your child objects. When you add a child record, your validation rule on the child object can check if the count is already 25 or greater.

```
AND (ISNEW(), Sample.Line_Count__c >= 25)
```

- Plan your implementation of roll-up summary fields carefully before creating them. After created, you can't change the detail object selected or delete any field referenced in your roll-up summary definition.
- Advanced currency management affects roll-up summary fields. If your org enables advanced currency management, delete the currency roll-up summary fields on accounts that summarize opportunity values and on opportunities that summarize custom object values. Otherwise, the fields continue to display with an invalid roll-up summary icon because their values are no longer calculated.
- Automatically derived fields, such as current date or current user, aren't allowed in a roll-up summary field. Forbidden fields include formula fields containing functions that derive values on the fly, such as DATEVALUE, NOW, and TODAY. Formula fields that include related object merge fields are also not allowed in roll-up summary fields.
- When you refer to a roll-up summary field in a list view or report, you can't use certain qualifiers, including:
 - Starts with
 - Contains
 - Does not contain
 - Includes

- Excludes
- Within

[Create a Roll-Up Summary Field](#)

Define a roll-up summary field on the object that's on the master side of a master-detail relationship.

See Also

- [Create a Custom Field](#)
- [Create a Roll-Up Summary Field](#)
- [Custom Fields Allowed Per Object](#)

Create a Roll-Up Summary Field

Define a roll-up summary field on the object that's on the master side of a master-detail relationship.

REQUIRED EDITIONS

! **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

USER PERMISSIONS NEEDED

To view roll-up summary field definitions: View Setup and Configuration

To edit roll-up summary field definitions: Customize Application

If a relationship doesn't exist, first create a master-detail relationship between the master object that displays the value and the detail object that contains the records you're summarizing.

To create a roll-up summary field:

1. Create a custom field on the object where you want the field displayed. Summary fields summarize the values from records on a related object, so the object on which you create the field is on the master side of a master-detail relationship. For instructions on creating a custom field, see [Create Custom Fields..](#)
2. Choose the **Roll-Up Summary** field type, and click **Next**.
3. Enter a field label and any other attributes, and click **Next**.
4. Select the object on the detail side of a master-detail relationship. This object contains the records that you want to summarize.
5. Select the type of summary.

COUNT	Totals the number of related records.
-------	---------------------------------------

SUM	Totals the values in the field you select in the Field to Aggregate option. Only number, currency, and percent fields are available.
MIN	Displays the lowest value of the field you select in the Field to Aggregate option for all directly related records. Only number, currency, percent, date, and date/time fields are available.
MAX	Displays the highest value of the field you select in the Field to Aggregate option for all directly related records. Only number, currency, percent, date, and date/time fields are available.

6. Enter filter criteria if you want a selected group of records in your summary calculation. If your organization uses multiple languages, enter filter values in your org's default language. When you use picklists to specify filter criteria, the selected values are stored in the org's default language. If you edit or clone existing filter criteria, first set the **Default Language** on the Company Information page to the same language that was used to set the original filter criteria. Otherwise, it's possible that the filter criteria aren't evaluated as expected.
7. Click **Next**.
8. Set the field-level security to determine whether the field is visible for specific profiles, and click **Next**.
9. Select the Dynamic Forms-enabled Lightning record pages that should include the field, then click **Next**.
If you don't have any Dynamic Forms-enabled Lightning record pages for the object, this step doesn't appear.
10. Choose the page layouts where you want to display the field.
The field is added as the last field in the first two-column section on the page layout. For user custom fields, the field is automatically added to the bottom of the user detail page.
11. Click **Save** to finish or **Save & New** to create more custom fields.

See Also

[Roll-Up Summary Field](#)

Lookup Filters

Improve user productivity and data quality with lookup filters. Lookup filters are administrator settings that restrict the valid values and lookup dialog results for lookup, master-detail, and hierarchical relationship fields.

REQUIRED EDITIONS

 **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: both Salesforce Classic and Lightning Experience

Available in: All Editions except for Database.com.

USER PERMISSIONS NEEDED

To manage lookup filters:	Customize Application
---------------------------	-----------------------

Administrators specify the restrictions by configuring filter criteria that compare fields and values on:

- The current record (source)
- The lookup object (target)
- The user's record, permissions, and role
- Records directly related to the target object

For example, you can:

- Restrict the Account Name field on opportunities to allow only accounts with a record type of Customer, filtering out Partner and Competitor.
- Restrict the Account Name field on opportunities to allow only active accounts.
- Restrict the Contact field on cases to allow only contacts associated with the account specified in the Account Name field on the case record.
- Restrict the Account Name field on cases to allow only users with the International Sales profile to create or edit cases for accounts outside the United States.

 **Tip** When you define a lookup filter, you can choose from a list of filter criteria that Salesforce suggests. The list is based on the relationships between objects in your org. To see the suggested criteria, select **Insert Suggested Criteria**.

In Salesforce Classic, administrators can make lookup filters required or optional. In Lightning Experience, all lookup filters are required, even if admins specify them as optional in Setup.

- For fields with required lookup filters, values that match the lookup filter criteria appear in the lookup dialog. When editing the record, users can't save invalid values that they type in the field. If a user tries to save an invalid value, Salesforce displays an error message, which administrators can customize.
- For fields with optional lookup filters (Salesforce Classic only), values that match the lookup filter criteria appear in the lookup dialog. To remove the filter and view all search results for the lookup field, users can select **Show all results** in the lookup dialog. Also, optional lookup filters let users save values that don't match the lookup filter criteria without Salesforce displaying any error message.

Lookup filter criteria can compare fields on the source object with different types of fields on the target object as long as the fields are compatible.

Source Object Field Type	Compatible Target Object Field Types
Currency	Currency, Roll-Up Summary
Date	Date, Date/Time, Roll-Up Summary

Date/Time	Date, Date/Time, Roll-Up Summary
Hierarchy	Hierarchy, Lookup, Master-Detail
Lookup	Hierarchy, Lookup, Master-Detail
Master-Detail	Lookup, Hierarchy, Master-Detail
Number	Number, Percent, Roll-Up Summary
Percent	Number, Percent, Roll-Up Summary
Picklist	Text, Text Area, Email, URL
Roll-Up Summary	Currency, Number, Date, Date/Time, Roll-Up Summary

Supported Objects

Salesforce supports lookup filters on relationship fields that point to:

- Accounts
- Assets
- Badges
- Badges Received
- Campaigns
- Cases
- Contacts
- Content Folders
- Contracts
- Endorsements
- Entitlements
- Ideas
- Leads
- Opportunities
- Order Products
- Orders
- Products
- Quotes
- Service contracts
- Skill Users
- Skills
- Social Personas
- Thanks
- User Provisioning Accounts
- User Provisioning Logs
- User Provisioning Requests
- Users

- Work Order Line Items
- Work Orders
- Zones
- Custom objects

Define Lookup Filters

Create and define lookup filters. Lookup filter criteria can compare fields of different types as long as they are compatible. Value-based filters are supported in Lightning Experience and Salesforce Classic.

Delete or Deactivate Lookup Filters

Deleting a lookup filter permanently removes it. You can't recover deleted lookup filters.

View a List of Lookup Filters for a Target Object

You can quickly see a list of all of the lookup filters that restrict the values of each target object. This is useful when creating similar filters for a target object. Also, lookup filters that reference fields on related objects count against the cross-object reference limit, which is the number of unique relationships allowed for a target object. The Related Lookup Filters list lets you see which lookup filters might impact that limit.

Dependent Lookups

A dependent lookup is a relationship field with a lookup filter that references fields on the source object. For example, you can configure the case Contact field to only show contacts associated with the account selected in the case Account Name field.

Considerations for Lookup Filters

Follow these guidelines when creating lookup filters.

Notes on Using Lookup Filters with Person Accounts

If your organization uses person accounts, keep these considerations in mind.

Lookup Filters Best Practices

Use these best practices when you create your lookup filter.

Lookup Filter Examples

Various examples for record types, record status, roles, and complex configurations in lookup filters.

Limitations on Lookup Filters

Keep these limitations in mind when working with lookup filters.

See Also

[Apply Lookup Searches](#)

Define Lookup Filters

Create and define lookup filters. Lookup filter criteria can compare fields of different types as long as they are compatible. Value-based filters are supported in Lightning Experience and Salesforce Classic.

REQUIRED EDITIONS

 **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: both Salesforce Classic and Lightning Experience

Available in: All Editions except for **Database.com**.

USER PERMISSIONS NEEDED

To define lookup filters:	Customize Application
---------------------------	-----------------------

1. From the management settings for the field's object, go to Fields.
2. Click **Edit** next to the name of the lookup or master-detail relationship field to which you want to apply the filter.
3. In the Lookup Filter Options section, click **Show Filter Settings**.
4. Specify the filter criteria a record must meet to be a valid value. To specify criteria, click **Insert Suggested Criteria** and choose from a list of suggested criteria, or manually enter your own criteria.
 - In the first column, click the lookup icon or start typing in the text box and select a field.
 - In the second column, select an operator.
 - In the third column, select *Value* if Salesforce should compare the field in the first column with a static value, or select *Field* if Salesforce should compare the field in the first column with the value of another field.
 - In the fourth column, enter the value or select the field that Salesforce should compare with the field in the first column.
 - Click **Add Filter Logic** to add Boolean conditions.
 - Select a suggested field from the Field text box. You can only select fields on the current record, the lookup object, or the user record. You can also choose related fields that are one relationship away from the lookup object. Salesforce assists you by listing the available fields and relationships when you click the lookup icon or click inside the text box.
5. Specify whether the filter is required or optional. For fields with optional lookup filters (Salesforce Classic only), values that match the lookup filter criteria appear in the lookup dialog. To remove the filter and view all search results for the lookup field, users can select **Show all results** in the lookup dialog. Also, optional lookup filters let users save values that don't match the lookup filter criteria without Salesforce displaying any error message. In Lightning Experience, all filters are required, even if admins specify them as optional in Setup. There's no **Show all results** view.

For required lookup filters, specify whether you want Salesforce to display the standard error message or a custom message when a user enters an invalid value.

6. Optionally, enter text to display in the lookup search dialog. Consider text that guides users in their searches and explains the business rule that the lookup filter implements.
7. Leave **Enable this filter** selected.
8. Save your changes.

See Also

[Considerations for Lookup Filters](#)

[Dependent Lookups](#)

[Lookup Filter Examples](#)

[Find Object Management Settings](#)

Delete or Deactivate Lookup Filters

Deleting a lookup filter permanently removes it. You can't recover deleted lookup filters.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions except for **Database.com**.

USER PERMISSIONS NEEDED

To define lookup filters:	Customize Application
---------------------------	-----------------------

1. From the management settings for the relationship field's object, go to Fields.
2. Scroll to the Custom Fields & Relationships related list.
3. Click the name of the field containing the lookup filter.
4. Click **Edit**.
5. To deactivate the lookup filter, deselect **Enable this filter**, then save your changes.
Deactivating a lookup filter preserves the lookup filter configuration but:
 - Prevents it from applying to the relationship field
 - Prevents it from impacting the cross-object references limit
 - Removes it as a dependency for fields referenced in the lookup filter criteria
6. To delete the lookup filter, click **Clear Filter Criteria**, then save your changes.

See Also

[Dependent Lookups](#)

[Considerations for Lookup Filters](#)

[Find Object Management Settings](#)

View a List of Lookup Filters for a Target Object

You can quickly see a list of all of the lookup filters that restrict the values of each target object. This is useful when creating similar filters for a target object. Also, lookup filters that reference fields on related objects count against the cross-object reference limit, which is the number of unique relationships allowed for a target object. The Related Lookup Filters list lets you see which lookup filters might impact that limit.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions except for **Database.com**.

USER PERMISSIONS NEEDED

To define lookup filters: Customize Application

To see which lookup filters affect the limit for a particular target object, from the management settings for the object, go to Related Lookup Filters.

See Also

[Dependent Lookups](#)
[Considerations for Lookup Filters](#)

Dependent Lookups

A dependent lookup is a relationship field with a lookup filter that references fields on the source object. For example, you can configure the case Contact field to only show contacts associated with the account selected in the case Account Name field.

REQUIRED EDITIONS

 **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions except for **Database.com**.

USER PERMISSIONS NEEDED

To manage dependent lookups: Customize Application

When a user changes the value of a referenced field on the source object, Salesforce immediately verifies that the value in the dependent lookup still meets the lookup filter criteria. If the value doesn't meet the criteria, an error message is displayed and users can't save the record until the value is valid.

If the referenced field on the source object is a lookup, master-detail, or hierarchy field, users can't change its value by typing. Instead, users must click the lookup icon and select a value in the lookup search dialog.

 **Tip** Dependent lookups are supported in Visualforce pages.

See Also

[Define Lookup Filters](#)
[Lookup Filter Examples](#)

Considerations for Lookup Filters

Follow these guidelines when creating lookup filters.

REQUIRED EDITIONS

! **Important** Where possible, we changed noninclusive terms to align with our company value of equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions except for **Database.com**.

USER PERMISSIONS NEEDED

To manage lookup filters:

Customize Application

- On the Fields page, the  icon indicates all fields with active lookup filters. The  icon indicates that the lookup filter is required.
- The lookup filters you create in Salesforce also appear in the partner portal and Customer Portal.
- Lookup filters are case-sensitive.
- If you convert a required lookup filter with a custom error message to be optional, Salesforce deletes the message.
- If you create a lookup filter that invalidates an existing value for that field, the value persists. However, when a user edits the record, Salesforce displays an error message and requires the user to change the invalid value before saving.
- You can't save changes that cause required lookup filters on related records to contain invalid values.
- Versions 16.0 and higher of the Salesforce API support lookup filters. Lookup filters are enforced when you load data through the API.
- If you configure a lookup filter to show inactive users only, the relationship field has no valid options. Inactive users are never valid for relationship fields that point to the User object.
- If you create a filtered lookup on a field that looks up to another object, deploy both objects into the organization at the same time.
- If the field criteria include a master-detail relationship field, lookup field filters don't work.
- If the value of a controlling field invalidates the value of a dependent master-detail relationship field, Salesforce doesn't display an error message.
- Dependent lookups are supported in Visualforce pages.
- To create a dependent lookup filter with ServiceResource.ResourceType, use only the first letter of the picklist value, for example T for Technician. See [ServiceResource](#) for more details.
- In Lightning Experience, a lookup filter doesn't work if a field referenced in the filtered lookup criteria isn't added to the page layout or list view. When a user opens the lookup dialog box, the value being searched is automatically deleted. To avoid this issue, add the missing field that is being used in the lookup field filter criteria to the page layout or list view.
- Salesforce Knowledge supports lookup filters to and from the Knowledge Article object, with limitations.
For details about the limitations, see [Lookup Filter Limitations - Knowledge Article Object](#).
- Value-based filters are supported in Lightning Experience and Salesforce Classic.
- If an unlocked 2GP package installs a lookup filter, then a later version of the package deletes that lookup filter, the lookup filter isn't removed from the subscriber org. The lookup filter must be deleted manually instead.

- When changing ownership of a record in Salesforce Classic, fields on the record that have active lookup filters aren't validated. As a workaround, we recommend doing lookup filter validation for a record's fields before changing the record's owner.

Lookup Filter Support Limitations - Knowledge Article Object

There are limitations when using lookup filters to and from Knowledge articles, as outlined here.

Limitations Using Lookup Filters From the Knowledge Article Object

Lookup filter limitations from the Knowledge Article Object

Salesforce Version	Lookup Filter Support
Lightning Experience	Supports lookup relationships for custom fields from Knowledge articles to another object (both standard and custom objects).
Classic	Does not support lookup search to any objects (custom or standard objects)

Limitations Using Lookup Filters to the Knowledge Article Object

Lookup filter limitations to the Knowledge Article Object

Lookup Scenario	Supported?
Lookup relationship or Master-Detail relationship using Salesforce Classic with Lightning disabled.	No
Deploying exported Knowledge object metadata using Metadata API for custom lookup fields on custom objects to Knowledge articles.	Yes
Filtering on Search and Recent Knowledge dropdown lists. Filters on the Search and Recent Knowledge dropdown lists are not respected.	No
Saving the lookup selections (from the Recent Knowledge dropdown list) to the default (primary) language Knowledge article version.	Yes
Saving the lookup selections (from the Recent Knowledge dropdown list) to Knowledge article versions other than the default (primary) language Knowledge article version.	No
Selecting and searching on Knowledge articles from the "Recent Knowledge" section when using the Lightning UI. A full search doesn't return results for any lookup filter used (even when there is no lookup filter applied). This limitation applies to the Lightning UI only.	No

Metadata API Limitations

For custom lookup fields from Knowledge articles to another object, deploying the exported Knowledge

object metadata using a Metadata API leads to the error “`Cannot specify 'lookupFilter' for a CustomField of type Lookup for entity Knowledge__kav`”.

Spanning Relationships in Lookup Filters

Filter criteria can include fields related to the target object (one level only). For example, a lookup field points to a contact. The lookup filter can reference fields on the account related to the contact using the Account Name relationship field. The lookup field can also reference fields on the contact related to the contact via the Reports To relationship field.

For required lookup filters, each field referenced on a related lookup object counts against the number of unique relationships allowed for the referenced object. The relationships aren't counted against the source object. For example, the two unique relationships described above count against the number allowed for the Contact object. Optional lookup filters don't count against the limit on the number of unique relationships allowed per object.

To see which lookup filters affect the limit for a particular target object, from the management settings for the object, go to Related Lookup Filters.

Lookup Filters vs. Validation Rules

Validation rules and lookup filters achieve similar ends, but offer different advantages.

Use a lookup filter:

- To improve user efficiency by limiting the number of available options in a lookup search dialog.
- To improve user efficiency by automating filters on lookup search dialogs that your users manually set.

Use a validation rule:

- If you're close to the maximum number of lookup filters allowed.
- To implement a complex business rule that requires you to use a formula. Formulas can reference fields that basic filter criteria can't reference, such as fields on the parent of the source object. Formulas can also use functions. For example, use `ISNEW` to apply the rule only on record creation, or `ISCHANGED` to apply the rule only when a field changes.

See Also

[Lookup Filters](#)

[Dependent Lookups](#)

Notes on Using Lookup Filters with Person Accounts

If your organization uses person accounts, keep these considerations in mind.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions except for **Database.com**.

- Person Accounts don't support Contact filters; however, Person Accounts support Account filters. For example, if the Account field has a dependent lookup filter that's added to a Person Account, dependent lookups are supported. If the Contact field has a dependent lookup filter that's added to a Person Account, dependent lookups aren't supported.
- Lookup filter criteria on Account Name only apply to business accounts, not person accounts. For example, your lookup filter criteria is `Account Name does not contain book`. Business accounts with "book" in the name, such as John's Bookstore, aren't valid. Person accounts with "book" in the name, such as John Booker, are valid. The person accounts show in the lookup dialog for the Account field. If you must filter on the name for a person account, use the First Name or Last Name fields instead.
- To restrict the values of a lookup field to one type of account (person or business), use the Is Person Account field in your lookup filter criteria. For example, to restrict a lookup to only person accounts, include the following in your lookup filter criteria: `Is Person Account equals True`.
- You can't package lookup filters that reference standard fields specific to person accounts, such as the Email and Title fields.

See Also

[Lookup Filters](#)

Lookup Filters Best Practices

Use these best practices when you create your lookup filter.

REQUIRED EDITIONS

! **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions except for **Database.com**.

Custom Help

Define custom help for fields with lookup filters to let users know about the business rule the filter enforces. For example, if the lookup filter restricts the **Account Name** on opportunities to only allow active accounts, define custom help that states *You can only associate active accounts with opportunities.*

Error Messages

To guide users who type invalid values, customize lookup filter error messages. For example, if

the lookup filter restricts the **Account Name** on opportunities to only allow active accounts, define an error message that states *Value doesn't exist or isn't an active account.*

 **Important** Salesforce translates the standard error message for required lookup filters, but not custom error messages. Use the Translation Workbench to translate lookup filter custom error messages. To restore the standard error message after modifying it, click **Reset to default message**.

Working with Master-Detail Relationship Fields

When creating a lookup filter on a master-detail relationship field, verify that the current values of the field on all of the detail records meet the criteria you specify. If you specify criteria that an existing value doesn't meet, Salesforce prevents the user from saving changes to the detail record. If this occurs, the user must first modify the value on the master record to meet the criteria. For example, consider a custom object with a master-detail relationship field that points to accounts. If you define a lookup filter that excludes all accounts with a **Create Date** before 01/01/2009, verify that no existing records of that custom object have a master-detail relationship with any account created before 2009. A quick way to do this is to create a report that shows all accounts with a **Create Date** before 01/01/2009.

Profile-Based Lookup Filters

When you reference the User object, such as Current User, use *Profile: ID* in filter criteria to define different filter criteria for different users (example: *Current User Profile: ID*), or to let administrators enter values that don't match the criteria. Avoid using *Profile: Name* due to technical limitations on standard and custom profiles.

If you enter *Current User Profile: Name* or *Profile: Name* in the Field column of your lookup filter criteria, Salesforce displays a lookup icon in that row. To select from a list of existing profiles rather than typing profile names, click the lookup icon.

Record IDs vs. Record Names

To reference a specific record in filter criteria, use the ID of the record instead of its name. IDs are always unique whereas names aren't.

Testing

After creating a lookup filter, test it to make sure it isn't too restrictive. Depending on their permissions, some users have read-only access to some relationship fields; ensure your lookup filters don't prevent those users from editing records critical to their job functions.

Dependent Lookups on Page Layouts and Mini Page Layouts in the Console

When designing page layouts with dependent lookups:

- If a dependent lookup is above its controlling field on a layout, make its lookup filter optional or redesign the layout. Moving a required dependent lookup above its controlling field can confuse users who typically start from the top of a page when entering data.
- Ensure that both the controlling and dependent fields are visible so users can correct invalid

values.

Lookup Filters and the Lookup Filter Fields Search Layout

Don't reference the same fields in both lookup filter criteria and the Lookup Filter Fields search layout. Users might assume that results from their custom search override administrator-controlled lookup filters.

See Also

[Lookup Filters](#)

Lookup Filter Examples

Various examples for record types, record status, roles, and complex configurations in lookup filters.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions except for **Database.com**.

USER PERMISSIONS NEEDED

To define lookup filters: Customize Application

Record Types in Lookup Filters

If the value of a relationship field should only consist of records with a particular record type, specify the record type in a lookup filter. For example, if the **Account Name** field on opportunities should only have accounts with a Customer Account custom record type, define the following lookup filter to restrict users to only creating or editing opportunities associated with accounts that have a Customer Account record type, excluding accounts with Partner Account and Competitor Account record types:

Filter Criteria	Account Name: Account Record Type equals value <i>Customer Account</i>
Custom Error Message	Account does not exist or is not a customer account.
Lookup Window Text	You can only associate customer accounts to an opportunity. Search results only display customer accounts.

Record Status in Lookup Filters

If the value of a relationship field should only consist of records with particular status, specify the status in a lookup filter. For example, consider a Job Application object with a relationship field that points to the **Position** object. If the relationship field should only have open positions, define the following lookup

filter to restrict users to only creating or editing job applications for positions with the **Status** field set to Open:

Filter Criteria	Position: Status equals value Open
Custom Error Message	<i>Position does not exist or is not an open position.</i>
Lookup Window Text	<i>You can associate only open positions with job applications. Search results display open positions only.</i>

Profiles in Lookup Filters

When a business rule does not apply to users with every profile, use the **Current User Profile** global variable fields to define lookup filters that only affect users with a particular profile.

For example, the following lookup filter on the Case object **Account Name** field restricts users with a “Domestic Sales” profile to only creating or editing opportunities associated with accounts that have a billing country of “USA” while allowing other users to associate opportunities with any account:

Filter Criteria	<ul style="list-style-type: none"> • Current User Profile: Name equals value Domestic Sales • Account Name: Billing Country equals value USA • Current User Profile: Name not equal to value Domestic Sales
Filter Logic	(1 AND 2) OR 3
Custom Error Message	<i>Account does not exist or the account billing country is not USA. Domestic sales reps can only create opportunities for accounts in the United States.</i>
Lookup Window Text	<i>Search results show only United States accounts in the for domestic sales representatives.</i>

You can modify the above example to simultaneously restrict users with a “Global Sales” custom profile to only associating opportunities to accounts with a non-US billing country:

Filter Criteria	<ul style="list-style-type: none"> • Current User Profile: Name equals value Global Sales • Account Name: Billing Country not equal to value USA • Current User Profile: Name equals value Domestic Sales • Account Name: Billing Country equals value USA • Current User Profile: Name not equal to value Global Sales, Domestic Sales
-----------------	---

Filter Logic	(1 AND 2) OR (3 AND 4) OR 5
Custom Error Message	<i>Account does not exist or the account billing country is not in your sales area. Sales reps can only create opportunities for accounts in their sales area.</i>
Lookup Window Text	<i>Search results only display accounts in your region.</i>

If you do not include line 5 in the filter criteria, users who are not in Global Sales or Domestic Sales cannot select or save any values on account records.

Roles in Lookup Filters

When a business rule does not apply to users in every role, use the `Current User Role` global variable fields to define lookup filters that only affect users with particular roles. For example, in a recruiting application that has a Position object with a lookup field to a Compensation Package object, you can restrict users from editing or creating positions that have an executive compensation plan unless they are executive administrators or vice presidents. To do this, define the following lookup filter on the Position object **Compensation Package Name** field:

Filter Criteria	<ul style="list-style-type: none"> • Current User Role: Name does not start with value VP • Current User Role: Name does not equal value Executive Administrator • Compensation Package: Plan Type does not equal value Executive • Current User Role: Name starts with value VP • Current User Role: Name equals value Executive Administrator
Filter Logic	((1 OR 2) AND 3) OR (4 OR 5)
Custom Error Message	<i>The compensation plan does not exist, or you have selected an executive compensation plan but do not have access to create executive positions.</i>
Lookup Window Text	<i>Search results only display compensation plans that are relevant to positions you are allowed to create.</i>

Include the condition you are testing and the opposite condition. In this example, lines 1, 2, and 3 of the filter criteria ensure that users who are not VPs or Executive Administrators *cannot* select Executive compensation plans, while lines 4 and 5 ensure that VPs and Executive Administrators *can* select Executive compensation plans.

Blank Values in Lookup Filters

Your lookup filter criteria might reference a field that users often leave blank. You can design your lookup filter criteria to accept blank values by using the **Add Filter Logic** in the filter criteria to create an OR condition. For example, if you have a **Partner Contact** custom field on opportunities, restrict the field to only allow contacts that are associated to an account with a Partner Account record type, or private contacts not associated with any account.

Filter Criteria	<ul style="list-style-type: none"> Partner Contact: Account: Account Record Type equals value Partner Account Partner Contact: Account: Account Name equals value
Filter Logic	1 OR 2
Custom Error Message	<i>The partner contact must be associated with a partner account, or must be a private contact.</i>
Lookup Window Text	<i>Search results only display contacts from partner accounts or your private contacts.</i>

User IDs in Lookup Filters

Using user IDs in optional lookup filters can significantly improve user efficiency by first showing lookup search dialog results that are most relevant to the user while still allowing users to see all results if necessary. For example, on a lookup field to accounts, you can create an optional lookup filter that restricts the search results to accounts that the user owns in the search lookup dialog results. If the user is looking for an account that someone else owns, the user can remove the filter.

Filter Criteria	Current User: User ID equals Field Account: Owner ID
Lookup Window Text	<i>By default, search results only display accounts you own. To search all accounts, click "Show all results."</i>

Simple Dependent Lookups

If the value of a relationship field should depend on the value of another relationship field on the current record, specify the field to field comparison in the criteria. For example, if the case **Contact Name** field should only have contacts associated to the account specified in the case **Account Name** field, use the following lookup filter:

Filter Criteria	Contact Name: Account ID equals field Case: Account ID
-----------------	--

Custom Error Message	<i>Contact does not exist or is not associated to the case account.</i>
Lookup Window Text	<i>Search results only display contacts associated to the case account.</i>

When comparing lookup fields in lookup filter criteria, Salesforce always uses the ID of the relationship field, not the name.

Complex Lookup Filters and Dependent Lookups

Achieving complex business rules with lookup filters often involves combining your rules with filter logic and fields of various types. For example, consider an app for booking conference rooms that has the following data model:

Object	Fields
Meeting	<ul style="list-style-type: none"> • Meeting Name • Office lookup to the Office object • Projector Required checkbox • Number of Participants number field • Conference Room lookup to the Conference Room object
Conference Room	<ul style="list-style-type: none"> • Conference Room Name • Has Projector checkbox • Number of Seats Available number field • Conference Room Location lookup to the Office object
Office	<ul style="list-style-type: none"> • Office Name

The following lookup filter on the meeting **Conference Room** field restricts the valid values to conference rooms that have a projector if the meeting requires one, as well as the necessary number of seats:

Filter Criteria	<ul style="list-style-type: none"> • Meeting: Projector Required equals field Meeting Conference Room: Has Projector • Meeting: Projector Required equals value <i>False</i> • Conference Room: Number of Seats Available greater or equal field Meeting: Number of Participants
-----------------	--

Filter Logic	(1 OR 2) AND 3
Custom Error Message	<i>Conference room not found or is insufficient for your meeting.</i>
Lookup Window Text	<i>Search results only display conference rooms that can support your meeting requirements.</i>

To refine the valid values even further, incorporate the office where the conference room is located:

Filter Criteria	<ul style="list-style-type: none"> • Meeting: Projector Required equals field Meeting Conference Room: Has Projector • Meeting: Projector Required equals value False • Conference Room: Number of Seats Available greater than field Meeting: Number of Participants • Meeting: Office equals Field Conference Room: Conference Room Location
Filter Logic	(1 OR 2) AND 3 AND 4
Custom Error Message	<i>Conference room not found or is insufficient for your meeting.</i>
Lookup Window Text	<i>Search results only display conference rooms that can support your meeting requirements.</i>

See Also

[Considerations for Lookup Filters](#)

Limitations on Lookup Filters

Keep these limitations in mind when working with lookup filters.

- Lookup filter criteria can't reference these types of fields:
 - Relationship fields on activities
 - System fields that are always read only, such as Created By and Modified By
 - Relationship fields that support queues, such as Case Owner and Lead Owner
- Each object can have up to five active required lookup filters. In Salesforce Classic, you can also have an unlimited number of optional lookup filters. If you reach the limit of required lookup filters, create optional filters. When a user saves a record, use validation rules to enforce your business rule. In Lightning Experience, all lookup filters are required.
- Lookup filters aren't available for external lookup relationship fields.
- Lookup filters on currency fields don't convert currencies. For example, your organization uses multiple currencies and there's lookup filter criteria *Expected Revenue greater than 100000*. The lookup shows all records with an Expected Revenue field value greater than 100,000, regardless of the

currency.

- You can't use special date values, such as "Today" or "This Month," in lookup filter criteria.
- You can't delete fields that are referenced in an active lookup filter.
- You can't change the field type of fields referenced by an active lookup filter.
- You can add up to 10 criteria rows in a lookup filter.
- Lookup filter criteria can't reference these types of fields on the source object:
 - Autonumber
 - Email
 - Encrypted
 - Formula

 **Note** Formula fields containing "\$USER", "\$USERROLE", "\$PROFILE", or "\$SOURCE" are supported.

- GeoLocation
 - Long text area
 - Multi-select picklist
 - Phone
 - Roll-up summary
 - Text
 - Text (Encrypted)
 - Text area (Rich)
 - Text area (Long)
 - Time
 - URL
- Lookup auto-completion doesn't work for user lookups with other dropdown lists. Auto-completion is primarily for organizations that have set up either a partner portal or customer portal.
 - In enhanced list views, you can't change fields that dependent lookup filter criteria reference.
 - Lookup filters don't support mass owner changes. If your lookup filter criteria reference the Owner field, performing a mass owner change can result in incorrect values. The incorrect values aren't noticeable until you try to save the record.
 - If a formula references global merge fields that the lookup filter doesn't support, the lookup filter can't reference the formula.
 - Lookup filter criteria on Account Name only apply to business accounts, not person accounts. For example, your lookup filter criteria is `Account Name does not contain book`. Business accounts with "book" in the name, such as John's Bookstore, aren't valid. Person accounts with "book" in the name, such as John Booker, are valid. The person accounts show in the lookup dialog for the Account field. If you must filter on the name for a person account, use the First Name or Last Name fields instead.
 - Lookup filters aren't available for the standard Product field when you create an OrderItem object.

Fields: What's Different or Not Available in the Salesforce Mobile App

Not every Lightning Experience feature is in the Salesforce mobile app. Find out what's different.

REQUIRED EDITIONS

Available in: Lightning Experience and the Salesforce mobile app for iOS and Android

Available in: all editions

Fields

Unsupported Fields

- division fields

Combo Boxes

- Combo boxes, which combine a picklist with a text field, aren't available. Typically the text field is available but the picklist is not.

Lookup Fields

- User-defined lookup filter fields aren't supported.
- You can't create a record from a lookup field like you can in Lightning Experience. You can create an account from the Account Name lookup field on the Salesforce App Lead Conversion page. You can't create records from other lookup fields on Mobile.
- Lookup fields in Salesforce Classic show record names regardless of sharing permissions. As a result, users can see the names of records that they can't access. In Lightning Experience and the Salesforce mobile app, lookup fields respect sharing permissions and only show the name of records that the user can access. The one exception is owner lookup fields, which always display the name of the record's owner, regardless of sharing permissions.

Picklist Fields

- Controlling and dependent picklists are supported, but doesn't display indicators on create and edit pages for these fields. To determine if a picklist field is dependent, and which picklist field controls it, switch to the full site.
- Disabled picklists aren't grayed out like they are in the full site.

Phone Number Fields

- The keypad that displays in phone number fields doesn't include parentheses, hyphens, or periods, and doesn't apply any phone number formatting when you save the record. To apply a specific phone number format, edit the record in the full site.

Rich Text Area Fields

Support for rich text area fields varies by the version of the Salesforce mobile app and the type of device.

App Version	View Rich Text Area Fields	Edit Rich Text Area Fields
Salesforce for Android	Yes	Yes
Salesforce for iOS	Yes	Yes

User Fields

- While user detail pages aren't available in the app, user fields are supported and appear on user profiles, in related lists, and so forth.
- There are some issues when these user fields appear in related lists or mobile cards.
 - The **Company Name** field is blank if an internal user is viewing a mobile card or related list entry related to another internal user. If the referenced user is an external user, the company name appears correctly.
 - The **Active** field is blank unless the user is inactive.

Calculate Field Values with Formulas

A formula is an algorithm that derives its value from other fields, expressions, or values. Formulas can help you automatically calculate the value of a field based on other fields.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

Reports and Approvals aren't available in **Database.com**

Watch a Demo:  [Getting Started With Formulas \(Salesforce Classic\)](#)

This video gives a brief introduction to Salesforce formulas, accessing the formulas editor in the app, and how to use the editor tools to create formulas.

Where are Formulas Used in Salesforce?

Many areas in Salesforce use formulas. Before you begin using formulas, review the differences in their uses.

Formula Data Types

The data type of a formula determines the type of data you expect returned from your formula.

Elements of a Formula

A formula can contain references to the values of fields, operators, functions, literal values, or other formulas.

Formula Operators and Functions by Context

Use these operators and functions when building formulas. All functions are available everywhere that

you can include a formula, such as formula fields, validation rules, approval processes, and workflow rules, unless otherwise specified.

Using Date, Date/Time, and Time Values in Formulas

Date formulas are useful for managing payment deadlines, contract ages, or any other features of your organization that are time or date dependent.

Build a Formula Field

Your custom formula fields require special attributes.

Formula Field Limits and Restrictions

Before you create formula fields, be aware of their limits and limitations.

Formula Best Practices

You can use the Formula Editor in Salesforce to construct a simple formula with a few clicks. But what if you want to build something more complex? Use these tips to help you map out formula logic and make it easier to troubleshoot errors.

Examples of Advanced Formula Fields

Review examples of formula fields for various types of apps that you can use and modify for your own purposes.

Formulas: How Do I ... ?

A collection of topics around formulas, including common math calculation, text functions, and more.

Common Formula Errors

Review common errors that can occur with formulas and how to fix them.

Get an Explanation for a Formula and Fix Errors with Einstein for Formulas

Use Einstein for Formulas to get an explanation for a formula and fix syntax errors in a formula used in Formula fields, default field values, and record validation rules. As an admin, you can use Einstein's assistance for existing formulas or for the new ones that you create.

Where are Formulas Used in Salesforce?

Many areas in Salesforce use formulas. Before you begin using formulas, review the differences in their uses.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: All Editions

Reports and Approvals are not available in **Database.com**

Use Formulas for:	To:
Approval Processes	Define the criteria a record must meet to enter the approval process.
Approval Steps	Define the criteria a record must meet to enter the approval step.

Use Formulas for:	To:
Assignment Rules for Leads and Cases	Define the criteria the lead or case must meet for it to be assigned.
Auto-Response Rules for Leads and Cases	Define the criteria a lead or case must meet to trigger an auto-response rule.
Case Escalation Rules	Specify criteria a case must meet for it to be escalated.
Custom Buttons and Links	Define the content for custom links and buttons.
Custom Fields	Create custom formula fields that automatically calculate a value based on other values, merge fields, or expressions. Users can view formula fields on record detail pages but can't see the underlying algorithm or edit the value of a formula field. Custom formula fields are not available in Connect Offline, Web-to-Lead forms, or Web-to-Case forms.
Custom Summary Formulas in Reports	Automatically calculate more totals based on existing report summaries using the values, merge fields, or expressions you specify. Users can't change these totals.
Data Validations	Verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can include a formula such as <code>CloseDate >= TODAY()</code> .
Default Field Values	<p>Apply a value to a custom field when a user creates a record. Use formulas to define a default value such as <code>TODAY() + 7</code>.</p> <p>Users can change a default value. Default field values can be based on a formula using values, merge fields, or expressions you specify.</p>
Escalation Rules	Define the criteria that a case must meet to be escalated.
Formula Fields	Automatically calculate the value of a custom field using the values, merge fields, or expressions you specify. Users can't change the value of a formula field.
Reports	Create custom summary formulas in your reports to calculate more totals based on the existing summaries in that report.
S-Controls	Define the content for s-controls.
Validation Rules	Prevent users from entering an invalid value in a standard or custom field. Validation rules can be based on formulas and display an error message to users when the value they enter is not valid.
Workflow Field Updates	Automatically change the value of a field to a value you specify. The formula can include other values, merge fields, or expressions. You can set

Use Formulas for:	To:
	field updates to occur as a result of a workflow rule or an approval process.
Workflow Rules	Define the criteria a record must meet to trigger a workflow rule.
Visualforce Pages	Define the content for Visualforce pages.

Common Formula Processes

	When are they executed?	Read only?	Can include functions?	Can specify null handling?	Can include references to parent merge fields?
Default Field Values	Record creation	No	Yes	No	No
Formula Fields	Record display	Yes	Yes	Yes	Yes
Validation Rules	Record save	Not applicable	Yes	No	Yes
Workflow Rules	Record save	Not applicable	Yes	No	Yes
Approval Processes	Record submitted for approval	Not applicable	Yes	No	Yes
Field Updates	Workflow or approval process	Not applicable	Yes	No	Yes
Custom Summary Formulas for Reports	Report display	Yes	Yes, a limited subset of functions	Yes	No

Formula Data Types

The data type of a formula determines the type of data you expect returned from your formula.

Data Type	Description
Checkbox	Returns a true or false value. The field appears as a checkbox in record detail pages and reports. Use <code>True</code> for checked values and <code>False</code> for unchecked values.

Data Type	Description
Currency	<p>Returns a number in currency format of up to 18 digits with a currency sign.</p> <p>Salesforce uses the round-half-up tie-breaking rule for currency fields. For example, 23.5 becomes 24, 22.5 becomes 23, -22.5 becomes -23, and -23.5 becomes -24.</p>
Date	<p>Returns data that represents a day on the calendar. The current date can be acquired by calling the built-in function <code>TODAY()</code> in a formula. This data type isn't available for custom summary formulas in reports.</p>
Date/Time	<p>Returns data that represents a moment in time. A date/time field includes the date and also the time of day including hour, minutes, and seconds. You can insert the current date and time in a formula using the <code>NOW()</code> function. This data type isn't available for custom summary formulas in reports.</p>
Number	<p>Returns a positive or negative integer or decimal of up to 18 digits. Salesforce uses the round half up tie-breaking rule for numbers in formula fields. For example, 12.345 becomes 12.35 and -12.345 becomes -12.35.</p>
Percent	<p>Returns a number in percent format of up to 18 digits followed by a percent sign. Percent data is stored as a decimal divided by 100, which means that 90% is equal to 0.90.</p>
Text	<p>Returns a string of up to 3900 characters. To display text in addition to the formula output, insert that text in quotes. Use the text data type for text, text area, URL, phone, email, address, and auto-number fields. This data type isn't available for custom summary formulas in reports. Text area isn't a supported data type.</p>
Time	<p>Returns data that represents a moment in time, without the date. A time field includes the time of day by hour, minutes, seconds, and milliseconds. You can insert the current time in a formula using the <code>TIMENOW()</code> function.</p> <p>In formula expressions, use the international date format (ISO) for text arguments. For example, use <code>TIMEVALUE("11:30:00.000")</code> instead of <code>TIMEVALUE("11:30 AM")</code>.</p>

See Also

[Build a Formula Field](#)

Elements of a Formula

A formula can contain references to the values of fields, operators, functions, literal values, or other formulas.

Use any or all of these elements to build a formula.

Element Name	Description
Literal Value	<p>A text string or number you enter that is not calculated or changed. For example, if you have a value that's always multiplied by 2% of an amount, your formula would contain the literal value of 2% of that amount:</p> <div data-bbox="344 502 1455 593" style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <pre>ROUND((Amount*0.02), 2)</pre> </div> <p>This example contains every possible part of a formula:</p> <ul style="list-style-type: none"> • A function called ROUND used to return a number rounded to a specified number of decimal places. • A field reference called Amount. • An operator, *, that tells the formula builder to multiply the contents of the Amount field by the literal value, 0.02. • A literal number, 0.02. Use the decimal value for all percents. To include actual text in your formula, enclose it in quotes. • The last number 2 in this formula is the input required for the ROUND function that determines the number of decimal places to return.
Field Reference	<p>Reference the value of another custom or standard field using a merge field. The syntax for a merge field is <code>field_name</code> for a standard field or <code>field_name__c</code> for a custom field. The syntax for a merge field on a related object is <code>object_name__r.field_name</code>. Use the Insert Field button or the drop-down list to insert a merge field in your formula where necessary.</p> <p>To reference a field from a custom metadata type record, use</p> <div data-bbox="344 1453 1455 1543" style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <pre>\$CustomMetadata.CustomMetadataTypeAPIName.RecordAPIName.FieldAPIName</pre> </div> <p>.</p>
Function	<p>A system-defined formula that can require input from you and returns a value or values. For example, TODAY() does not require input but returns the current date. The TEXT(value) function requires your percent, number, or currency input and returns text.</p>
Operator	<p>A symbol that specifies the type of calculation to perform or the order in which to do it. For example, the + symbol specifies two values should be added. The open and close</p>

Element Name	Description
	parentheses specify which expressions you want evaluated first.
Comment	
	<p>An annotation within a formula that begins with a forward slash followed by an asterisk (<code>/*</code>). and concludes with an asterisk followed by a forward slash (<code>*/</code>). For example,</p> <pre data-bbox="367 517 832 544">/*This is a formula comment*/</pre> <p>Comments are ignored when processing a formula.</p> <p>Comments are useful for explaining specific parts of a formula to anyone viewing the formula definition. For example:</p> <pre data-bbox="367 813 1428 1151">AND (/*competitor field is required, check to see if field is empty */ LEN(Competitor__c) = 0, /* rule only enforced for ABCD record types */ RecordType.Name = "ABCD Value", /* checking for any closed status, allows for additional closed picklist values in the future */ CONTAINS(TEXT(StageName), "Closed"))</pre>

You can also use comments to *comment out* sections of your formula when debugging and checking the syntax to locate errors in the formula.

- Nesting comments causes a syntax error. For example, you cannot save a formula that has the following:

```
/* /* comment */ */
```

- Commenting out a whole formula causes a syntax error.
- Comments count against the character and byte size limits in formulas.

Formula Operators and Functions by Context

Use these operators and functions when building formulas. All functions are available everywhere that you can include a formula, such as formula fields, validation rules, approval processes, and workflow rules, unless otherwise specified.

Within an email template, merge fields can only be used in formula functions and operations when the

merge field belongs to the record that the email is related to. Otherwise, these fields don't resolve.

Extraneous spaces in these samples are ignored.

- [Math Operators](#)
- [Logical Operators](#)
- [Text Operators](#)
- [Date and Time Functions](#)
- [Logical Functions](#)
- [Math Functions](#)
- [Text Functions](#)
- [Summary Functions](#)
- [Advanced Functions](#)

Math Operators

Operator	Description
+ (Add)	Calculates the sum of two values.
- (Subtract)	Calculates the difference of two values.
* (Multiply)	Multiplies its values.
/ (Divide)	Divides its values.
^ (Exponentiation)	Raises a number to the power of a specified number.
() (Open Parenthesis and Closed Parenthesis)	Specifies that the expressions within the open parenthesis and close parenthesis are evaluated first. All other expressions are evaluated using standard operator precedence.

Logical Operators

Operator	Description
= and == (Equal)	Evaluates if two values are equivalent. The = and == operators are interchangeable.
<> and != (Not Equal)	Evaluates if two values aren't equivalent.
< (Less Than)	Evaluates if a value is less than the value that follows this symbol.
<a href;"="">> (Greater Than)	Evaluates if a value is greater than the value that follows this symbol.
<= (Less Than or Equal)	Evaluates if a value is less than or equal to the value that follows this symbol.
>= (Greater Than or Equal)	Evaluates if a value is greater than or equal to the value that follows this symbol.

Operator	Description
Equal)	symbol.
&& (And)	Evaluates if two values or expressions are both true. Use this operator as an alternative to the logical function AND.
 (Or)	Evaluates if at least one of multiple values or expressions is true. Use this operator as an alternative to the logical function OR.

Text Operators

Operator	Description
& and + (Concatenate)	Connects two or more strings.

Date and Time Functions

Function	Description
ADDMONTHS	Returns the date that is the indicated number of months before or after a specified date. If the specified date is the last day of the month, the resulting date is the last day of the resulting month. Otherwise, the result has the same date component as the specified date.
DATE	Returns a date value from the year, month, and day values that you enter. Salesforce displays an error on the detail page if the value of the DATE function in a formula field is an invalid date, such as February 29 in a non-leap year.
DATEVALUE	Returns a date value for a date/time or text expression.
DATETIMEVALUE	Returns a year, month, day, and GMT time value.
DAY	Returns a day of the month in the form of a number from 1 through 31.
DAYOFYEAR	Returns the day of the calendar year in the form of a number from 1 through 366.
FORMATDURATION	Formats the number of seconds with optional days, or the difference between times or dateTimes as HH:MI:SS.
HOUR	Returns the local time hour value without the date in the form of a number from 1 through 24.
ISOWEEK	Returns the ISO 8601-week number, from 1 through 53, for the given date, ensuring that the first week starts on a Monday.
ISOYEAR	Returns the ISO 8601 week-numbering year in 4 digits for the given date,

Function	Description
	ensuring that the first day is a Monday.
MILLISECOND	Returns a milliseconds value in the form of a number from 0 through 999.
MINUTE	Returns a minute value in the form of a number from 0 through 60.
MONTH	Returns the month, a number from 1 (January) through 12 (December) in number format of a given date.
NOW	Returns a date/time representing the current moment.
SECOND	Returns a seconds value in the form of a number from 0 through 60.
TIMENOW	Returns a time value in GMT representing the current moment. Use this function instead of the NOW function if you only want to track time, without a date.
TIMEVALUE	Returns the time value without the date, such as business hours.
TODAY	Returns the current date as a date data type.
UNIXTIMESTAMP	Returns the number of seconds since 1 Jan 1970 for the given date, or number of seconds in the day for a time.
WEEKDAY	Returns the day of the week for the given date, using 1 for Sunday, 2 for Monday, through 7 for Saturday.
YEAR	Returns the four-digit year in number format of a given date.

Logical Functions

Function	Description
AND	Returns a TRUE response if all values are true, and returns a FALSE response if one or more values are false.
BLANKVALUE	Determines if an expression has a value, and returns a substitute expression if it doesn't. If the expression has a value, returns the value of the expression.
CASE	Checks a given expression against a series of values. If the expression is equal to a value in the series, returns the corresponding result. If it isn't equal to a value in the series, returns the <code>else_result</code> .
IF	Determines if expressions are true or false. Returns a given value if true and another value if false.
ISBLANK	Determines if an expression has a value, and returns TRUE if it doesn't. If it contains a value, returns FALSE.
ISCLONE	Checks if the record is a clone of another record, and returns TRUE if one item

Function	Description
	is a clone. Otherwise, returns FALSE.
ISNEW	Checks if the formula is running during the creation of a new record, and returns TRUE if it is. If an existing record is being updated, returns FALSE.
ISNULL	Determines if an expression is null (blank), and returns TRUE if it is. If it contains a value, returns FALSE. You must use ISBLANK instead of ISNULL in new formulas. ISBLANK has the same functionality as ISNULL, but also supports text fields. Salesforce continues to support ISNULL, so you don't change any existing formulas.
ISNUMBER	Determines if a text value is a number, and returns TRUE if it is. Otherwise, returns FALSE.
NOT	Returns FALSE for TRUE and TRUE for FALSE.
NULLVALUE	Determines if an expression is null (blank) and returns a substitute expression if it is. If the expression isn't blank, returns the value of the expression. You must use BLANKVALUE instead of NULLVALUE in new formulas. BLANKVALUE has the same functionality as NULLVALUE, but it also supports text fields. Salesforce continues to support NULLVALUE, so changing the existing formulas isn't necessary.
OR	Determines if expressions are true or false. Returns TRUE if any expression is true, and returns FALSE if all expressions are false.
PRIORVALUE	Returns the previous value of a field.

Math Functions

Function	Description
ABS	Calculates the absolute value of a number. The absolute value of a number is the number without its positive or negative sign.
ACOS	Returns the arc cosign of the number in radians, if the given number is from -1 through 1. Otherwise, returns NULL.
ASIN	Returns the arc sine of the number in radians, if the given number is from -1 through 1. Otherwise, returns NULL.
ATAN	Returns the arc tangent of the number in radians.
ATAN2	Returns the arc tangent of the quotient of y and x in radians.

Function	Description
CEILING	Rounds a number up to the nearest integer, away from zero if negative.
CHR	Returns a string with the first character's code point as the given number.
COS	Returns the cosine of the number in radians, if the given number is from -1 through 1. Otherwise, returns NULL.
DISTANCE	Calculates the distance between two locations in miles or kilometers.
EXP	Returns a value for e raised to the power of a number that you specify.
FLOOR	Returns a number rounded down to the nearest integer, towards zero if negative.
FROMUNIXTIME	Returns the datetime that represents the given number as the seconds elapsed since 1 Jan 1970.
GEOLOCATION	Returns a geolocation based on the provided latitude and longitude. Must be used with the DISTANCE function.
LN	Returns the natural logarithm of a specified number. Natural logarithms are based on the constant e value of 2.71828182845904.
LOG	Returns the base 10 logarithm of a number.
MAX	Returns the highest number from a list of numbers.
MCEILING	Rounds a number up to the nearest integer, towards zero if negative.
MFLOOR	Rounds a number down to the nearest integer, away from zero if negative.
MIN	Returns the lowest number from a list of numbers.
MOD	Returns a remainder after a number is divided by a specified divisor.
PI	Returns pi.
PICKLISTCOUNT	Returns the number of selected values in a multi-select picklist.
ROUND	Returns the nearest number to a number that you specify, constraining the new number by a specified number of digits.
SIN	Returns the sine of the number, where the number is given in radians.
SQRT	Returns the positive square root of a given number.
TAN	Returns the tangent of the number, where the number is given in radians.
TRUNC	Truncates a number to a specified number of digits.

Text Functions

Function	Description
ASCII	Returns the first character's code point from the given string as a number.
BEGINS	Determines if text begins with specific characters. Returns TRUE if it does, and returns FALSE if it doesn't.
BR	Inserts a line break in a string of text.
CASESAFEID	Converts a 15-character ID to a case-insensitive 18-character ID.
CONTAINS	Compares two arguments of text, and returns TRUE if the first argument contains the second argument. If not, returns FALSE.
FIND	Returns the position of a string within a string of text represented as a number.
GETSESSIONID	Returns the user's session ID.
HTMLENCODE	Encodes text and merge field values for use in HTML by replacing characters that are reserved in HTML, such as the greater-than sign (>), with HTML entity equivalents, such as > .
HYPERLINK	Creates a link to a URL specified that is linkable from the text specified.
IMAGE	Inserts an image with alternate text and height and width specifications.
INCLUDES	Determines if any value selected in a multi-select picklist field equals a text literal that you specify.
INITCAP	Returns the text as lowercase with the first character of each word in uppercase.
ISPICKVAL	Determines if the value of a picklist field is equal to a text literal that you specify.
JSENCODE	Encodes text and merge field values for use in JavaScript by inserting escape characters, such as a backslash (\), before unsafe JavaScript characters, such as the apostrophe (').
JSINHTMLENCOD	Encodes text and merge field values for use in JavaScript inside HTML tags by replacing characters that are reserved in HTML with HTML entity equivalents and inserting escape characters before unsafe JavaScript characters. JSINHTMLENCOD (someValue) is a convenience function that is equivalent to JSENCODE (HTMLENCODE (someValue)) . That is, JSINHTMLENCOD first encodes someValue with HTMLENCODE , and then encodes the result with JSENCODE .
LEFT	Returns the specified number of characters from the beginning of a text string.
LEN	Returns the number of characters in a specified text string.

Function	Description
LOWER	Converts all letters in the specified text string to lowercase. Any characters that aren't letters are unaffected by this function. Locale rules are applied if a locale is provided.
LPAD	Inserts characters that you specify to the left-side of a text string.
MID	Returns the specified number of characters from the middle of a text string given the starting position.
REVERSE	Returns the characters of a source text string in reverse order.
RIGHT	Returns the specified number of characters from the end of a text string.
RPAD	Inserts characters that you specify to the right-side of a text string.
SUBSTITUTE	Substitutes new text for old text in a text string.
TEXT	Converts a percent, number, date, date/time, or currency type field into text anywhere formulas are used. Also, converts picklist values to text in approval rules, approval step rules, workflow rules, escalation rules, assignment rules, auto-response rules, validation rules, formula fields, field updates, and custom buttons and links.
TRIM	Removes the spaces and tabs from the beginning and end of a text string.
UPPER	Converts all letters in the specified text string to uppercase. Any characters that aren't letters are unaffected by this function. Locale rules are applied if a locale is provided.
URLENCODE	Encodes text and merge field values for use in URLs by replacing characters that are illegal in URLs, such as blank spaces, with the code that represent those characters as defined in <i>RFC 3986, Uniform Resource Identifier (URI): Generic Syntax</i> . For example, blank spaces are replaced with %20, and exclamation points are replaced with %21.
VALUE	Converts a text string to a number.

Summary Functions

These functions are available with summary, matrix, and joined reports.

Function	Description
PARENTGROUPVAL	Returns the value of a specified parent grouping. A “parent” grouping is any level above the one containing the formula. You can use this function only in custom summary formulas and at grouping levels for reports, but not at summary levels.

Function	Description
PREVGROUPVAL	Returns the value of a specified previous grouping. A “previous” grouping is one that comes before the current grouping in the report. Choose the grouping level and increment. The increment is the number of columns or rows before the current summary. The default is 1, the maximum is 12. You can use this function only in custom summary formulas and at grouping levels for reports, but not at summary levels.

Advanced Functions

Function	Description
CURRENCYRATE	Returns the conversion rate to the corporate currency for the given currency ISO code. If the currency is invalid, returns 1.0.
GETRECORDIDS	Returns an array of strings in the form of record IDs for the selected records in a list, such as a list view or related list.
IMAGEPROXYURL	Securely retrieves external images, and prevents unauthorized requests for user credentials.
INCLUDE	Returns content from an s-control snippet. Use this function to reuse common code in many s-controls.
ISCHANGED	Compares the value of a field to the previous value, and returns TRUE if the values are different. If the values are the same, returns FALSE.
JUNCTIONIDLIST	Returns a JunctionIDList based on the provided IDs.
LINKTO	Returns a relative URL in the form of a link (href and anchor tags) for a custom s-control or Salesforce page.
PREDICT	Returns an Einstein Discovery prediction for a record based on the specified record ID or for a list of fields and their values.
REGEX	Compares a text field to a regular expression, and returns TRUE if there's a match. Otherwise, returns FALSE. A regular expression is a string used to describe a format of a string according to certain syntax rules.
REQUIRESCRIPT	Returns a script tag with source for a URL that you specify. Use this function when referencing the Lightning Platform AJAX Toolkit or other JavaScript toolkits.
URLFOR	Returns a relative URL for an action, s-control, Visualforce page, or a file in a static resource archive in a Visualforce page.
VLOOKUP	Returns a value by looking up a related value on a custom object similar to the VLOOKUP() Excel function.

All Formula Operators and Functions

Use operators and functions when building formulas. All functions are available everywhere that you can include a formula such as formula fields, validation rules, approval processes, and workflow rules, unless otherwise specified.

See Also

- [Examples of Advanced Formula Fields](#)
- [Considerations for the Time Custom Field Type](#)

All Formula Operators and Functions

Use operators and functions when building formulas. All functions are available everywhere that you can include a formula such as formula fields, validation rules, approval processes, and workflow rules, unless otherwise specified.

+ (Add)

Calculates the sum of two values.

- (Subtract)

Calculates the difference of two values.

* (Multiply)

Multiplies its values.

/ (Divide)

Divides its values.

^ (Exponentiation)

Raises a number to a power of a specified number.

() (Open Parenthesis and Close Parenthesis)

Specifies that the expressions within the open parenthesis and close parenthesis are evaluated first. All other expressions are evaluated using standard operator precedence.

= and == (Equal)

Evaluates if two values are equivalent. The = and == operators are interchangeable.

<> and != (Not Equal)

Evaluates if two values aren't equivalent.

< (Less Than)

Evaluates if a value is less than the value that follows this symbol.

> (Greater Than)

Evaluates if a value is greater than the value that follows this symbol.

<= (Less Than or Equal)

Evaluates if a value is less than or equal to the value that follows this symbol.

>= (Greater Than or Equal)

Evaluates if a value is greater than or equal to the value that follows this symbol.

&& (AND)

Evaluates if two values or expressions are both true. Use this operator as an alternative to the logical

function AND.

|| (OR)

Evaluates if at least one of multiple values or expressions is true. Use this operator as an alternative to the logical function OR.

& and + (Concatenate)

Connects two or more strings.

ABS

Calculates the absolute value of a number. The absolute value of a number is the number without its positive or negative sign.

ADDMONTHS

Returns the date that is the indicated number of months before or after a specified date. If the specified date is the last day of the month, the resulting date is the last day of the resulting month. Otherwise, the result has the same date component as the specified date.

AND

Returns a TRUE response if all values are true; returns a FALSE response if one or more values are false.

BEGINS

Determines if text begins with specific characters and returns TRUE if it does. Returns FALSE if it doesn't.

BLANKVALUE

Determines if an expression has a value and returns a substitute expression if it doesn't. If the expression has a value, returns the value of the expression.

BR

Inserts a line break in a string of text.

CASE

Checks a given expression against a series of values. If the expression is equal to a value, returns the corresponding result. If it isn't equal to any of the values, returns the `else_result`.

CASESAFEID

Converts a 15-character ID to a case-insensitive 18-character ID. In Salesforce Classic, the function converts only valid Salesforce 15-character IDs. If you pass in an invalid ID, the function returns the ID passed in. In Lightning Experience, the function converts any 15-character ID.

CEILING

Rounds a number up to the nearest integer, away from zero if negative.

CONTAINS

Compares two arguments of text and returns TRUE if the first argument contains the second argument. If not, returns FALSE.

CURRENCYRATE

Returns the conversion rate to the corporate currency for the given currency ISO code. If the currency is invalid, returns 1.0.

DATE

Returns a date value from the year, month, and day values you enter. Salesforce displays an error on the detail page if the value of the DATE function in a formula field is an invalid date, such as February 29 in a non-leap year.

DATEVALUE

Returns a date value for a date/time or text expression.

DATETIMEVALUE

Returns a year, month, day, and GMT time value.

DAY

Returns a day of the month in the form of a number from 1 through 31.

DISTANCE

Calculates the distance between two locations in miles or kilometers.

EXP

Returns a value for e raised to the power of a number you specify.

FIND

Returns the position of a string within a string of text represented as a number.

FLOOR

Returns a number rounded down to the nearest integer, towards zero if negative.

GEOLOCATION

Returns a geolocation based on the provided latitude and longitude. Must be used with the DISTANCE function.

GETRECORDIDS

Returns an array of strings in the form of record IDs for the selected records in a list, such as a list view or related list.

GETSESSIONID

Returns the user's session ID.

HOUR

Returns the local time hour value without the date in the form of a number from 1 through 24.

HTMLENCODE

Encodes text and merge field values for use in HTML by replacing characters that are reserved in HTML, such as the greater-than sign (>), with HTML entity equivalents, such as > .

HYPERLINK

Creates a link to a URL specified that is linkable from the text specified.

IF

Determines if expressions are true or false. Returns a given value if true and another value if false.

IMAGE

Inserts an image with alternate text and height and width specifications.

IMAGEPROXYURL

Securely retrieves external images and prevents unauthorized requests for user credentials.

INCLUDE

Returns content from an s-control snippet. Use this function to reuse common code in many s-controls.

INCLUDES

Determines if any value selected in a multi-select picklist field equals a text literal you specify.

ISBLANK

Determines if an expression has a value and returns TRUE if it does not. If it contains a value, this function returns FALSE.

ISCHANGED

Compares the value of a field to the previous value and returns TRUE if the values are different. If the values are the same, this function returns FALSE.

ISCLONE

Checks if the record is a clone of another record and returns TRUE if one item is a clone. Otherwise, returns FALSE.

ISNEW

Checks if the formula is running during the creation of a new record and returns TRUE if it is. If an existing record is being updated, this function returns FALSE.

ISNULL

Determines if an expression is null (blank) and returns TRUE if it is. If it contains a value, this function returns FALSE.

ISNUMBER

Determines if a text value is a number and returns TRUE if it is. Otherwise, returns FALSE.

ISPICKVAL

Determines if the value of a picklist field is equal to a text literal you specify.

JSENCODE

Encodes text and merge field values for use in JavaScript by inserting escape characters, such as a backslash (\), before unsafe JavaScript characters, such as the apostrophe (').

JSINHTMLENCODE

Encodes text and merge field values for use in JavaScript inside HTML tags by replacing characters that are reserved in HTML with HTML entity equivalents and inserting escape characters before unsafe JavaScript characters.

JUNCTIONIDLIST

Returns a JunctionIDList based on the provided IDs.

LEFT

Returns the specified number of characters from the beginning of a text string.

LEN

Returns the number of characters in a specified text string.

LINKTO

Returns a relative URL in the form of a link (href and anchor tags) for a custom s-control or Salesforce page.

LN

Returns the natural logarithm of a specified number. Natural logarithms are based on the constant e value of 2.71828182845904.

LOG

Returns the base 10 logarithm of a number.

LOWER

Converts all letters in the specified text string to lowercase. Any characters that are not letters are

unaffected by this function. Locale rules are applied if a locale is provided.

LPAD

Inserts characters you specify to the left-side of a text string.

MAX

Returns the highest number from a list of numbers.

MCEILING

Rounds a number up to the nearest integer, towards zero if negative.

MFLOOR

Rounds a number down to the nearest integer, away from zero if negative.

MID

Returns the specified number of characters from the middle of a text string given the starting position.

MILLISECOND

Returns a milliseconds value in the form of a number from 0 through 999.

MIN

Returns the lowest number from a list of numbers.

MINUTE

Returns a minute value in the form of a number from 0 through 60.

MOD

Returns a remainder after a number is divided by a specified divisor.

MONTH

Returns the month, a number from 1 (January) through 12 (December) in number format of a given date.

NOT

Returns FALSE for TRUE and TRUE for FALSE.

NOW

Returns a date/time representing the current moment.

NULLVALUE

Determines if an expression is null (blank) and returns a substitute expression if it is. If the expression is not blank, returns value of the expression.

OR

Determines if expressions are true or false. Returns TRUE if any expression is true. Returns FALSE if all expressions are false.

PARENTGROUPVAL

This function returns the value of a specified parent grouping. A “parent” grouping is any level above the one containing the formula.

PREDICT

Returns an Einstein Discovery prediction for a record based on the specified record ID or for a list of fields and their values.

PREVGROUPVAL

This function returns the value of a specified previous grouping. A “previous” grouping is one that comes before the current grouping in the report.

PRIORVALUE

Returns the previous value of a field.

REGEX

Compares a text field to a regular expression and returns TRUE if there is a match. Otherwise, returns FALSE.

REQUIRESCRIPT

Returns a script tag with URL source that you specify. Use this function when referencing the Lightning Platform AJAX Toolkit or other JavaScript toolkits.

REVERSE

Returns the characters of a source text string in reverse order.

RIGHT

Returns the specified number of characters from the end of a text string.

ROUND

Returns the nearest number to a number you specify, constraining the new number by a specified number of digits.

RPAD

Inserts characters that you specify to the right-side of a text string.

SECOND

Returns a seconds value in the form of a number from 0 through 60.

SQRT

Returns the positive square root of a given number.

SUBSTITUTE

Substitutes new text for old text in a text string.

TEXT

Converts a percent, number, date, date/time, or currency type field into text anywhere formulas are used. Also, converts picklist values to text in approval rules, approval step rules, workflow rules, escalation rules, assignment rules, auto-response rules, validation rules, formula fields, field updates, and custom buttons and links.

TIMENOW

Returns a time value in GMT representing the current moment. Use this function instead of the NOW function if you only want to track time, without a date.

TIMEVALUE

Returns the time value without the date, such as business hours.

TODAY

Returns the current date as a date data type.

TRIM

Removes the spaces and tabs from the beginning and end of a text string.

UPPER

Converts all letters in the specified text string to uppercase.

URLENCODE

Encodes text and merge field values for use in URLs by replacing characters that are illegal in URLs,

such as blank spaces, with the code that represent those characters as defined in RFC 3986, Uniform Resource Identifier (URI): Generic Syntax.

URLFOR

Returns a URL for an action, an s-control, a Visualforce page, or a file in a static resource archive. URLFOR is available for use in custom buttons and links, s-controls, and Visualforce pages.

VALUE

Converts a text string to a number.

VLOOKUP

Returns a value by looking up a related value on a custom object similar to the VLOOKUP() Excel function. This function is only available in validation rules.

WEEKDAY

Returns the day of the week for the given date, using 1 for Sunday, 2 for Monday, through 7 for Saturday.

YEAR

Returns the four-digit year in number format of a given date.

+ (Add)

Calculates the sum of two values.

Use

`value1 + value2` and replace each *value* with merge fields, expressions, or other numeric values.

-  **Note** If the values are text, the + concatenates.
-  **Formula Field Example** `Amount + Maint_Amount__c + Services_Amount__c` This formula calculates the sum of the product **Amount**, maintenance amount, and services fees. **Maint amount** and **Service Fees** are custom currency fields.
-  **Report Example** `EMAIL_OPT_OUT:SUM + DO_NOT_CALL:SUM` calculates all Email Opt Out fields plus all Do Not Call fields on the leads in your report. This formula is a number data type that returns a positive integer.
-  **Validation Rule Example** Let's say you have a custom object that allows users to track the total number of hours worked in a week. Use the following example to ensure that users can't save a time card record with more than 40 hours in a work week.

```
Monday_Hours__c +
Tuesday_Hours__c +
Wednesday_Hours__c +
Thursday_Hours__c +
Friday_Hours__c > 40
```

Use a formula like this one in a validation rule to display the following error message when the total number of hours entered for each work day is greater than 40: “Your total hours can’t exceed 40.” This example requires five custom fields on your custom object, one for each day of work.

- (Subtract)

Calculates the difference of two values.

Use

`value1 - value2` and replace each *value* with merge fields, expressions, or other numeric values.

-  **Calculate Difference Example** This formula calculates the difference of the product Amount less the Discount Amount. Discount Amount is a custom currency field. `Amount - Discount_Amount__c`
-  **Report Example** `AMOUNT:SUM - Product.Discount_Amount__c:SUM` calculates the difference of all Amount fields and all Discounted Amount custom fields on the products in your report. This formula is a currency data type that returns a currency sign and decimal places.

* (Multiply)

Multiplies its values.

Use

`value1 * value2` and replace each *value* with merge fields, expressions, or other numeric values.

-  **Consulting Days Example** This formula calculates the number of consulting days times 1200 given that this formula field is a currency data type and consulting charges a rate of \$1200 per day. **Consulting Days** is a custom field. `Consulting_Days__c * 1200`
-  **Report Example** `RowCount * AGE:AVG` calculates the record count times the average age value of your report. This formula is a number data type that returns a positive or negative integer or decimal.

/ (Divide)

Divides its values.

REQUIRED EDITIONS

Available in:

Use

`value1 / value2` and replace each *value* with merge fields, expressions, or other numeric values.

-  **Revenue Amount** `AnnualRevenue / NumberOfEmployees` This formula calculates the revenue amount per employee using a currency field.

```
IF(NumberOfOpportunities > 0,  
    NumberOfWonOpportunities / NumberOfOpportunities, null)
```

-  **% Won Opportunities Example** `WON:SUM / RowCount` calculates the percent of **Won** opportunities using a record count representing the number of all opportunities in your report. This formula is a number data type that returns a positive or negative integer.

-  **% Difference between Cost and Sales Price Example** `(TOTAL_PRICE:SUM - QUANTITY:SUM * Product2.Cost_c:SUM) / (QUANTITY:SUM * Product2.Cost_c:SUM)` calculates the average percent difference between what a product costs and its selling price on a product-by-product level. `Product2.Cost_c:SUM` is a custom currency field named **Cost** on products, which includes the cost of each product. This formula is a percent data type that returns a positive or negative integer. For best results, use this formula on a summary Opportunities with Products report that is summarized by Product Name and includes summary totals for Quantity, Total Price, and Cost.

`^` (Exponentiation)

Raises a number to a power of a specified number.

Use

number^{integer} and replace *number* with a merge field, expression, or another numeric value; replace *integer* with a merge field that contains an integer, expression, or any integer.

Tips

Avoid replacing *integer* with a negative number.

-  **Number of Employees Example** `NumberOfEmployees^4` calculates the number of employees to the 4th power.

-  **Report Example** `ACTIVE:SUM ^ 2` calculates the number of active Salesforce users to the 2nd power for administration. This formula is a number data type that returns a positive integer.

() (Open Parenthesis and Close Parenthesis)

Specifies that the expressions within the open parenthesis and close parenthesis are evaluated first. All other expressions are evaluated using standard operator precedence.

Use

(expression1) expression2... and replace each *expression* with merge fields, expressions, or other numeric values.

-  **Subtraction and Division Example** `(Unit_Value__c - Old_Value__c) / New_Value__c` calculates the difference between the old value and new value divided by the new value.

-  **Report Example** `(DURATIONHOURS:SUM * RowCount) / 24` calculates the duration of all event times the record count per 24 hours. This formula is a percent data type that returns a positive or negative integer or decimal, representing what percent of a day is spent on events.

= and == (Equal)

Evaluates if two values are equivalent. The = and == operators are interchangeable.

-  **Important** Don't use this function for a null comparison, such as `MyDateTime__c == null`. Use `ISBLANK` instead.

Use

expression1=expression2 or *expression1 == expression2*, and replace each *expression* with merge fields, expressions, or other numeric values.

-  **Due Date Example** `Due Date = CreatedDate + 5` returns true if the due date is equal to five days following a record's created date.

-  **Commission Amount Example**

```
IF(Probability =1, ROUND(Amount*0.02, 2), 0)
```

This formula calculates the 2% commission amount of an opportunity that has a probability of 100%. All other opportunities have a commission value of 0. Possible results:

- An opportunity with a Probability of 90% has a commission of 0.

- An opportunity with a Probability of 100% and an Amount of \$100,000 has a commission of \$2,000.

<> and != (Not Equal)

Evaluates if two values aren't equivalent.

Description:	Evaluates if two values aren't equivalent.
Use:	<i>expression1 <> expression2</i> or <i>expression1 != expression2</i> , and replace each <i>expression</i> with merge fields, expressions, or other numeric values.
Example:	<pre>IF(Maint_Amount__c + Services_Amount__c <> Amount, "DISCOUNTED", "FULL PRICE")</pre> <p>This formula displays DISCOUNTED on a product if its maintenance amount and services amount don't equal the product amount. Otherwise, displays FULL PRICE. Note that this example uses two custom currency fields for Maint Amount and Services Amount.</p>

 **Important** Don't use this function for a null comparison, such as `MyDateTime__c != null`. Use [ISBLANK](#) instead.

Use

expression1 <> expression2 or *expression1 != expression2*, and replace each *expression* with merge fields, expressions, or other numeric values.

Discount Example

```
IF(Maint_Amount__c + Services_Amount__c <> Amount,
    "DISCOUNTED", "FULL PRICE")
```

This formula displays DISCOUNTED on a product if its maintenance amount and services amount don't equal the product amount. Otherwise, displays FULL PRICE. Note that this example uses two custom currency fields for Maint Amount and Services Amount.

< (Less Than)

Evaluates if a value is less than the value that follows this symbol.

Use

Evaluates if a value is less than the value that follows this symbol.

-  **Revenue Example** `IF(AnnualRevenue < 1000000, 1, 2)` assigns the value 1 with revenues less than one million and the value 2 to revenues greater than one million.

> (Greater Than)

Evaluates if a value is greater than the value that follows this symbol.

Use

value1 > *value2* and replace each *value* with merge fields, expressions, or other numeric values.

-  **Net Worth Example** `IF(commission__c > 1000000, "High Net Worth", "General")` assigns the High Net Worth value to a commission greater than one million. Note, this is a text formula field that uses a commission custom field.

<= (Less Than or Equal)

Evaluates if a value is less than or equal to the value that follows this symbol.

Use

value1 <= *value2* and replace each *value* with merge fields, expressions, or other numeric values.

-  **Revenue Example** `IF(AnnualRevenue <= 1000000, 1, 2)` assigns the value 1 with revenues less than or equal to one million and the value 2 with revenues greater than one million.

>= (Greater Than or Equal)

Evaluates if a value is greater than or equal to the value that follows this symbol.

Use

value1 >= *value2* and replace each *value* with merge fields, expressions, or other numeric values.

-  **Commission Example** `IF(Commission__c >= 1000000, "YES", "NO")` assigns the YES value with a commission greater than or equal to one million. Note, this is a text formula field that uses a custom currency field called Commission.

&& (AND)

Evaluates if two values or expressions are both true. Use this operator as an alternative to the logical function AND.

Use

`(logical1) && (logical2)` and replace *logical1* and *logical2* with the values or expressions that you want evaluated.

 **Generic Example** `IF((Price<100 && Quantity<5), "Small", null)` This formula displays Small if the price is less than 100 and the quantity is less than five. Otherwise, this field is blank.

|| (OR)

Evaluates if at least one of multiple values or expressions is true. Use this operator as an alternative to the logical function OR.

Use

`(logical1) || (logical2)` and replace any number of logical references with the values or expressions you want evaluated.

 **Case Example** `IF((ISPICKVAL(Priority, "High")) || (ISPICKVAL(Status , "New")), ROUND(NOW()-CreatedDate, 0), null)` This formula returns the number of days a case has been open if the Status is new or the Priority is high. If the case was opened today, this field displays a zero.

 **Validation Rule Example**

```
(Discount_Rate__c < 0) || (Discount_Rate__c > 0.40)
```

This validation rule formula displays this error message when the **Discount Rate** custom field value isn't between 0% and 40%: "Discount Rate cannot exceed 40%."

& and + (Concatenate)

Connects two or more strings.

Use

`string1&string2` and replace each *string* with merge fields, expressions, or other values.



Trip Expense Example `"Expense-" & Trip_Name__c & "-" & ExpenseNum__c` This formula displays the text Expense- followed by trip name and the expense number. This is a text formula field that uses an expense number custom field.

ABS

Calculates the absolute value of a number. The absolute value of a number is the number without its positive or negative sign.

Use

`ABS(number)` and replace *number* with a merge field, expression, or other numeric value that has the sign you want removed.



Revenue Example `ABS(ExpectedRevenue)` calculates the positive value of the **Expected Revenue** amount regardless of whether it's positive or negative.

ADDMONTHS

Returns the date that is the indicated number of months before or after a specified date. If the specified date is the last day of the month, the resulting date is the last day of the resulting month. Otherwise, the result has the same date component as the specified date.

Use

`ADDMONTHS (date, num)` and replace *date* with the start date and *num* with the number of months to be added.



Generic Example `ADDMONTHS (StartDate, 5)` Adds 5 months to the start date. For example, if the start date is September 20, 2017, the resulting date is February 20, 2018. If the start date is September 30, 2017, the resulting date is February 28, 2018.

AND

Returns a TRUE response if all values are true; returns a FALSE response if one or more values are false.

Use the `&& (AND)` function as an alternative to the operator.

Use

`AND(logical1, logical2, ...)` and replace *logical1, logical2, ...* with the values that you want evaluated.



Formula Field Example `IF(AND(Price<1,Quantity<1),"Small", null)` This formula displays Small if the price and quantity are less than one. This field is blank if the asset has a price or quantity greater than one.

BEGINS

Determines if text begins with specific characters and returns TRUE if it does. Returns FALSE if it doesn't.

Use

`BEGINS(text, compare_text)` and replace *text*, *compare_text* with the characters or fields you want to compare.

Tips

- This function is case-sensitive so be sure your *compare_text* value has the correct capitalization.
- When using this function in a validation rule or workflow rule, fields that are blank are considered valid. For example, if you have a validation rule that tests to see if the serial number of an asset begins with "3," all assets that have a blank serial number are considered valid.



Generic Example `IF(BEGINS (Product_type_c, "ICU"), "Medical", "Technical")`

This example returns the text Medical if the text in any Product Type custom text field begins with ICU. For all other products, it displays Technical.

BLANKVALUE

Determines if an expression has a value and returns a substitute expression if it doesn't. If the expression has a value, returns the value of the expression.

Use

`BLANKVALUE(expression, substitute_expression)` and replace *expression* with the expression you want evaluated; replace *substitute_expression* with the value you want to replace any blank values.

Tips

- Use BLANKVALUE instead of NULLVALUE in new formulas. BLANKVALUE has the same functionality as NULLVALUE, but also supports text fields. Salesforce continues to support NULLVALUE, so you don't need to change existing formulas.
- A field is not empty if it contains a character, blank space, or zero. For example, a field that contains a space inserted with the spacebar is not empty.
- Use the BLANKVALUE function to return a specified string if the field doesn't have a value; use the

[ISBLANK](#) function if you only want to check if the field has a value.

- If you use this function with a numeric field, the function only returns the specified string if the field doesn't have a value and isn't configured to treat blank fields as zeroes.



Formula Example `BLANKVALUE(Department, "Undesignated")` This formula returns the value of the Department field if the Department field contains a value. If the Department field is empty, this formula returns the word Undesignated. `(BLANKVALUE(Payment_Due_Date__c, StartDate +5))` This formula returns the date five days after the contract start date whenever Payment Due Date is blank. Payment Due Date is a custom date field.

BR

Inserts a line break in a string of text.

Use

`BR()`

Tips

- Don't remove the parentheses after the function name.
- Keep the parentheses empty. They don't contain values.
- Remember to surround the BR() with concatenation operators: & or +.
- Avoid using this function in mail merge templates.
- This function isn't available in custom buttons and links, s-controls, or reports.



Formula Example

```
CASE(ShippingCountry,  
    "USA",  
        ShippingStreet & BR() &  
        ShippingCity & ",  
        " & ShippingState & " " &  
        ShippingPostalCode & BR()  
        & ShippingCountry,  
    "France",  
        ShippingStreet & BR() &  
        ShippingPostalCode & " " &  
        ShippingCity & BR() &  
        ShippingCountry, "etc")
```

This formula field displays a formatted mailing address for a contact in standard format, including spaces and line breaks where appropriate depending on the country.

CASE

Checks a given expression against a series of values. If the expression is equal to a value, returns the corresponding result. If it isn't equal to any of the values, returns the `else_result`.

Use

`CASE(expression, value1, result1, value2, result2, ..., else_result)` and replace `expression` with the field or value you want compared to each specified value. Replace each value and result with the value that must be equivalent to return the result entry. Replace `else_result` with the value you want returned when the expression doesn't equal any values.

Tips

- Be sure your `value1, value2...` expressions are the same data type.
- Be sure your `result1, result2...` expressions are the same data type.
- CASE functions can't contain functions that return true or false. Instead, make true or false expressions return numbers such as:

```
CASE(1, IF(ISPICKVAL (Term__c, "12"), 1, 0),
    12 * Monthly_Commit__c,
    IF(ISPICKVAL(Term__c, "24"), 1, 0),
    24 * Monthly_Commit__c, 0)
```

In this formula, Term is a picklist field that is multiplied by the Monthly Commit whenever it contains the value 1 for true.

- The `else_result` value is required.
- CASE functions return an error whenever any of the expressions return an error, regardless of which one must be returned. For example, `CASE(Field__c, "Partner", "P", "Customer", "C", LEFT(Field__c, -5))` returns an error even if the value of the field is "Partner" or "Customer" because the last statement is illogical.
- If the field in your CASE function is blank, it returns your `else_result` value. For example, this formula: `CASE(Days_Open__c, 3, "Reassign", 2, "Assign Task", "Maintain")` displays Maintain if the Days Open field is blank, 0, or any value other than 2 or 3.
- Use CASE functions to determine if a picklist value is equal to a particular value. For example the formula `CASE(Term__c, "12", 12 * Monthly_Commit__c, "24", 24 * Monthly_Commit__c, 0)` multiplies the Monthly Commit amount by 12 whenever the Term is 12 or multiplies the Monthly Commit amount by 24 whenever the Term is 24. Otherwise, the result is zero.

-  **Formula Field Example: Days Open for Cases** Use this example of a custom formula field called Days Open to display different text depending on the number of days a case has been open:

```
CASE(Days_Open__c, 3,
```

```
"Reassign", 2, "Assign Task", "Maintain")
```

This text is displayed.

- “Reassign” for any case open three days.
- “Assign Task” for any case open two days.
- “Maintain” for all other cases.



Formula Field Example: Last Activity Month This formula field displays the month of the last activity or None if there are no activities.

```
CASE (MONTH (LastActivityDate) ,  
1, "January",  
2, "February",  
3, "March",  
4, "April",  
5, "May",  
6, "June",  
7, "July",  
8, "August",  
9, "September",  
10, "October",  
11, "November",  
12, "December",  
"None")
```



Default Value Example: Discount Rate Use this default value formula to insert a different discount rate on an opportunity based on the department of the person creating the opportunity.

```
CASE (User.Department, "IT", 0.25, "Field", 0.15, 0)
```

In this example, the formula inserts a discount rate of 25% on any opportunity created by a user in the IT department or 15% on any opportunity created by someone in the Field department. A zero is applied if the creator doesn't belong to either of these departments. This is a custom percent field on opportunities that uses the standard user field Department.



Default Value Example: Product Language You want to associate a product with its language so that your users know the type of documentation or adapter to include. Use this default value formula to automatically set the language of a product based on the country of the user creating the product. In this example, the default value is Japanese if the user's country is Japan and English if the user's country is US. If neither is true, the default value unknown is inserted into the Product Language field.

```
CASE ($User.Country , "Japan", "Japanese", "US", "English", "unknown")
```

CASESAFEID

Converts a 15-character ID to a case-insensitive 18-character ID. In Salesforce Classic, the function converts only valid Salesforce 15-character IDs. If you pass in an invalid ID, the function returns the ID passed in. In Lightning Experience, the function converts any 15-character ID.

Use

`CASESAFEID (id)` and replace *id* with the object's ID.

Tips

- Convert to 18-character IDs for better compatibility with Excel.
- The CASESAFEID function is available everywhere that you can define a formula except reports and s-controls.
- If you're using Lightning Experience, and you want the function to convert only valid 15-character IDs, contact Salesforce Customer Support to enable this functionality.



Formula Example in Salesforce Classic

```
CASESAFEID (Id)
```

This formula replaces the 15-character ID with the 18-character, case-insensitive ID.

- `CASESAFEID ('A01xx000003DHur')` returns `A01xx000003DHur`. The ID isn't valid because it begins with a capital letter.
- `CASESAFEID ('001xx000003DHur')` returns something like `001xx000003DHurAAG` because the ID passed in is valid.

CEILING

Rounds a number up to the nearest integer, away from zero if negative.

Use

`CEILING (number)` and replace *number* with the field or expression you want rounded.



Formula Example

`CEILING (2.5)` returns 3, which is 2.5 rounded up to the nearest integer.

`CEILING (-2.5)` returns -3, which is -2.5 rounded away from zero for a negative number.

CONTAINS

Compares two arguments of text and returns TRUE if the first argument contains the second argument. If

not, returns FALSE.

Use

`CONTAINS (text, compare_text)` and replace `text` with the text that contains the value of `compare_text`.

Tips

- This function is case-sensitive so be sure your `compare_text` value has the correct capitalization.
- When using this function in a validation rule or workflow rule, fields that are blank are considered valid. For example, if you have a validation rule that tests to see if the serial number of an asset contains “A,” all assets that have a blank serial number are considered valid.
- The CONTAINS function doesn’t support multi-select picklists. Use [INCLUDES](#) to see if a multi-select picklist has a specific value.



Formula Example

```
IF (CONTAINS (Product_Type__c, "part"), "Parts", "Service")
```

This formula checks the content of a custom text field named **Product_Type** and returns `Parts` for any product with the word “part” in it. Otherwise, it returns `Service`. Note that the values are case-sensitive, so if a `Product_Type` field contains the text “Part” or “PART,” this formula returns `Services`.

CURRENCYRATE

Returns the conversion rate to the corporate currency for the given currency ISO code. If the currency is invalid, returns 1.0.

Use

`CURRENCYRATE (currency_ISO_code)` and replace `currency_ISO_code` with a currency ISO code, such as “USD”.



Function Example

`CURRENCYRATE ("USD")` returns the conversion rate to US dollars.

DATE

Returns a date value from the year, month, and day values you enter. Salesforce displays an error on the detail page if the value of the DATE function in a formula field is an invalid date, such as February 29 in a non-leap year.

Use

`DATE (year,month,day)` and replace *year* with a four-digit year, *month* with a two-digit month, and *day* with a two-digit day.

 **Formula Example** `DATE (2005, 01, 02)` creates a date field of January 2, 2005.

DATEVALUE

Returns a date value for a date/time or text expression.

As of Winter '20, the DATEVALUE() formula option provides more accurate daylight savings time values without workarounds. The option avoids an existing one-hour discrepancy when processing times between 11:00 PM and 1:00 AM. From Setup, in the Quick Find box, enter *Company Information*. Under Locale Settings, select **Improve DATEVALUE() accuracy for DST**.

 **Important** If your org's custom formulas include workarounds that adjust date values between 11:00 PM and 1:00 AM, remove them before enabling this setting. If you don't remove the workarounds, your data could be inaccurate. Enabling the preference can also increase the compiled size of existing formulas with the DATEVALUE() function.

Use

`DATEVALUE (expression)` and replace *expression* with a date/time or text value, merge field, or expression.

Tips

- If the field referenced in the function isn't a valid text or date/time field, the formula field displays #ERROR!
- When entering a date, surround the date with quotes and use this format: YYYY-MM-DD, that is, a four-digit year, two-digit month, and two-digit day.
- If the *expression* doesn't match valid date ranges, such as the MM isn't between 01 and 12, the formula field displays #ERROR!
- Dates and times are always calculated using the user's time zone, except in list views, reports, and related lists. These items calculate dates and times using Coordinated Universal Time.

 **Closed Date Example** `DATEVALUE (ClosedDate)` displays a date field based on the value of the Date/Time Closed field.

 **Date Value Example** `DATEVALUE ("2005-11-15")` returns November 15, 2005 as a date value.

DATETIMEVALUE

Returns a year, month, day, and GMT time value.

Use

`DATETIMEVALUE (expression)` and replace *expression* with a date/time or text value, merge field, or expression.

Tips

- DATETIMEVALUE is always calculated using GMT time zone and can't be changed.
- When entering a specific date, surround the date with quotes and use the following format: YYYY-MM-DD, that is, a four-digit year, two-digit month, and two-digit day.
- If the *expression* doesn't match valid date ranges, such as the MM isn't between 01 and 12, the formula field displays #ERROR!

 **Closed Date Example** `DATETIMEVALUE (ClosedDate)` displays a date field based on the value of the **Date/Time Closed** field.

 **Date Value Example** `DATETIMEVALUE ("2005-11-15 17:00:00")` returns November 15, 2005 5:00 PM GMT as a date and time value.

DAY

Returns a day of the month in the form of a number from 1 through 31.

Use

`DAY (date)` and replace *date* with a date field or value such as `TODAY()`.

 **Formula Example** `DAY (Code_Freeze__c)` returns the day in your custom code freeze date. Note this function doesn't work on date/time fields.

DISTANCE

Calculates the distance between two locations in miles or kilometers.

Use

`DISTANCE (mylocation1, mylocation2, 'unit')` and replace *mylocation1* and *mylocation2* with two location fields, or a location field and a value returned by the **GEOLOCATION** function. Replace *unit* with mi (miles) or km (kilometers).

Tips

- The DISTANCE function returns a number data type. Distance is always calculated in decimals, even if you're displaying the geolocation notation in degrees, minutes, and seconds in the user interface. Specify the number of decimal places to show when you create a custom field.
- The DISTANCE function isn't available in reports, but it can be used in list views. To use DISTANCE in your reports, set up a formula field, and then reference the field in your reports.
- DISTANCE is the only formula function that can use **GEOLOCATION** parameters.
- There are limitations on DISTANCE accuracy and equality calculations.
 - DISTANCE supports only the logical operators > and <, returning values within (<) or beyond (>) a specified radius.
 - Distance is calculated as a straight line, regardless of geography and topography between the two points.

For more details, see “[How SOQL Calculates and Compares Distances](#)” in the *SOQL and SOSL Reference*.



Example: Distance Between Two Geolocation Fields `DISTANCE(warehouse_location__c, store_location__c, 'mi')` This formula returns the distance, in miles, between the warehouse and the store. In this example, `warehouse_location__c` and `store_location__c` are the names of two custom geolocation fields.



Example: Distance Between an Address Field and a Geolocation Field

`DISTANCE(BillingAddress, store_location__c, 'mi')` This formula returns the distance, in miles, between an account's billing address and a store. In this example, `BillingAddress` is the standard billing address field on an Account object, and `store_location__c` is the name of a custom geolocation field.



Example: Distances with Conditions `IF(DISTANCE(warehouse_location__c, ShippingAddress, 'mi')<10, "Near", "Far")` This formula updates a text formula field to `Near` if the distance between the warehouse and the account shipping address compound field is less than 10 miles. Otherwise, it updates the text field to `Far`.



Tip Although DISTANCE can be calculated in miles or kilometers, the unit isn't returned in the calculation. If possible, include the unit of measure in the name of your distance formula field, so users know whether the distance is in miles or kilometers.

EXP

Returns a value for e raised to the power of a number you specify.

Use

`EXP(number)` and replace `number` with a number field or value such as 5.

 **Exponent of a Literal Value Example** `EXP(3)` This formula returns the value of e to the third power.

 **Compound Interest Example** `Principal_c * EXP(Rate_c * Years_c)` This formula calculates the compound interest based on a custom currency field for principal, custom percent field for rate, and custom number field for years.

FIND

Returns the position of a string within a string of text represented as a number.

Use

`FIND(search_text, text[, start_num])` and replace *search_text* with the string you want to find, replace *text* with the field or expression you want to search, and replace *start_num* with the number of the character from which to start searching from left to right.

Tips

- Be sure to remove the brackets, [and], from your formula before validating it.
- If the field referenced in your *text* parameter is blank, the formula field displays 0.
- Your *search_text* parameter is case-sensitive and can't contain any wildcard characters.
- If your search doesn't return any results, a 0 displays in the field.
- The *start_num* parameter is optional. If you don't enter a *start_num* value, the formula uses the value one, or the first character in the string.
- If your *start_num* isn't greater than zero, a 0 displays in the field.
- If your *start_num* is greater than the length of the text, a 0 displays in the field.
- When entering your *start_num* parameter, remember that some fields like the Website field are unique because a `http://` is automatically appended to the beginning of the text you enter.
- The first character in a string is designated as one rather than zero.

 **Street Address Example** `FIND(" ", Street)` returns the character position of the first space in the **Street** field. You can use this number to find out the length of the street address as a means of separating a street address from a street name in an address field.

 **Deriving Website Addresses Example** `SUBSTITUTE(Email, LEFT(Email, FIND("@", Email)), "www.")` finds the location of the @ sign in a person's email address to determine the length of text to replace with a "www." as a means of deriving their website address.

FLOOR

Returns a number rounded down to the nearest integer, towards zero if negative.

Use

`FLOOR (number)` and replace *number* with a number field or value such as 5.245.

-  **Example** `FLOOR (2.5)` returns 2, which is 2.5 rounded down to the nearest integer.
`FLOOR (-2.5)` returns -2, which is -2.5 rounded towards zero for a negative number.

GEOLOCATION

Returns a geolocation based on the provided latitude and longitude. Must be used with the DISTANCE function.

Use

`GEOLOCATION (latitude, longitude)` and replace *latitude* and *longitude* with the corresponding geolocation, numerical code values.

Tips

- The GEOLOCATION function returns a location data type that can be used only by, and must be used with, the DISTANCE function. The GEOLOCATION function doesn't work on its own.

-  **Example: Distance Between a Custom Geolocation Field and Fixed Coordinates**

`DISTANCE (warehouse_location__c, GEOLOCATION (37.775, -122.418), 'km')` This formula returns the distance, in kilometers, between the warehouse and the known latitude and longitude 37.775°, -122.418° (San Francisco).

GETRECORDIDS

Returns an array of strings in the form of record IDs for the selected records in a list, such as a list view or related list.

Use

`{ !GETRECORDIDS (object_type) }` and replace *object_type* with a reference to the custom or standard object for the records you want to retrieve.

Tips

- Use global variables to access special merge fields for s-controls, custom buttons, and links.
- Activities are special types of objects. Use `{!GETRECORDIDS($ObjectType.Task)}` when creating a task list button. Use `{!GETRECORDIDS($ObjectType.Event)}` when creating an event list button.

- This function is only available in custom buttons, links, and s-controls.



Custom Button Example

```
{!REQUIRESCRIPT (" /soap/ajax/13.0/connection.js" ) }
var records = {!GETRECORDIDS($ObjectType.Sample) };
var newRecords = [];
if (records[0] == null) {
    alert("Please select at least one row")
} else {
    for (var n=0; n<records.length; n++) {
        var c = new sforce.SObject("Case");
        c.id = records[n];
        c.Status = "New";
        newRecords.push(c);
    }
    result = sforce.connection.update(newRecords);
    window.location.reload();
}
```

In this example, all selected case records are updated with a Status of New. To set up this code in your org, create a custom list button for cases with these attributes.

- **Display Type** is List Button
- **Behavior** is Execute JavaScript
- **Content Source** is OnClick JavaScript

Paste this sample code into the content of your custom button. Finally, add the list button to the page layout that contains the Cases related list, such as accounts or opportunities. Users can select any number of cases in the related list and click the list button to change the status of those cases at once. Notice the check for `records[0] == null`, which displays a message to users when they don't select at least one record in the list.

GETSESSIONID

Returns the user's session ID.

Use

`GETSESSIONID()`

Tips

Important `$Api.Session_ID` and `GETSESSIONID()` return the same value, an identifier for the

current session in the current context. This context varies depending on where the global variable or function is evaluated. For example, if you use either in a custom formula field, and that field is displayed on a standard page layout in Salesforce Classic, the referenced session is a basic Salesforce session. That same field (or the underlying variable or formula result), when used in a Visualforce page, references a Visualforce session instead. Session contexts are based on the domain of the request. That is, the session context changes whenever you cross a hostname boundary, such as from `.salesforce.com` to `.vf.force.com` or `.lightning.force.com`. Session identifiers from different contexts, and the sessions themselves, are different. When you transition between contexts, the old session is replaced by the new one, and the old session is no longer valid. The session ID also changes at this time. Normally Salesforce transparently handles session hand-off between contexts, but if you're passing the session ID around yourself, you must reaccess `$Api.Session_ID` or `GETSESSIONID()` from the new context to ensure a valid session ID. Not all sessions are created equal. In particular, sessions obtained in a Lightning Experience context have reduced privileges, and don't have API access. You can't use these session IDs to make API calls. `{ !$Api.Session_ID }` isn't generated for guest users. If you use a JWT-based access token for session authentication, you can't use `$Api.Session_ID` or `GETSESSIONID()`. To use `$Api.Session_ID` or `GETSESSIONID()`, use an opaque access token instead. Make sure that the "Issue JSON Web Token (JWT)-based access tokens for named users" setting isn't selected for your external client app or connected app.



Link Example

```
HYPERLINK  
("https://www.myintegration.com?sid="&  
GETSESSIONID() & "?&rowID=&Name & "action=CreateTask", "Create  
a Meeting Request")
```

creates a link to an application outside of Salesforce, passing the parameters so that it can connect to Salesforce via the API and create the necessary event.

HOUR

Returns the local time hour value without the date in the form of a number from 1 through 24.

Use

`HOUR(time)` and replace *time* with a time value or value such as `TIMENOW()`.



Example `HOUR(TIMEVALUE(ClosedDate))` displays only the hour in a time field based on the value of the Time Closed field. `HOUR(TIMEVALUE("17:30:45.125"))` returns 17.

HTMLENCODE

Encodes text and merge field values for use in HTML by replacing characters that are reserved in HTML,

such as the greater-than sign (>), with HTML entity equivalents, such as `>`.

Use

`{ !HTMLENCODE (text) }` and replace *text* with the merge field or text string that contains the reserved characters.

Tips

This function is only available in custom buttons and links, and in Visualforce.

 **Example** If the merge field `foo_c` contains `Enter the user's name`, `{ !HTMLENCODE (foo_c) }` results in: `Enter the user's name`

HYPERLINK

Creates a link to a URL specified that is linkable from the text specified.

Use

`HYPERLINK(url, friendly_name [,target])` and replace *url* with the Web address, replace *friendly_name* with the link text, and, optionally, replace *target* with the window or frame in which to display the content.

Tips

- Hyperlink formula fields are of type text.
- Include the protocol and URL in quotes as in `HYPERLINK("http://www.cnet.com", "cnet")`.
- Avoid using text functions such as LEN, LEFT, or RIGHT on HYPERLINK function results.
- The URL can't contain JavaScript. This increases security for your org. Using JavaScript is permitted in packages, sandbox copies, and change sets.
- Use a relative link to link to Salesforce pages. If your full link is `https://yourInstance.salesforce.com/00U/e`, then its relative link is `/00U/e`. Relative links allow the hyperlink to work correctly on all Salesforce pages. Use the relative URL in a hyperlink formula to add it to a search layout. Make sure to prepend your relative URL with a forward slash “`/`”.
- Use the `$Api` variable to reference API URLs.
- Be sure to remove the brackets, [and], from your formula before validating it.
- The *target* parameter is optional. If you don't specify a *target*, the link opens in a new browser window. Some common *target* parameters are:
 - `_blank`–Displays link in a new unnamed window.
 - `_self`–Displays link in the same frame or window as the element that refers to it.
 - `_parent`–Displays link in the immediate frameset parent of the current frame. This value is the same as `_self` if the current frame has no parent.

- _top—Displays link in the full original window, canceling any other frames. This value is the same as _self if the current frame has no parent.

For more information on basic HTML tags, consult an HTML reference on the Internet.

- In Chatter feed links created with a hyperlink formula, the *target* parameter doesn't render and defaults to _blank.
- The HYPERLINK function is available everywhere that you can define a formula except default values, field updates, s-controls, validation rules, approval processes, custom buttons and links, and workflow rules.



Create Events Example

```
HYPERLINK("/00U/e?
retURL=%2F006x0000001T8Om&what_id="
& Id,
"Create Event")
```

adds a link called Create Event that, when clicked, creates an event that is associated with the current object.



Phone Dialer Example `HYPERLINK("http://servername/call?id=" & Id & "&phone=" & Phone, Phone)` creates a linkable phone number field that automatically dials the phone number when clicked. In this example, replace "servername" and "call" with the name of your dialing tool and the command it uses to dial. The merge field, Id, inserts the identifier for the contact, lead, or account record. The first Phone merge field tells the dialing tool what number to call and the last Phone merge field uses the value of the Phone field as the linkable text the user clicks to dial.

IF

Determines if expressions are true or false. Returns a given value if true and another value if false.

Use

`IF(logical_test, value_if_true, value_if_false)` and replace *logical_test* with the expression you want evaluated; replace *value_if_true* with the value you want returned if the expression is true; replace *value_if_false* with the value you want returned if the expression is false.

Tips

- Make sure your *value_if_true* and *value_if_false* expressions are the same data type.
- When using an IF function with the \$Profile.UserType variable to determine the type of Salesforce user license the logged in user has, use the following values:
 - Standard for Salesforce
 - PowerPartner for PRM User
 - CustomerSuccess for Customer Portal User

- PowerCustomerSuccess for Customer Portal Manager

For example, use the following formulas to determine if the logged in user has the license type in quotes:

```
IF(ISPICKVAL($Profile.UserType , "Standard"), 100, 0.1)
```

```
IF(ISPICKVAL($Profile.UserType , "PowerPartner"), 100, 0.1)
```

```
IF(ISPICKVAL($Profile.UserType , "CustomerSuccess"), 100, 0.1)
```

 **Note** \$Profile merge fields are only available in Enterprise, Unlimited, Performance, and Developer Editions.

 **Formula Field Example: Overdue Payments** `IF(AND(Payment_Due_Date__c < TODAY(), Payment_Status__c = "UNPAID") , "PAYMENT OVERDUE", null)` This formula determines if the payment due date is past and the payment status is “UNPAID.” If so, returns the text “PAYMENT OVERDUE” and if not, leaves the field blank. This example uses a custom date field called **Payment Due Date** and a text custom field called **Payment Status**.

 **Formula Field Example: Insert Tax Rate** Use this default value formula to set the tax rate of an asset based on the user's city. Create a custom percent field with the following default value:

```
IF($User.City = "Napa", 0.0750,
    IF($User.City = "Paso Robles", 0.0725,
        IF($User.City = "Sutter Creek", 0.0725,
            IF($User.City = "Los Olivos", 0.0750,
                IF($User.City = "Livermore", 0.0875, null
                    )
                )
            )
        )
    )
```

 **Custom Button Example**

```
{!
IF(Sample.BillingCountry = "US",
"http://maps.google.com/maps?q=&Sample.BillingStreet&
+"&Sample.BillingCity+"&"&Sample.BillingState+"&"&Sample.BillingCountry,
(IF(Sample.BillingCountry = "UK",
"http://maps.google.co.uk/maps?q=&Sample.BillingStreet
+"&"&Sample.BillingCity+"&"&Sample.BillingCountry,
"http://maps.google.com")))
)}
```

```
}
```

This example uses the IF function to determine if an address is in the United States or United Kingdom so that it can use the appropriate type of Google map to display the address.

IMAGE

Inserts an image with alternate text and height and width specifications.

Use

`IMAGE(image_url, alternate_text, height, width)` and replace `image_url` with the full path to the image. Replace `alternate_text` with the string of text you want to appear when the image can't be rendered for some reason. This text can be used by screen reader software. Replace `height` with the vertical size of the image in pixels. Replace `width` with the horizontal size of the image in pixels.

For reports, images aren't automatically resized to fit into a report column. Use the `height` and `width` parameters to explicitly size the image so it fits into the column without being partially cut off.

Tips

- The `height` and `width` parameters are optional.
- Use a text string to replace the `image_url` and `alternate_text` parameters. Surround each text string in quotes.
- Use numbers to replace the `height` and `width` parameters.
- Add images to your Documents tab if you want to display them elsewhere. For example, store the image of a product in a document folder, copy the URL to the document, and paste that URL in the `image_url` parameter of a formula field on the Products tab.
- If you use Internet Explorer, you sometimes must change your security settings so that Explorer doesn't display a warning prompt when images use HTTP protocol. See the online help for Internet Explorer for instructions on changing your security settings.
- The IMAGE function cannot include the GETSESSIONID function as one of its arguments.
- The IMAGE function is available only in formula fields and email templates.
- You can't display an image related to a contact in a custom formula field if it's referenced through a person account.



Example `HYPERLINK("ymsgr:sendIM?" & Yahoo_Name__c,`
`IMAGE("http://opi.yahoo.com/online?u=" & Yahoo_Name__c & "&m;=g&t;=0",`
`"Yahoo"))` This formula displays a clickable Yahoo! Messenger icon indicating if the person is logged on to the service. Users can click the icon to launch a Yahoo! Messenger conversation with the person. This example uses a custom text field called **Yahoo Name** on contacts where you can store the contact's Yahoo! Messenger ID.

IMAGEPROXYURL

Securely retrieves external images and prevents unauthorized requests for user credentials.

Use

```
<apex:image value="{ ! IMAGEPROXYURL ("http://exampledomain.com/pic.png") }" />
```

 and replace `http://exampledomain.com/pic.png` with your image.

Tips

- Use IMAGEPROXYURL for all images hosted on servers you don't control.
- The rendered image URL can change at any time. Don't copy and paste it anywhere.
- Don't use the rendered image URL outside of Salesforce.

Function Example

```
<apex:image id="salesforce-twitter"
value="{ ! IMAGEPROXYURL ("https://pbs.twimg.com/profile_images/1014182734606
897153/JfveQU3Z_400x400.jpg") }"
alt="Salesforce on Twitter" />
```

This IMAGEPROXYURL function retrieves and displays an image from Twitter's image host, an external source. This function loads the Salesforce Twitter profile image over HTTPS. This function also prevents the image from making unauthorized requests for user credentials.

INCLUDE

Returns content from an s-control snippet. Use this function to reuse common code in many s-controls.

Use

```
{ ! INCLUDE (source, [inputs]) }
```

 and replace `source` with the s-control snippet you want to reference. Replace `inputs` with any information you need to pass the snippet.

Tips

- Because this function references an s-control snippet and does not copy it, it always runs the latest content of the s-control snippet. Remember that changing your s-control snippet affects all INCLUDE functions that refer to it.
- Use the \$Request global variable to access any information inside the snippet.
- This function is only available in custom buttons, links, and s-controls.



S-Control Example: Include a Header Snippet

```
<html>
  <body>
    { !INCLUDE (
      $SControl.Header_Snippet,
      [title = "My Title", theme = "modern"]
    ) }
  </body>
</html>
```

This example references a snippet that provides a header for a page that you created to display in a web tab. It displays the page title “My Title.” Use the \$SControl global variable to reference a custom s-control.



S-Control Example: Include Input Parameters

Use the following two examples to see how you can create a reusable snippet and include it in an s-control.

```
<h2 class="{$Request.titleTheme}.title">
  {$Request.titleText}
</h2>
```

This snippet requires two input parameters: *titleTheme* and *titleText*. It is a reusable HTML tag that presents a page title and theme based on input parameters. Next, create an s-control that includes this snippet:

```
<html>
  <head/>
  <body>
    { !INCLUDE (
      $SControl.Title_Snippet,
      [titleTheme = "modern", titleText = "My Sample Title"]
    ) }
    Insert your page-specific content here ...
  </body>
</html>
```

This s-control uses the snippet titled *Title_Snippet* to display the title of the page “My Sample Title” and modern theme. Replace *Insert your page specific content here* with your own HTML content and use the s-control as the source of a web tab to create your own pages in Salesforce.

INCLUDES

Determines if any value selected in a multi-select picklist field equals a text literal you specify.

Use

`INCLUDES(multiselect_picklist_field, text_literal)` and replace `multiselect_picklist_field` with the merge field name for the multi-select picklist; and replace `text_literal` with the multi-select picklist value you want to match in quotes.

Tips

- The `text_literal` expression must be of type text and enclosed in quotes. It cannot be a merge field or the result of a function.
- Salesforce returns an error if any of the following occurs:
 - You do not provide a `text_literal` expression.
 - You provide an empty `text_literal` expression, such as `" "` or `" " "`.
- Use `ISBLANK` to determine if a multi-select picklist field is empty.
- Use the `PRIORVALUE` function inside the `INCLUDES` function to check if the previous value of a multi-select picklist field included a specific value. For example:

```
INCLUDES(  
    PRIORVALUE(multiselect_picklist_field),  
    text_literal  
)
```

 **Function Example** `INCLUDES(Hobbies__c, "Golf")` returns TRUE if one of the selected values in the **Hobbies** custom multi-select picklist field is Golf.

ISBLANK

Determines if an expression has a value and returns TRUE if it does not. If it contains a value, this function returns FALSE.

Use

`ISBLANK(expression)` and replace `expression` with the expression you want evaluated.

Tips

- Use `ISBLANK` instead of `ISNULL` in new formulas. `ISBLANK` has the same functionality as `ISNULL`, but also supports text fields. Salesforce continues to support `ISNULL`, so you do not need to change any existing formulas.
- A field is not empty if it contains a character, blank space, or zero. For example, a field that contains a space inserted with the spacebar is not empty.
- Use the `BLANKVALUE` function to return a specified string if the field doesn't have a value; use the `ISBLANK` function if you only want to check if the field has a value.

- If you use this function with a numeric field, the function only returns TRUE if the field has no value and is not configured to treat blank fields as zeroes.
- If you use this function with a picklist, use ISBLANK(TEXT(<picklist>)) to convert the picklist items into a text value.



Formula Example

```
(IF(ISBLANK(Maint_Amount__c), 0, 1) +  
 IF(ISBLANK(Services_Amount__c), 0, 1) +  
 IF(ISBLANK(Discount_Percent__c), 0, 1) +  
 IF(ISBLANK(Amount), 0, 1) +  
 IF(ISBLANK(Timeline__c), 0, 1)) / 5
```

This formula takes a group of fields and calculates what percent of them are being used by your personnel. This formula field checks five fields to see if they are blank. If so, a zero is counted for that field. A “1” is counted for any field that contains a value, and this total is divided by five (the number of fields evaluated). This formula requires you to select the **Treat blank fields as blanks** option under Blank Field Handling while the Advanced Formula subtab is showing.

ISCHANGED

Compares the value of a field to the previous value and returns TRUE if the values are different. If the values are the same, this function returns FALSE.

Use

`ISCHANGED (field)` and replace *field* with the name of the field you want to compare.

Tips

- This function is available only in:
 - Assignment rules
 - Validation rules
 - Field updates
 - Workflow rules if the evaluation criteria is set to **Evaluate the rule when a record is: created, and every time it's edited**.
 - Formula criteria for executing actions in Process Builder.
- Use the NOT function to reverse the return values of TRUE and FALSE.
- This function returns FALSE when evaluating any field on a newly created record.
- If a text field was previously blank, this function returns TRUE when it contains any value.
- For number, percent, or currency fields, this function returns TRUE when:
 - The field was blank and now contains any value
 - The field was zero and now is blank
 - The field was zero and now contains any other value

 **Validation Rule Example** The following validation rule prevents users from changing an object name after it has been created: `ISCHANGED (Name) . NOT (AND (ISCHANGED (Priority), ISPICKVAL (Priority, "Low")))` is a validation rule that ensures if a user changes the Priority of a case, the new priority cannot be "Low." `NOT (AND (ISCHANGED (CloseDate), OR (MONTH (CloseDate) <> MONTH (TODAY ()), YEAR (CloseDate) <> YEAR (TODAY ())), $Profile.Name <> "Sales Manager"))` is a validation rule that prevents a user from changing the Close Date of an opportunity to a date outside of the current month and year unless that user has the Sales Manager profile.

 **Note** \$Profile merge fields are only available in Enterprise, Unlimited, Performance, and Developer Editions.

ISCLONE

Checks if the record is a clone of another record and returns TRUE if one item is a clone. Otherwise, returns FALSE.

Use

`ISCLONE ()`

Tips

- This function cannot be used with fields.
- Use the NOT function to reverse the return values of TRUE and FALSE.

 **Validation Rule Example** Use `(ISCLONE ())` to create a validation rule on an object and identify a record that's a clone of another record.

ISNEW

Checks if the formula is running during the creation of a new record and returns TRUE if it is. If an existing record is being updated, this function returns FALSE.

Use

`ISNEW ()`

Tips

- This function is available only in validation rules, field updates, workflow rules, assignment rules, and processes.

- Use the NOT function to reverse the return values of TRUE and FALSE.
- This function always returns FALSE when used in a workflow rule with a time-based trigger.
- This function always returns FALSE when used in a field update for an approval action.
- This function is unsupported for Data Cloud objects.



Validation Rule Example Use the following validation rule to prevent users from creating a record with a close date in the past. `AND (ISNEW(), CloseDate < TODAY())` checks if the user is creating an opportunity and, if so, ensures that the **Close Date** is today or after today. Use this validation rule to ensure users add at least one product to an opportunity after they have created it.

```
NOT (OR (ISNEW(), HasOpportunityLineItem))
```

In this example, the validation rule formula displays this error message when an existing opportunity does not have any products: "You must add products to this opportunity before saving." This formula doesn't display an error on the initial save because they cannot add products until after saving the record initially; but it prevents them from resaving or closing an opportunity that does not contain products.

ISNULL

Determines if an expression is null (blank) and returns TRUE if it is. If it contains a value, this function returns FALSE.

Use

`ISNULL (expression)` and replace *expression* with the expression you want evaluated.

Tips

- Text fields are never null, so using this function with a text field always returns false. For example, the formula field `IF (ISNULL (new__c) 1, 0)` is always zero regardless of the value in the **New** field. For text fields, use the ISBLANK function instead.
- Multi-select picklist fields are never null in s-controls, buttons, and email templates, so using this function with a multi-select picklist field in those contexts always returns false.
- Don't use ISNULL functions in date/time fields. Empty date and date/time fields always return true when referenced in ISNULL functions.
- Don't use ISNULL functions in text and lookup fields. Empty text and lookup fields always return false when referenced in ISNULL functions. Use the ISBLANK function instead.
- Choose **Treat blank fields as blanks** for your formula when referencing a number, percent, or currency field in an ISNULL function. Choosing **Treat blank fields as zeroes** gives blank fields the value of zero so none of them are null.
- Merge fields can be handled as blanks, which can affect the results of components like s-controls because they can call this function.
- When using a validation rule to ensure that a number field contains a specific value, use the ISNULL

function to include fields that do not contain any value. For example, to validate that a custom field contains a value of '1', use the following validation rule to display an error if the field is blank or any other number:

```
OR(ISNULL(field__c), field__c<>1)
```

Formula Example

```
(IF(ISNULL(Maint_Amount__c), 0, 1) +
 IF(ISNULL(Services_Amount__c), 0, 1) +
 IF(ISNULL(Discount_Percent__c), 0, 1) +
 IF(ISNULL(Amount), 0, 1) +
 IF(ISNULL(Timeline__c), 0, 1)) / 5
```

This formula takes a group of fields and calculates what percent of them are being used by your personnel. This formula field checks five fields to see if they are blank. If so, a zero is counted for that field. A "1" is counted for any field that contains a value, and this total is divided by five (the number of fields evaluated). This formula requires you to select the **Treat blank fields as blanks** option under Blank Field Handling while the Advanced Formula subtab is showing.

Validation Rule Example

```
AND(ISPICKVAL(StageName, "Closed Won"),
 ISNULL(Project_Start_Date__c))
```

This validation rule makes the **Project Start Date** custom date field conditionally required whenever the stage is Closed Won.

ISNUMBER

Determines if a text value is a number and returns TRUE if it is. Otherwise, returns FALSE.

Use

`ISNUMBER(text)` and replace *text* with the merge field name for the text field.

Tips

- This function returns FALSE for blank values.
- The ISNUMBER function is not aware of your locale. For example, `ISNUMBER("123,12")` and `ISNUMBER("1 000")` return FALSE even if the user's locale is "French."
- Chinese, Japanese, Korean, and special characters including a space return FALSE.
- The ISNUMBER function returns TRUE for scientific formatting, such as "2E2" or "123.123."

Validation Rule Example

```
OR(LEN(Bank_Account_Number__c) <> 10, NOT(ISNUMBER(Bank_Account_Number__c)))
```

This validation rule ensures a custom text field called **Bank Account Number** is a number of 10 digits and is not blank.

ISPICKVAL

Determines if the value of a picklist field is equal to a text literal you specify.

Use

`ISPICKVAL(picklist_field, text_literal)` and replace *picklist_field* with the merge field name for the picklist; replace *text_literal* with the picklist value in quotes. *text_literal* cannot be a merge field or the result of a function.

Tips

- Replace *picklist_field* with a custom or standard field of type picklist.
- Your *text_literal* expression must be of type text and enclosed in quotes. It cannot be a merge field or the result of a function.
- Use [CASE](#) functions to determine if a picklist value is equal to a particular value.
- When using the ISPICKVAL function to return the previous value of a picklist field, include the PRIORVALUE function inside the ISPICKVAL function as in this example:

```
ISPICKVAL(PRIORVALUE
(picklist_field),
text_literal)
```

Contract Activation Example `IF(ISPICKVAL(Status, "Activated"), NOW() - ActivatedDate, null)`

calculates the number of days since the contract was activated. If the contract status is not “Activated,” this field is blank.

Commission Accounts Example

```
IF(ISPICKVAL(StageName, "Closed Won"),
ROUND(Amount *0.02, 2), 0)
```

This example calculates the commission amount for any opportunity that has a “Closed Won” stage. The value of this field is the amount times 0.02 for any closed/won opportunity. Open or lost opportunities have a zero commission value.

Competitor-Triggered Workflow Example

```
ISPICKVAL(Stage, "Closed Lost") && INCLUDES(Competitor__c, "Acme")
```

In a workflow rule or process, this formula configures Salesforce to trigger the associated actions if the **Competitor** multi-select picklist field on a lost business is Acme.

JSENCODE

Encodes text and merge field values for use in JavaScript by inserting escape characters, such as a backslash (\), before unsafe JavaScript characters, such as the apostrophe (').

Use

`{ !JSENCODE (text) }` and replace *text* with the merge field or text string that contains the unsafe JavaScript characters.

Tips

This function is only available in custom buttons and links, and in Visualforce.

 **Merge Field Example** If the merge field `foo__c` contains `Enter the user's name`, `{ !JSENCODE (foo__c) }` results in: `\u003CB\u003EEnter the user\'s name\u003C\B\u003E`

JSINHTMLENCODE

Encodes text and merge field values for use in JavaScript inside HTML tags by replacing characters that are reserved in HTML with HTML entity equivalents and inserting escape characters before unsafe JavaScript characters.

`JSINHTMLENCODE (someValue)` is a convenience function that is equivalent to `JSENCODE (HTMLENCODE (someValue))`. That is, `JSINHTMLENCODE` first encodes *someValue* with `HTMLENCODE`, and then encodes the result with `JSENCODE`.

Use

`{ !JSINHTMLENCODE (text) }` and replace *text* with the merge field or text string that contains the unsafe JavaScript characters.

Tips

- This function is only available in custom buttons and links, and in Visualforce.

 **Merge Field Example** If the merge field `foo__c` contains `Enter the user's name`, `{!JSINHTMLENCODE(foo__c)}` results in: `Enter the user's name`

JUNCTIONIDLIST

Returns a JunctionIDList based on the provided IDs.

A JunctionIDList is a string array of referenced ID values that represent the many-to-many relationship of an underlying junction entity.

Use

`JUNCTIONIDLIST(id, id, ...)` and replace `id` with the Salesforce ID you want to use.

 **Contact Record ID Example** `JUNCTIONIDLIST(Case.ContactId)` This formula returns the case's contact record ID. When used on the email action for cases, you can use this formula as a predefined value for the To Recipients field. Using this formula as a predefined value for the field ensures that sent emails are always associated with a Salesforce record. In the case feed publisher, users see the contact name instead of the ID or email address.

LEFT

Returns the specified number of characters from the beginning of a text string.

Use

`LEFT(text, num_chars)` and replace `text` with the field or expression you want returned; replace `num_chars` with the number of characters from the left you want returned.

Tips

- Reference auto-number fields as text fields in formulas.
- If the `num_chars` value is less than zero, Salesforce replaces the value with zero.

 **Custom Field Example** `TRIM(LEFT(LastName, 5)) & "-" & TRIM(RIGHT(SSN__c, 4))` This formula displays the first five characters of a name and the last four characters of a social security number separated by a dash. Note that this example uses a text custom field called **SSN**.

LEN

Returns the number of characters in a specified text string.

Use

`LEN(text)` and replace *text* with the field or expression whose length you want returned.

Formula Example

```
LEN(PartNumber__c)
```

This formula returns the number of characters in a **Product Code** field.

LINKTO

Returns a relative URL in the form of a link (href and anchor tags) for a custom s-control or Salesforce page.

Use

`{!LINKTO(label, target, id, [inputs], [no override])}` and replace *label* with the text for the link, *target* with the URL, and *id* with a reference to the record. Inputs are optional and can include any additional parameters you want to add to the link. The *no override* argument is also optional and defaults to “false.” It applies to targets for standard Salesforce pages such as \$Action.Account.New. Replace *no override* with “true” when you want to display a standard Salesforce page regardless of whether you have defined an override for it elsewhere.

Tips

- Avoid using this function in an inline s-control if you want it to open in a new window.
- Enclose multiple *inputs* in brackets to indicate they are together:

```
{!LINKTO("View Case", $Action.Case.View, Case.Id, [parm1="A", parm2="B"] )}
```

- Set *inputs* to null if you do not have any to pass yet you want to set the *no override* argument:

```
{!LINKTO("View Case", $Action.Case.View, Case.Id, null, true)}
```

- When you override the tab home page for a standard or custom tab, use the tab’s **\$Action** global variable as the *target* value, and the tab’s object type for the *id* value. For example,
`LINKTO("Accounts Tab", $Action.Account.Tab, $ObjectType.Account)`
- This function is only available in custom buttons, links, and s-controls.

New Account S-Control Example

```
<html>
<body>
```

```
{!LINKTO(
    "Create a New Account",
    $Action.Account.New,
    $ObjectType.Account
)
</body>
</html>
```

This example allows users to click a link to create an account. It is useful in account list views or Web tabs where you want users to create an account directly from that page. Use the \$Action global variable to access the new account page in Salesforce.



Email Window S-Control Example

```
<html>
<body>
{!LINKTO("Email link",
"mailto:support@yourcompany.com?subject=Please%20Help")};
</body>
</html>
```

This example launches a new email window addressed to support@yourcompany.com with the subject “Please Help” whenever a user clicks “Mail link.”



Link to Another S-Control Example

```
<html>
<body>
{!LINKTO("Check for duplicates",
$Scontrol.dedup_account, Account.Id)};
</body>
</html>
```

Use this example to generate a page containing a hyperlink labeled “Check for duplicates.” When users click this link, Salesforce runs your custom s-control. This example assumes you have already created a custom s-control to find duplicate accounts and merge their information.

LN

Returns the natural logarithm of a specified number. Natural logarithms are based on the constant e value of 2.71828182845904.

Use

`LN (number)` and replace *number* with the field or expression for which you want the natural logarithm.

Note: the LN function is the inverse of the EXP function.

-  **Formula Example** `LN(10)` returns the natural logarithm of 10, which is 2.30. `LN(Value__c)` returns the natural logarithm of a custom number field called **Value**.

LOG

Returns the base 10 logarithm of a number.

Use

`LOG(number)` and replace *number* with the field or expression from which you want the base 10 logarithm calculated.

-  **Salary Example** `LOG(Salary__c)` calculates the logarithm of a person's salary. In this example, **Salary** is a custom currency field.

-  **Hydrogen Example** `-LOG(Hydrogen__c)` calculates the pH and acidity using the LOG function and a custom number field called **Hydrogen**, which represents the concentration of Hydrogen ions in the liquid measured in moles per liter.

LOWER

Converts all letters in the specified text string to lowercase. Any characters that are not letters are unaffected by this function. Locale rules are applied if a locale is provided.

Use

`LOWER(text, [locale])` and replace *text* with the field or text you wish to convert to lowercase, and *locale* with the optional two-character ISO language code or five-character locale code, if available.

-  **MYCOMPANY.COM Example** `LOWER("MYCOMPANY.COM")` returns "mycompany.com."

-  **Ticker Symbol Example** `LOWER(TickerSymbol)` returns the text in **Ticker Symbol** in lower case characters.

-  **Turkish Language Locale Rules Example** The Turkish language has two versions of the letter "i": one dotted and one dotless. The locale rules for Turkish require the ability to capitalize the dotted i, and allow the dotless I to be lowercase. To correctly use the `LOWER()` function with the Turkish language locale, use the Turkish locale code `tr` in the `LOWER()` function as follows:
`LOWER(text, "tr")` Using this locale code ensures that Salesforce doesn't transform any dotted i in the *text* to a dotless I.

LPAD

Inserts characters you specify to the left-side of a text string.

Use

`LPAD(text, padded_length[, pad_string])` and replace the variables:

- *text* is the field or expression you want to insert characters to the left of.
- *padded_length* is the number of total characters in the text that's returned.
- *pad_string* is the character or characters that are inserted. *pad_string* is optional and defaults to a blank space.

If the value in *text* is longer than *pad_string*, *text* is truncated to the size of *padded_length*.

Tips

Leading blank spaces and zeros are omitted.

 **Field Name Example: Padding** `LPAD(Name, 20)` truncates the Name field after 20 characters.

For example, if the name is `mycompany.com`, the value returned is "mycompany.com."

 **My Company Example: No Change** `LPAD('my_company.com', 14, 'z')` returns

"my_company.com" without change because it has 14 characters.

 **Field Name Example: Padded with Z** `LPAD(Name, 15, 'z')` returns the name

"zmycompany.com."

 **Field Name Example: Truncating Ex** `LPAD(Name, 2)` truncates the name after the second

character. For example, if the name is `mycompany.com`, the value returned is "my."

MAX

Returns the highest number from a list of numbers.

Use

`MAX(number, number, ...)` and replace *number* with the fields or expressions from which you want to retrieve the highest number.

 **Service Charge** `MAX(0.06 * Total_Cost__c, Min_Service_Charge__c)` In this example, the formula field calculates a service charge of 6% of the total cost or a minimum service charge, whichever is greater. Note that **Min Service Charge** is a custom currency field with a **default value** of \$15. However, you could make it a formula field if your minimum service charge is always the same

amount.



Book Royalties Example

```
MAX(0.10 * Pages__c,  
    (Retail_Price__c * 0.07) * Total_Sold__c)
```

This formula determines which amount to pay in royalties for a book. It displays the greater of two amounts: \$0.07 for each book sold or \$0.10 per page. It assumes you have custom number fields for **Pages** and **Total Sold** and a custom currency field for **Retail Price**.



Commissions Example

```
MAX($User.Commission_Percent__c * Price,  
    Price * Account_Discount__c, 100)
```

This formula determines what commission to log for an asset based on which is greater: the user's commission percentage of the price, the price times the discount percent stored for the account or 100 dollars. This example assumes you have two custom percent fields on users and assets.

MCEILING

Rounds a number up to the nearest integer, towards zero if negative.

Use

`MCEILING (number)`



Rounding Example `MCEILING (2.5)` returns 3, which is 2.5 rounded up to the nearest integer.
`MCEILING (-2.5)` returns -2, which is -2.5 rounded up towards zero for a negative number.

MFLOOR

Rounds a number down to the nearest integer, away from zero if negative.

Use

`MFLOOR (number)`



Rounding Example `MFLOOR (2.5)` returns 2, which is 2.5 rounded down to the nearest integer.
`MFLOOR (-2.5)` returns -3, which is -2.5 rounded away from zero for a negative number.

MID

Returns the specified number of characters from the middle of a text string given the starting position.

Use

`MID(text, start_num, num_chars)` and replace *text* with the field or expression to use when returning characters; replace *start_num* with the number of characters from the left to use as a starting position; replace *num_chars* with the total number of characters to return.

 **String Example** `MID(Division, 3, 4)` returns four characters of the **Division** name beginning with the third character from the left. On a user record, this represents the department code.

MILLISECOND

Returns a milliseconds value in the form of a number from 0 through 999.

Use

`MILLISECOND(time)` and replace *time* with a time value or value such as `TIMENOW()`.

 **Formula Example** `MILLISECOND(TIMEVALUE(ClosedDate))` displays only the milliseconds in a time field based on the value of the Time Closed field.

`MILLISECOND(TIMEVALUE("17:30:45.125"))` returns 125.

MIN

Returns the lowest number from a list of numbers.

Use

`MIN(number, number, ...)` and replace *number* with the fields or expressions from which you want to retrieve the lowest number.

 **401k Matching Example**

```
MIN(250, Contribution__c /2)
```

This example formula determines which amount to provide in employee 401K matching based on a matching program of half of the employee's contribution or \$250, whichever is less. It assumes you have a custom currency field for **Contribution**.

 **Bonus Example**

```
MIN(Gross__c * Bonus_Percent__c,
    Performance__c / Number_of_Employees__c)
```

This example determines an employee's bonus amount based on the smallest of two amounts: the employee's gross times bonus percent or an equally divided amount of the company's performance amount among all employees. It assumes you have a custom number field for **Number of Employees**, a custom percent field for **Bonus Percent**, and currency custom fields for the employee's **Gross** and company's **Performance**.

MINUTE

Returns a minute value in the form of a number from 0 through 60.

Use

`MINUTE(time)` and replace *time* with a time value or value such as `TIMENOW()`.

 **Formula Example** `MINUTE(TIMEVALUE(ClosedDate))` displays only the minutes in a time field based on the value of the Time Closed field. `MINUTE(TIMEVALUE("17:30:45.125"))` returns 30.

MOD

Returns a remainder after a number is divided by a specified divisor.

Use

`MOD(number, divisor)` and replace *number* with the field or expression you want divided; replace *divisor* with the number to use as the divisor.

 **Scheduling Example** `MOD(3, 3)` returns 0 `MOD(4, 3)` returns 1 `MOD(123, 100)` returns 23 To prevent users from scheduling meetings on a Saturday or Sunday, use a validation rule to apply a custom date field called **My Date**.

```
CASE(MOD(My_Date__c - DATE(1900, 1, 7), 7),
0, 0,
6, 0,
1) = 0
```

This example displays the following error message when the value of **My Date** is not Monday through Friday: "My Date is not a weekday."

MONTH

Returns the month, a number from 1 (January) through 12 (December) in number format of a given date.

Use

`MONTH(date)` and replace *date* with the field or expression for the date containing the month you want returned.

- 🕒 **SLA Expiration Example** `MONTH(SLAExpirationDate__c)` returns the month that your service-level agreement expires. This example uses a custom date field called **SLA Expiration Date**.
- 🕒 **Current Month Example** `MONTH(TODAY())` returns the current month in a number format. For example, the month of February would be the value “2.”

NOT

Returns FALSE for TRUE and TRUE for FALSE.

Use

`NOT(logical)` and replace *logical* with the expression that you want evaluated.

- 🕒 **Formula Example** `IF(NOT(ISPICKVAL(Status, "Closed")), ROUND(NOW() -CreatedDate, 0), null)` checks to see if a variable is open and if so, calculates the number of days it has been open by subtracting the date and time created from the current date and time. The result is the number of days open rounded to zero decimal places. If the variable is not open, this field is blank.

NOW

Returns a date/time representing the current moment.

Use

`NOW()`

Tips

- Do not remove the parentheses.
- Keep the parentheses empty. They do not need to contain a value.
- Use a date/time field in a NOW function instead of a date field. **Created Date** and **Last Modified Date** are date/time fields whereas **Last Activity Date** is a date field.

- Use **TODAY** if you prefer to use a date field.
- Dates and times are always calculated using the user's time zone.
- Use addition and subtraction operators with a NOW function and other date/time fields to return a number representing a number of days. For example `NOW() - CreatedDate` calculates the number of days since the created date of a record. In this example, the formula field data type is a number.
- Use addition and subtraction operators with a NOW function and numbers to return a date and time. For example `NOW() +5` calculates the date and time five days ahead of now. In this example, the formula field data type is a date/time.



Open Lead Example `IF(ISPICKVAL(Status, "Open"), ROUND(NOW()-CreatedDate, 0), null)` This formula checks to see if a lead is open and if so, calculates the number of days it has been open by subtracting the date and time created from the current date and time. The result is the number of days open rounded to zero decimal places. If the lead is not open, this field is blank.

NULLVALUE

Determines if an expression is null (blank) and returns a substitute expression if it is. If the expression is not blank, returns value of the expression.



Important Use **BLANKVALUE** instead of **NULLVALUE** in new formulas. **BLANKVALUE** has the same functionality as **NULLVALUE**, but also supports text fields. Salesforce continues to support **NULLVALUE**, so you don't need to change existing formulas.

Use

`NULLVALUE(expression, substitute_expression)` and replace *expression* with the expression you want to evaluate; replace *substitute_expression* with the value you want to replace any blank values.

Tips

- Avoid using this function with text fields because they are never null even when they are blank. Instead, use the **BLANKVALUE** function to determine if a text field is blank.
- Don't use **NULLVALUE** for date/time fields.
- Choose **Treat blank fields as blanks** for your formula when referencing a number, percent, or currency field in a **NULLVALUE** function. Choosing **Treat blank fields as zeroes** gives blank fields the value of zero so none of them are null.
- Use the same data type for both the *expression* and *substitute_expression*.



Custom Field Example `(NULLVALUE(Sample_Due_Date__c, StartDate +5))` This formula returns the date five days after the start date whenever **Sample Due Date** is blank. **Sample Due Date** is a custom date field.

OR

Determines if expressions are true or false. Returns TRUE if any expression is true. Returns FALSE if all expressions are false.

Use this function as an alternative to the operator [|| \(OR\)](#)

Use

`OR(logical1, logical2...)` and replace any number of logical references with the expressions you want evaluated.

 **Formula Field Example** `IF(OR(ISPICKVAL(Priority, "High"), ISPICKVAL(Status, "New")), ROUND(NOW()-CreatedDate, 0), null)` This formula returns the number of days a case has been open if the **Status** is new or the **Priority** is high. If the case was opened today, this field displays a zero.

 **Validation Rule Example**

```
OR(Sample_Rate__c < 0, Sample_Rate__c > 0.40)
```

This validation rule formula displays this error message when the **Sample Rate** custom field value isn't between 0% and 40%: "Sample Rate cannot exceed 40%."

PARENTGROUPVAL

This function returns the value of a specified parent grouping. A "parent" grouping is any level above the one containing the formula.

You can use this function only in custom summary formulas and at grouping levels for reports, but not at summary levels.

Use

- Summary and Joined:

```
PARENTGROUPVAL(summary_field, grouping_level)
```

- Matrix:

```
PARENTGROUPVAL(summary_field, parent_row_grouping, parent_column_grouping)
```

Where `summary_field` is the summarized field value, `grouping_level` is `GRAND_SUMMARY` or the API name of the parent level group for summary reports, and `parent_row_level` and `parent_column_level` are the parent levels for matrix reports.

In reports with multiple grouping levels, you can set the `grouping_level` to be any group level higher than the formula display level.

Formula Example

```
TOTAL_PRICE:SUM/PARENTGROUPVAL(TOTAL_PRICE:SUM, GRAND_SUMMARY)
```

This formula calculates, for each product, its relative size compared to the grand total. In this example, the report is a summary of opportunities and their products, grouped by **Product Name**.

PREDICT

Returns an Einstein Discovery prediction for a record based on the specified record ID or for a list of fields and their values.

Use

`PREDICT(PredDefId, [recordId] | [field, value, ...])`. Replace *PredDefId* with the Prediction Definition ID of a deployed prediction in your org. Specify the *recordId* of the record to predict or a list of fields and their associated values (*[field, value, ...]*).

Tips

- The specified *PredDefId* must link to a deployed and active prediction in your Salesforce org. To insert a value, select *PredDefId*, click **Insert Field**, select **SmartDataDiscovery > Prediction Definitions**, select an available prediction, select **Prediction Definition Id**, and then click **Insert**.



Note The syntax for *PredDefId* must match the following pattern:

```
$SmartDataDiscovery.PredictionDefinitions.<predictionDevName>.Id
```

- If you specify a *recordId*, the function returns a prediction for the data values in that record.
- If you specify a list of field names and values, the function returns a prediction for the data values you provided. Be sure to provide all the fields that the prediction requires as input.
- To view Einstein Discovery predictions, users must have the View Einstein Discovery Recommendations system permission. To learn more, see [Assign Einstein Discovery Permission Sets to Users](#).
- The PREDICT function is available when defining formulas for the following Process Automation components: approval processes, flows, processes (in Process Builder), workflow rules, and Next Best Action.
- The PREDICT function is not supported in formula-based fields on Salesforce objects.

recordId Example

```
PREDICT($SmartDataDiscovery.PredictionDefinitions.Recommende_kFjqS_1370.Id)
```

This example calls the PREDICT function and passes a prediction definition and recordId.

List of Fields and Value Example

```
PREDICT ($SmartDataDiscovery.PredictionDefinitions.Recommende_kFjqS_1370.I  
d,  
'Customer_Typec', Text(Customer_Typec), 'List_Price__c', List_Price__c)
```

This example calls the PREDICT function and passes a prediction definition and list of fields with associated values.

PREVGROUPVAL

This function returns the value of a specified previous grouping. A “previous” grouping is one that comes before the current grouping in the report.

Choose the grouping level and increment. The increment is the number of columns or rows before the current summary. The default is 1; the maximum is 12. You can use this function only in custom summary formulas and at grouping levels for reports, but not at summary levels.

Use

```
PREVGROUPVAL(summary_field, grouping_level [, increment])
```

Where `summary_field` is the name of the grouped row or column, `grouping_level` is the API name of the peer level group whose summary value for the previous grouping is used, and `increment` is the number of groupings previous.

In reports with multiple grouping levels, you can specify the `grouping_level` to be the same group level as the formula display level or a group level higher than the formula display level.

Formula Example

```
AMOUNT:SUM - PREVGROUPVAL(AMOUNT:SUM, CLOSE_DATE)
```

This formula calculates, for each month, the difference in amount from the previous month shown in the report. In this example, the report is an opportunity matrix with columns grouped by **Close Date** and rows by **Stage**.

PRIORVALUE

Returns the previous value of a field.

Use

`PRIORVALUE (field)`

Tips

- This function is available only in:
 - Assignment rules
 - Validation rules
 - Field updates
 - Workflow rules if the evaluation criteria is set to **Evaluate the rule when a record is: created, and every time it's edited**.
 - Formula criteria for executing actions and setting input values in Process Builder.
- This function doesn't return default values.
- When users create a record, this function returns the value of the *field* referenced rather than null. For example, if you create an account named "Acme," `PRIORVALUE (Account.Name)` returns `Acme`.
- When using the ISPICKVAL function to return the previous value of a picklist field, include the PRIORVALUE function inside the ISPICKVAL function as in this example:

```
ISPICKVAL (PRIORVALUE
(picklist_field),
text_literal)
```

- To check if the previous value of a multi-select picklist field includes a specific value, use the PRIORVALUE function inside the INCLUDES function. For example:

```
INCLUDES (
    PRIORVALUE (multiselect_picklist_field),
    text_literal
)
```

 **Validation Rule Example** This validation rule prevents users from changing the expected revenue of an opportunity after it's closed: `AND (PRIORVALUE (Amount) > Amount, IsClosed)`.

REGEX

Compares a text field to a regular expression and returns TRUE if there is a match. Otherwise, returns FALSE.

A regular expression is a string used to describe a format of a string according to certain syntax rules.

Use

`REGEX (text, regex_text)` and replace *text* with the text field, and *regex_text* with the regular

expression you want to match.

Tips

- Regular expression syntax is based on [Java Platform SE 6 syntax](#). However, backslash characters (\) must be changed to double backslashes (\\) because backslash is an escape character in Salesforce.
- The Salesforce regular expression engine matches an entire string as opposed to searching for a match within a string. For example, if you are searching for the name Marc Benioff, use the regular expression, `.*Marc Benioff.*`, to find a match in a string like the following:

According to Marc Benioff, the social enterprise increases customer success.

If you use the regular expression, `Marc Benioff`, the only string that this regular expression matches is:

Marc Benioff

- Capture groups and substitutions are ignored.
- This function is available everywhere formulas exist except formula fields and custom buttons and links.

 **Validation Rule Example** This example ensures that a custom field called **SSN** matches a regular expression representing a valid social security number format of the form 999-99-9999.

```
NOT (
    OR (
        LEN (SSN__c) = 0,
        REGEX(SSN__c, "[0-9]{3}-[0-9]{2}-[0-9]{4}")
    )
)
```

REQUIRESCRIPT

Returns a script tag with URL source that you specify. Use this function when referencing the Lightning Platform AJAX Toolkit or other JavaScript toolkits.

Use

`{ !REQUIRESCRIPT(url) }` and replace *url* with the link for the script that is required.

For the AJAX Toolkit:

```
{ !requireScript("/soap/ajax/13.0/connection.js") }
```

Returns:

```
<script src="/soap/ajax/13.0/connection.js"></script>
```

Tips

- Use global variables to access special merge fields for s-controls.
- Use this function when creating custom buttons or links where you have set Behavior to "Execute JavaScript" and Content Source to "OnClick JavaScript" because the script tag must be outside the OnClick code.
- This function is only available for custom buttons and links that have Content Source set to "OnClick JavaScript."
- When working in Visualforce, use INCLUDESCRIPT instead.



Custom Button Example

```
{!REQUIRESCRIPT("/soap/ajax/13.0/connection.js")}  
var c = new sforce.SObject("Case");  
c.id = "{!Case.Id}";  
c.Status = "New";  
result = sforce.connection.update([c]);  
window.location.reload();
```

This example sets the Status of a case to “New” whenever a user clicks a custom button from the case detail page. To set up this code in your organization, define a custom button for cases that has the following attributes:

- Display Type is “Detail Page Button”
- Behavior is “Execute JavaScript”
- Content Source is “OnClick JavaScript”

Next, paste this example code into your custom button definition and add it to your case page layouts.

REVERSE

Returns the characters of a source text string in reverse order.

Use

`REVERSE (text)` and replace `text` with the text string that you want returned in the reverse order.



Formula Example

`REVERSE (Sample)` returns the text `elpmas`.

RIGHT

Returns the specified number of characters from the end of a text string.

Use

`RIGHT(text, num_chars)` and replace *text* with the field or expression you want returned; replace *num_chars* with the number of characters from the right you want returned.

Tips

- Reference auto-number fields as text fields in formulas.
- If the *num_chars* value is less than zero, Salesforce replaces the value with zero.

 **Formula Example** `TRIM(LEFT(LastName, 5)) &"-"&TRIM(RIGHT(SSN__c, 4))` displays the first five characters of a name and the last four characters of a social security number separated by a dash. Note that this formula assumes you have a text custom field called **SSN**

ROUND

Returns the nearest number to a number you specify, constraining the new number by a specified number of digits.

Use

`ROUND(number, num_digits)` and replace *number* with the field or expression you want rounded; replace *num_digits* with the number of decimal places you want to consider when rounding.

 **Example** `ROUND(1.5, 0) = 2` `ROUND(1.2345, 0) = 1` `ROUND(-1.5, 0) = -2` `ROUND(225.49823, 2) = 225.50`

 **Simple Discounting Example** `ROUND(Amount - Amount * Discount_Percent__c, 2)` Use this formula to calculate the discounted amount of an opportunity rounded off to two digits. This example is a number formula field on opportunities that uses a custom percent field called **Discount Percent**.

RPAD

Inserts characters that you specify to the right-side of a text string.

Use

`RPAD(text, padded_length[, 'pad_string'])` and replace the variables:

- *text* is the field or expression after which you want to insert characters.
- *pad_length* is the number of total characters in the text string that is returned.
- *pad_string* is the character or characters to insert. *pad_string* is optional and defaults to a blank space.

If the value in *text* is longer than *pad_string*, *text* is truncated to the size of *padded_length*.

Tips

Ending blank spaces are omitted.

- 🕒 **Field Name Example: Padding Default** `RPAD(Name, 20)` truncates the **Name** field after 20 characters. For example, if the name is `mycompany.com`, the value returned is “mycompany.com.”
- 🕒 **My Company Example: No Change** `RPAD('my_company.com', 14, 'z')` returns “my_company.com” without change because it has 14 characters.
- 🕒 **Field Name Example: Padding with a Character** `RPAD(Name, 15, 'z')` returns “mycompany.comz” .
- 🕒 **Field Name Example: Truncating** `RPAD(Name, 2)` truncates the name after the second character. For example, if the name is `mycompany.com`, the value returned is “my.”

SECOND

Returns a seconds value in the form of a number from 0 through 60.

Use

`SECOND(time)` and replace *time* with a time value or value such as `TIMENOW()`.

- 🕒 **Formula Example** `SECOND(ClosedDate)` displays only the seconds in a time field based on the value of the Time Closed field. `SECOND(TIMEVALUE("17:30:45.125"))` returns 45.

SQRT

Returns the positive square root of a given number.

Use

`SQRT(number)` and replace *number* with the field or expression you want computed into a square root.

Tips

- Calculating the square root of a negative number results in an error on the detail page.
- Avoid division by zero errors by including an IF function such as: `IF(Amplitude__c >= 0, SQRT(Amplitude__c), null)`.

 **Square Root Example** `SQRT(25)` returns the square root of 25, which is 5.

 **Amplitude Example** `SQRT(Amplitude__c)` returns the square root of a custom number field representing the amplitude of an earthquake.

SUBSTITUTE

Substitutes new text for old text in a text string.

Use

`SUBSTITUTE(text, old_text, new_text)` and replace *text* with the field or value for which you want to substitute values, *old_text* with the text you want replaced, and *new_text* with the text you want to replace the *old_text*

Tips

- Each term provided in quotes is case-sensitive.
- If the *old_text* appears more than one time, each occurrence is replaced with the *new_text* value provided, even when it results in duplicates.

 **String Example** `SUBSTITUTE(Name, "Coupon", "Discount")` returns the name of an opportunity that contains the term “Coupon” with the opportunity name plus “Discount” wherever the term “Coupon” existed. `SUBSTITUTE>Email, LEFT>Email, FIND("@", Email)), "www."` finds the location of the @ sign in a person's email address to determine the length of text to replace with a “www.” as a means of deriving their website address.

TEXT

Converts a percent, number, date, date/time, or currency type field into text anywhere formulas are used. Also, converts picklist values to text in approval rules, approval step rules, workflow rules, escalation rules, assignment rules, auto-response rules, validation rules, formula fields, field updates, and custom buttons and links.

Use

`TEXT (value)` and replace *value* with the field or expression you want to convert to text format. Avoid using any special characters besides a decimal point (period) or minus sign (dash) in this function.

Tips

- The returned text is not formatted with any currency, percent symbols, or commas.
- Values are not sensitive to locale. For example, 24.42 EUR is converted into the number 24.42.
- Percents are returned in the form of a decimal.
- Dates are returned in the form of YYYY-MM-DD, that is, a four-digit year and two-digit month and day.
- Date/time values are returned in the form of YYYY-MM-DD HH:MM:SSZ where YYYY is a four-digit year, MM is a two-digit month, DD is a two-digit day, HH is the two-digit hour, MM are the minutes, SS are the seconds, and Z represents the zero meridian indicating the time is returned in UTC time zone.
- Picklist fields are supported in TEXT functions used in these kinds of formulas: validation rules, approval rules, approval step rules, workflow rules, escalation rules, assignment rules, auto-response rules, formula fields, field updates, and custom buttons and links. In other formulas, use ISPICKVAL or CASE when referencing a picklist field.
- If the text value you enter is a URL, the text automatically converts as a hyperlink.
- When converting decimal results to text in Lightning, leading zeros are removed. When converting decimal results to text in Classic, leading zeros are retained.

 **Expected Revenue in Text Example** `TEXT (ExpectedRevenue)` returns the expected revenue amount of an opportunity in text format without a dollar sign. For example, if the **Expected Revenue** of a campaign is "\$200,000," this formula field displays "200000."

 **Asset ID Example** `SerialNumber &"-"& TEXT (Quantity)` returns an asset ID number starting with the serial number and ending with the quantity separated by a dash. The **Serial Number** field is already text but the **Quantity** field is a number, requiring the TEXT function before it.

 **Picklist Values in Math Equations Example**

```
VALUE (LEFT (TEXT (Quantity), 5)) * Unit
```

This formula multiplies the first five numbers of the **Quantity** picklist by the **Unit** numeric field.

 **Compare Two Picklists Example**

```
IF (TEXT (bug_status) = TEXT (case_status), "Match", "Out of Sync")
```

This formula compares the values of the **bug_status** picklist with values of the **case_status** picklist.

 **Picklist Values from Parent Records Example**

```
TEXT (Account.Industry)
```

This formula field on opportunities shows the industry of the associated account.

Concatenate Picklist Values Example

```
TEXT(Account.Industry) & " - " & TEXT(Account.SubIndustry__c)
```

This formula field on opportunities shows the industry and subindustry of the associated account.

Validation Rule Example: Block the Save of a Closed Opportunity

`CONTAINS(TEXT(Status), "Closed")` returns TRUE if the **Status** picklist contains the value “Closed,” such as “Closed Won” and “Closed Lost.” This validation rule formula blocks users from saving changes to a closed opportunity.

Validation Rule Example: Use Numeric Functions on Numeric Picklist Values

`VALUE(LEFT(TEXT(Quantity), 5)) * Unit > 10000` multiplies the first five numbers of the **Quantity** picklist by the **Unit** numeric field, and returns TRUE if the result is greater than 10,000.

Directly Compare Two Picklists Example

`TEXT(bug_status) = TEXT(case_status)` compares the values of the **bug_status** picklist with values of the **case_status** picklist, and returns TRUE if they are equal.

TIMENOW

Returns a time value in GMT representing the current moment. Use this function instead of the NOW function if you only want to track time, without a date.

Use

```
TIMENOW()
```

Tips

- Do not remove the parentheses.
- Keep the parentheses empty. They do not need to contain a value.
- Use **TODAY** if you prefer to use a date field.
- The displayed value is based on the organization’s Locale settings.

Formula Example

`IF(ISPICKVAL(Rating , "Hot"), TIMENOW(), TIMEVALUE(CreatedDate))` This formula checks to see if a lead is rated “Hot” and if so, returns the current time. Otherwise it returns the time since someone created the lead.

TIMEVALUE

Returns the time value without the date, such as business hours.

Use

`TIMEVALUE (value)` and replace *value* with a date/time or text value, merge field, or expression.

Tips

- The displayed value is formatted based on the org's Locale settings.
- Don't use `TIMEVALUE` on a time field. A time field's value is already in time format. For example, for a time field with an API name `timefield_c`, `TIMEVALUE(timefield_c)` doesn't do anything.
- `TIMEVALUE` converts the input value to the time in GMT. This function doesn't take the user's time zone into account.

 **Time Example** `TIMEVALUE(ClosedDate)` displays a time value based on the value of the Date/Time Closed field. `TIMEVALUE ("17:30:45.125")` returns 5:30 PM.

TODAY

Returns the current date as a date data type.

Use

`TODAY ()`

Tips

- Do not remove the parentheses.
- Keep the parentheses empty. They do not need to contain a value.
- Use a date field with a TODAY function instead of a date/time field. **Last Activity Date** is a date field whereas **Created Date** and **Last Modified Date** are date/time fields.
- See [NOW](#) if you prefer to use a date/time field.
- Dates and times are always calculated using the user's time zone.
- Use addition and subtraction operators with a TODAY function and other date fields to return a number representing a number of days. For example `TODAY () -LastActivityDate` calculates the number of days since the last activity date. In this example, the formula field data type is a number.
- Use addition and subtraction operators with a TODAY function and numbers to return a date. For example `TODAY () +5` calculates the date five days ahead of today. In this example, the formula field data type is a date.

 **Formula Example** `TODAY () -Sample_date_c` calculates how many days in the sample are left.

 **Validation Rule Example**

```
SampleDate < TODAY()
```

This example ensures that users cannot change the **Sample Date** to any date in the past.

TRIM

Removes the spaces and tabs from the beginning and end of a text string.

Use

`TRIM(text)` and replace *text* with the field or expression you want to trim.

 **Formula Example** `TRIM(LEFT(LastName, 5)) & "-" & RIGHT.FirstName, 1)` returns a network ID for users that contains the first five characters of their last name and first character of their first name separated by a dash.

UPPER

Converts all letters in the specified text string to uppercase.

Any characters that are not letters are unaffected by this function. Locale rules are applied if a locale is provided.

Use

`UPPER(text, [locale])` and replace *text* with the field or expression you wish to convert to uppercase, and *locale* with the optional two-character ISO language code or five-character locale code, if available.

 **MYCOMPANY.COM Example** `UPPER("mycompany.com")` returns "MYCOMPANY.COM."

 **MYCOMPANY.COM 123 Example** `UPPER("Mycompany.com 123")` returns "MYCOMPANY.COM 123."

 **Turkish Language Locale Rules Example** The Turkish language has two versions of the letter i: one dotted and one dotless. The locale rules for Turkish require the ability to capitalize the dotted i, and allow the dotless I to be lowercase. To correctly use the `UPPER()` function with the Turkish language locale, use the Turkish locale code `tr` in the `UPPER()` function as follows:

`UPPER(text, "tr")` Using this locale code ensures that Salesforce doesn't transform any dotted i in the *text* to a dotless I.

URLENCODE

Encodes text and merge field values for use in URLs by replacing characters that are illegal in URLs, such as blank spaces, with the code that represent those characters as defined in RFC 3986, Uniform Resource Identifier (URI): Generic Syntax.

For example, blank spaces are replaced with `%20`, and exclamation points are replaced with `%21`.

Use

`{ !URLENCODE (text) }` and replace *text* with the merge field or text string that you want to encode.

Tips

- This function is only available in custom buttons and links, and in Visualforce.
- Custom buttons and links with URL content sources have separate encoding settings. If you use the URLENCODE function to encode a custom button or link that has an encoding setting specified, Salesforce first encodes the URL according to the custom button or link setting, then encodes the result. For example, if the URL in a custom link contains a space and its encoding setting is UTF8, Salesforce first encodes the space to a plus sign (+), then the URLENCODE function converts the plus sign to its character code, `%2B`.
- When you include the standard **Account** field on opportunities (`Opportunity.Account`) in the URLENCODE function, the value of the field is the account ID, not the account name. To encode the account name, create a custom cross-object formula field on opportunities that spans to the account name, and use that field in the URLENCODE function instead of `Opportunity.Account`. For example, if the cross-object formula is **AccountNameFormula__c**, use the following:

```
http://www.google.com/search?q={ !URLENCODE  
(Opportunity.AccountNameFormula__c) }
```



Merge Field Example If the merge field `foo_c` contains `Mark's page`,
`{ !URLENCODE (foo_c) }` results in: `%3CB%3EMark%27s%20page%3C%2Fb%3E`

URLFOR

Returns a URL for an action, an s-control, a Visualforce page, or a file in a static resource archive. URLFOR is available for use in custom buttons and links, s-controls, and Visualforce pages.

Use

`{ !URLFOR(target, [id], [inputs], [no override]) }` and replace *target* with the URL or action, s-control, or static resource merge variable; *id* with an optional reference to the record; and *inputs* with any optional parameters. The *no override* argument is also optional and defaults to `false`.

It applies to targets for standard Salesforce pages such as `$Action.Account.New`. Replace `no override` with `true` when you want to display a standard Salesforce page regardless of whether you have defined an override for it elsewhere.

The input values can be dynamic. For example, to include an account ID, specify:

```
{ !URLFOR($Page.myVisualforcePage, null, [accountId=Account.Id]) }
```

The resulting URL includes a parameter with the ID, such as:

```
https://MyDomainName--PackageName.vf.force.com/apex/myVisualforcePage?accountId=001B0000002tx0l
```

You can also use non-string variables, like an SObject variable. For example, if you supply `[myAccountParam=Account]`, the value is converted to a string, and the URL contains `?MyAccountParam=001B0000002tx0l`. You can also use a parameter to supply a static value, such as `[param1=55]`.

-  **Note** Because parameter names are static, you can't use a variable to determine the parameter name. For example, if you use `[myVariable="value"]` and set `myVariable` to `"param1"`, the resulting URL includes `?myVariable=value1` and not the `param1` value.

Tips

- Use global variables to access special merge fields for actions, s-controls, and static resources.
- If an input parameter name begins with any character other than a letter or dollar sign (\$), enclose it in quotation marks.
- Enclose multiple *inputs* in brackets to indicate they are together:

```
{ !URLFOR($Action.Case.View, Case.Id, [param1="A", param2="B"]) }
```

- Set *inputs* to `null` if you don't have any inputs to pass, but you want to set the `no override` argument:

```
{ !URLFOR($Action.Case.View, Case.Id, null, true) }
```

- When you override a standard action, that action is no longer available in Salesforce. For example, if you override the new account action, that affects the New button on all pages, such as the account detail page, account related lists on other detail pages, and the Create New dropdown list in the sidebar. To override a standard action but still access it, use the `no override` argument in your s-control to reference that action.
- When you override the tab home page for a standard or custom tab, use the tab's `$Action` global variable as the *target* value, and the tab's object type for the *id* value. For example,
`URLFOR($Action.Account.Tab, $ObjectType.Account)`.

 **Note** As of Winter '25, all Visualforce pages are served on the `force.com` domain or a site domain. `URLFOR` currently returns an absolute URL for all Visualforce pages. See [Ensure Access to Your Visualforce Pages in Summer '24 and Winter '25](#).

Visualforce Example

```
<apex:image url="{!!URLFOR($Resource.TestZip, 'images/Bluehills.jpg')}""  
width="50" height="50"/>
```

In this example, the `<apex:image>` component references a .jpg file contained within a .zip file that has been uploaded as a static resource. When uploaded, the name of the static resource was defined as `TestZip`, and the path to the image within the resource is `images/Bluehills.jpg`.

VALUE

Converts a text string to a number.

Use

`VALUE (text)` and replace *text* with the field or expression you want converted into a number.

Tips

Make sure the text in a VALUE function doesn't include special characters other than a decimal point (period) or minus sign (dash). If the text in a VALUE function is a non-numerical/invalid format, the formula isn't calculated and resolves to a blank value. For example, the formula `1 +`

`VALUE (Text_field__c)` produces these results:

- If Text field is 123, the result is 124.
- If Text field is blank, the result is blank.
- If Text field is \$123, the result is blank.
- If Text field is EUR123, the result is blank.

 **Lead Number** `VALUE (Lead_Number__c)` returns a number for the text value in the auto-number field Lead Number. This function is useful if you want to use the Lead Number field in a calculation. Note that auto-number fields are actually text fields and must be converted to a number for numeric calculations.

 **Round Robin Lead Assignment** `MOD(VALUE (Lead_Number__c), 3)` This formula is for a custom formula field named Round_Robin_ID that assigns each lead a value of 0, 1, or 2. This formula uses a custom auto-number field called Lead Number that assigns each lead a unique number starting with 1. The MOD function divides the lead number by the number of lead queues available (three in this example) and returns a remainder of 0, 1, or 2. Use the value of this formula field in your lead assignment rules to assign lead records to different queues. For example:

- Round_Robin_ID = 0 is assigned to Queue A
- Round_Robin_ID = 1 is assigned to Queue B
- Round_Robin_ID = 2 is assigned to Queue C

VLOOKUP

Returns a value by looking up a related value on a custom object similar to the VLOOKUP() Excel function. This function is only available in validation rules.

Use

VLOOKUP (*field_to_return*, *field_on_lookup_object*, *lookup_value*). Replace *field_to_return* with the field that contains the value you want returned, *field_on_lookup_object* with the field on the related object that contains the value you want to match, and *lookup_value* with the value you want to match.

Tips

- *field_to_return* must be an auto number, roll up summary, lookup relationship, master-detail relationship, checkbox, date, date/time, email, number, percent, phone, text, text area, or URL field type.
- *field_on_lookup_object* must be the **Record Name** field on a custom object.
- *field_on_lookup_object* and *lookup_value* must be the same data type.
- If more than one record matches, the value from the first record is returned.
- The value returned must be on a custom object.
- You can't delete the custom field or custom object referenced in this function.

 **Validation Rule Example** This example checks that a billing postal code is valid by looking up the first five characters of the value in a custom object called Zip_Code__c that contains a record for every valid ZIP code in the US. If the ZIP code isn't found in the Zip_Code__c object or the billing state doesn't match the corresponding State_Code__c in the Zip_Code__c object, an error displays.

```
AND (
    LEN(BillingPostalCode) > 0,
    OR(BillingCountry = "USA", BillingCountry = "US"),
    VLOOKUP(
        $ObjectType.Zip_Code__c.Fields.State_Code__c,
        $ObjectType.Zip_Code__c.Fields.Name,
        LEFT(BillingPostalCode, 5)
    ) <> BillingState
)
```



Note

- Use this example when the billing country is US or USA.
- You can download US ZIP codes in CSV file format from <http://zips.sourceforge.net>.

See Also

[Validation Rules](#)

WEEKDAY

Returns the day of the week for the given date, using 1 for Sunday, 2 for Monday, through 7 for Saturday.

Use

`WEEKDAY (date)`

 **Formula Example** `WEEKDAY(customdate1__c)` returns the day of the week for the given date in `customdate1__c`

YEAR

Returns the four-digit year in number format of a given date.

Use

`YEAR (date)` and replace `date` with the field or expression that contains the year you want returned.

 **Formula Example** `YEAR(TODAY()) - YEAR(Initial_Meeting__c)` returns the number of years since your initial meeting with a client. This example uses a custom date field called **Initial Meeting**.

Using Date, Date/Time, and Time Values in Formulas

Date formulas are useful for managing payment deadlines, contract ages, or any other features of your organization that are time or date dependent.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

Two data types are used for working with dates: Date and Date/Time. One data type, Time, is independent of the date for tracking time such as business hours. Most values that are used when

working with dates are of the Date data type, which store the year, month, and day. Some fields, such as CreatedDate, are Date/Time fields, meaning they not only store a date value, but also a time value (stored in GMT but displayed in the users' time zone). Date, Date/Time, and Time fields are formatted in the user's locale when viewed in reports and record detail pages. A Time value's precision is in milliseconds. A Date/Time value's precision is in seconds.

You can use operations like addition and subtraction on Date, Date/Time, and Time values to calculate a future date or elapsed time between two dates or times. If you subtract one date from another, for example, the resulting value will be the difference between the two initial values in days (Number data type). The same operation between two Date/Time values returns a decimal value indicating the difference in number of days, hours, and minutes. The same operation between two Time values returns millisecond

For example, if the difference between two Date/Time values is 5.52, that means the two values are separated by five days, 12 hours (0.5 of a day), and 28 minutes (0.02 of a day). You can also add numeric values to Dates and Date/Times. For example, the operation `TODAY() + 3` returns three days after today's date. For more information and examples of working with dates, see the list of [Sample Date Formulas](#).

Throughout the examples, the variables `date` and `date/time` are used in place of actual Date and Date/Time fields or values.

Keep in mind that complex date functions tend to compile to a larger size than text or number formula functions, so you might run into issues with formula compile size.

TODAY(), NOW() and TIMENOW()

The `TODAY()` function returns the current day, month, and year as a Date data type. This function is useful for formulas where you are concerned with how many days have passed since a previous date, the date of a certain number of days in the future, or if you just want to display the current date.

The `NOW()` function returns the Date/Time value of the current moment. It's useful when you are concerned with specific times of day as well as the date.

The `TIMENOW()` function returns a value in GMT representing the current time without the date. Use this function instead of the `NOW()` function if you want the current hour, minute, seconds, or milliseconds. This value is useful for tracking time like work shifts or elapsed time,

For details on how to convert between Date values and Date/Time values, see [Converting Between Date/Time and Date](#).

The DATE() Function

The `DATE()` function returns a Date value, given a year, month, and day. Numerical Y/M/D values and the `YEAR()`, `MONTH()`, and `DAY()` functions are valid parameters for `DATE()`. For example `DATE(2013, 6, 1)` returns June 1, 2013. Similarly, `DATE(YEAR(TODAY()), MONTH(TODAY()) +`

`3, 1)` returns the Date value of the first day three months from today in the current year, assuming the date is valid (for example, the month falls between 1 and 12).

If the inputted Y/M/D values result in an invalid date, the `DATE()` function returns an error, so error checking is an important part of working with Date values. You can read about methods for handling invalid dates in [Sample Date Formulas](#).

Converting Between Date/Time and Date

Date and Date/Time aren't interchangeable data types, so when you want to perform operations between Date and Date/Time values, you need to convert the values so they are both the same type. Some functions (such as `YEAR()`, `MONTH()`, and `DAY()`) also only work on Date values, so Date/Time values must be converted first.

Use the `DATEVALUE(date/time)` function to return the Date value of a Date/Time. For example, to get the year from a Date/Time, use `YEAR(DATEVALUE(date/time))`.

 **Note** If a formula references a Date/Time field but returns a Date, the time component is ignored without converting to the user's time zone. This behavior can result in an incorrect date. To correct for the user's time zone, use the `DATEVALUE()` function to convert the Date/Time field.

You can convert a Date value to a Date/Time using the `DATETIMEVALUE(date)` function. The time will be set to 12:00 a.m. in Greenwich Mean Time (GMT), and then converted to the time zone of the user viewing the record when it's displayed. For a user located in San Francisco, `DATETIMEVALUE(TODAY())` returns 5:00 p.m. on the previous day (during Daylight Saving Time) rather than 12:00 a.m. of the current day. See [A Note About Date/Time and Time Zones](#) for more information.

Converting Between Date/Time and Time

The `TIMEVALUE()` function returns a Time data type value in “HH:MM:SS.MS” (hours:minutes:seconds.milliseconds) format using a 24-hour clock. Numerical H/M/S/MS values and the `HOUR()`, `MINUTE()`, `SECONDS()`, and `MILLISECONDS()` functions are valid parameters for `TIMEVALUE()`.

Use the `TIMEVALUE(value)` function to return the Time value of a Date/Time type, text, merge field or expression. For example, extract the time from a ClosedDate Date/Time value with `TIMEVALUE(ClosedDate)`.

Converting Between Date and Text

If you want to include a date as part of a string, wrap the Date value in the `TEXT()` function to convert it to text. For example, if you want to return today's date as text, use:

```
"Today's date is " & TEXT( TODAY() )
```

This returns the date in the format “YYYY-MM-DD” rather than in the locale-dependent format. You can change the format by extracting the day, month, and year from the date first and then recombining them in the format you want. For example:

```
"Today's date is " & TEXT( MONTH( date ) ) & "/" & TEXT( DAY( date ) ) & "/" & TEXT( YEAR( date ) ) )
```

You can also convert text to a Date so you can use the string value with your other Date fields and formulas. You’ll want your text to be formatted as “YYYY-MM-DD”. Use this formula to return the Date value:

```
DATEVALUE( "YYYY-MM-DD" )
```

Converting Between Date/Time and Text

You can include Date/Time values in a string using the `TEXT()` function, but you need to be careful of time zones. For example, consider this formula:

```
"The current date and time is " & TEXT( NOW() )
```

In this formula, `NOW()` is offset to GMT. Normally, `NOW()` would be converted to the user’s time zone when viewed, but because it’s been converted to text, the conversion won’t happen. So if you execute this formula on August 1st at 5:00 PM in San Francisco time (GMT-7), the result is “The current date and time is 2013-08-02 00:00:00Z”.

When you convert a Date/Time to text, a “Z” is included at the end to indicate GMT. `TEXT(date/time)` returns “Z” if the field is blank. So if the Date/Time value you’re working with might be blank, check for this before converting to text:

```
IF(
    ISBLANK( date/time ),
    "",
    TEXT( date/time )
)
```

To convert a string to a Date/Time value, use `DATETIMEVALUE()` passing in a string in the format “YYYY-MM-DD HH:MM:SS”. This method returns the Date/Time value in GMT.

Converting Between Time and Text

If you want to include time as part of a string, wrap the Time value in the `TEXT()` function to convert it to text. For example, if you want to return the current time as text, use:

```
"The time is " & TEXT( TIMENOW() )
```

This function returns the time in the format “HH:MM:SS.MS”.

You can also convert text to a Time data type so you can use the string value with your other Time fields and formulas. Format your text as “HH:MM:SS.MS” on a 24-hour clock. Use the `TIMEVALUE()` function:

```
TIMEVALUE("17:30:45.125")
```

A Note About Date/Time and Time Zones

Date and Date/Time values are stored in GMT. When a record is saved, field values are adjusted from the user’s time zone to GMT, and then adjusted back to the viewer’s time zone when displayed in record detail pages and reports. With Date conversions this doesn’t pose a problem, since converting a Date/Time to a Date results in the same Date value.

When working with Date/Time fields and values, however, the conversion is always done in GMT, not the user’s time zone. Subtracting a standard Date/Time field from another isn’t a problem because both fields are in the same time zone. When one of the values in the calculation is a conversion from a Text or Date value to a Date/Time value, however, the results are different.

Let’s say a San Francisco user enters a value of 12:00 AM on August 2, 2013 in a custom Date/Time field called `Date_Time_c`. This value is stored as 2013-08-02 07:00:00Z, because the time difference in Pacific Daylight Time is GMT-7. At 12:00 p.m. PDT on August 1st, the user views the record and the following formula is run:

```
Date_Time_c - NOW()
```

In the calculation, `NOW()` is 2013-08-01 19:00:00Z, and then subtracted from 2013-08-02 07:00:00Z, to return the expected result of 0.5 (12 hours).

Suppose that instead of `NOW()`, the formula converts the string “2013-08-01 12:00:00” to a Date/Time value:

```
Date_Time_c - DATETIMEVALUE( "2013-08-01 12:00:00" )
```

In this case, `DATETIMEVALUE("2013-08-01 12:00:00")` is 2013-08-01 12:00:00Z, and returns a result of 0.79167, or 19 hours.

There's no way to determine a user's time zone in a formula. If all of your users are in the same time zone, you can adjust the time zone difference by adding or subtracting the time difference between the users' time zone and GMT to your converted values. However, since time zones can be affected by Daylight Saving Time, and the start and end dates for DST are different each year, this is difficult to manage in a formula. We recommend using Apex for transactions that require converting between Date/Time values and Text or Date values.

See Also

[TIMEVALUE](#)

[DATEVALUE](#)

[DATETIMEVALUE](#)

[Tips for Building Formulas](#)

[Considerations for the Time Custom Field Type](#)

Build a Formula Field

Your custom formula fields require special attributes.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To view formula field details: View Setup and Configuration

To create, change, or delete formula fields: Customize Application

 **Note** The [Getting Started with Formulas \(Salesforce Classic\)](#) help video includes a live demo of these steps.

1. Begin building a formula field the same way you create a custom field. See [Create Custom Fields](#).
2. Select the data type for the formula. Choose the appropriate data type for your formula based on the output of your calculation. See [Formula Data Types](#).
3. Choose the number of decimal places for currency, number, or percent data types. This setting is ignored for currency fields in multicurrency organizations. Instead, the **Decimal Places** for your currency setting apply. Salesforce uses the round half up tie-breaking rule for numbers in formula fields. For example, 12.345 becomes 12.35 and -12.345 becomes -12.35.
4. Click **Next**.
5. Build your formula. Formula fields can contain up to 3,900 characters, including spaces, return characters, and comments. If your formula requires more characters, create separate formula fields and reference them in another formula field. The maximum number of displayed characters after an evaluation of a formula expression is 1,300
 - a. If you are building a formula in the **Advanced Formula** tab or for approvals or rules, such as

workflow, validation, assignment, auto-response, or escalation, click **Insert Field**, choose a field, and click **Insert**. To create a basic formula that passes specific Salesforce data, select the **Simple Formula** tab, choose the field type in the **Select Field Type** drop-down list, and choose one of the fields listed in the **Insert Field** drop-down list. Build cross-object formulas to span to related objects and reference merge fields on those objects.

- b. To insert an operator, choose the appropriate operator icon from the **Insert Operator** drop-down list.
- c. Optionally, click the **Advanced Formula** tab to use functions and view other operators and merge fields. Functions are prebuilt formulas that you can customize with your input parameters.
- d. To insert a function, double-click its name in the list, or select it and click **Insert Selected Function**. To filter the list of functions, choose a category from the **Functions** drop-down list. Select a function and click **Help on this function** to view a description and examples of formulas using that function.
- e. Consider adding comments to your formula, especially if it is complicated. Comments must begin with a forward slash followed by an asterisk (`/*`), and conclude with an asterisk followed by a forward slash (`*/`).

Comments are useful for explaining specific parts of a formula to anyone viewing the formula definition. For example:

```
AND(
/*competitor field is required, check to see if field is empty */
LEN(Competitor__c) = 0,
/* rule only enforced for ABCD record types */
RecordType.Name = "ABCD Value",
/* checking for any closed status, allows for additional closed picklist values in the future */
CONTAINS(TEXT(StageName), "Closed")
)
```

6. To check your formula for errors, click **Check Syntax**.
7. Optionally, enter a description of the formula in the **Description** box.
8. If your formula references any number, currency, or percent fields, choose an option for handling blank fields. To give any blank fields a zero value, choose **Treat blank fields as zeros**. To leave these fields blank, choose **Treat blank fields as blanks**.
9. Click **Next**.
10. Set the field-level security to determine whether the field should be visible for specific profiles, and click **Next**.
11. Choose the page layouts that should display the field. The field is added as the last field in the first two-column section on the page layout. For user custom fields, the field is automatically added to the bottom of the user detail page.
12. Click **Save** to finish or **Save & New** to create more custom fields.



Note Because formula fields are automatically calculated, they are read-only on record detail pages and do not update last modified date fields. Formula fields are not visible on edit pages. In account formulas, all business account fields are available as merge fields. However, account fields exclusive

to person accounts such as **Birthdate** and **Email** are not available.

Tips for Building Formulas

Keep these tips in mind when working with formulas.

What Is a Cross-Object Formula?

A *Cross-object formula* is a formula that spans two related objects and references merge fields on those objects. A cross-object formula can reference merge fields from a master (“parent”) object if an object is on the detail side of a master-detail relationship. A cross-object formula also works with lookup relationships.

See Also

[Elements of a Formula](#)

[Merge Fields for Formulas](#)

[Tips for Building Formulas](#)

[Formula Operators and Functions by Context](#)

Tips for Building Formulas

Keep these tips in mind when working with formulas.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

- Formula fields that a user can see may reference fields that are hidden or read only using field-level security. If the formula field contains sensitive information, use field-level security to hide it.
- You can add activity formula fields to task and event page layouts. Note that a task-related formula field on an event page layout may not be useful. Likewise, event-related formula fields on task page layouts may not be useful.
- To determine if a record is a task or event, use the **IsTask** merge field. For example:

```
IF(IsTask, "This is a task", "This is an event")
```

Tips for Working with Date and Date/Time Formula Fields

Keep these tips in mind when working with Date and Date/Time formula fields.

Tips for Working with Hyperlink Formula Fields

Use these tips to understand how links open from formula custom fields that contain a **HYPERLINK** function.

Tips for Using Merge Fields in Formulas

Keep these tips in mind when using merge fields in formulas.

Tips for Working with Number Formula Fields

Useful tips for working with number fields.

Tips for Working with Picklist and Multi-Select Picklist Formula Fields

Consider these tips when creating single- and multi-select picklist formula fields.

Tips for Referencing Record Types in Formulas

Reference record types in formulas if you want different workflow rules, validation rules, and lookup filters to apply to different record types.

Tips for Working with Text Formula Fields

Keep these tips in mind when working with text formula fields.

See Also

[Build a Formula Field](#)

[Common Formula Errors](#)

Tips for Working with Date and Date/Time Formula Fields

Keep these tips in mind when working with Date and Date/Time formula fields.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

- Dates and times are always calculated using the user's time zone.
- Date and date/time fields can't be used interchangeably. The name alone may not indicate if a field is a date or date/time. For example, Created Date and Last Modified Date are date/time fields whereas Last Activity Date is a date field. Use the DATEVALUE function to convert a date/time field into a date field.



Note The Created Date and Last Modified Date fields display only the date, not the date and time.

- Use addition and subtraction operators with date or date/time fields to calculate duration. For example, subtract a date from another date to calculate the number of days between the two. Likewise, you can subtract the date/time from another date/time to get the number of days between the two as a number. See NOW or TODAY for suggested use.
- Use addition and subtraction operators with numbers to return another date or date/time. For example, `{!CreatedDate} + 5` calculates the date and time five days after a record's created date. Note that the expression returns the same data type as the one given; a date field plus or minus a number returns a date, and a date/time field plus or minus a number returns a date/time.
- When calculating dates using fractions, Salesforce ignores any numbers beyond the decimal. For example:

`TODAY() + 0.7` is the same as `TODAY() + 0`, which is today's date.

`TODAY() + 1.7` is the same as `TODAY() + 1`, which is tomorrow's date.

`TODAY() + (-1.8)` is the same as `TODAY() + (-1)`, which is yesterday's date.

- To calculate the value of two fractions first, group them within parentheses. For example:

`TODAY() + 0.5 + 0.5` is the same as `TODAY() + 0 + 0`, which is today's date.

`TODAY() + (0.5+0.5)` is the same as `TODAY() + 1`, which is tomorrow's date.

- Years can't be zero and must be between -4713 and 9999.

See Also

[Tips for Building Formulas](#)

Tips for Working with Hyperlink Formula Fields

Use these tips to understand how links open from formula custom fields that contain a `HYPERLINK` function.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

To create clickable links in Formula fields, use the `HYPERLINK` function or a valid URL. If you have formula custom fields that contain a `HYPERLINK` function, the server generates an HTML anchor for the link. For example, this function: `HYPERLINK("/apex/VF_TEST", "VFLINK", '_self')` generates this HTML output: `VFLINK`. If the `HYPERLINK` function doesn't contain a target attribute, it defaults to a value of `target="_blank"` in the generated HTML. A value of `"_blank"` opens the link in a new window or browser tab, outside of Lightning Experience or the Lightning Console.

Note

- The `HYPERLINK` function doesn't support the `bizztel` protocol on record home pages for custom objects in Lightning Experience.
- The `HYPERLINK` function only supports specific URL protocols in Lightning Experience. If the formula uses an unsupported protocol, the URL is removed and the anchor tag is rendered without a valid href. Supported protocols include: `ftp`, `ftps`, `http`, `https`, `mailto`, `tel`, `callto`, `cid`, `xmpp`, `ciscotel`, and `navision`. For details, see the [Formatted Rich Text](#).

The **Lightning Experience Honors Target Values for Hyperlinks in Formula Fields** critical update ensures that the `target` value for hyperlinks is honored, whether it's explicitly configured or set by default. If you have enabled the critical update, you can keep the target page within Lightning navigation by adding a value of `target="_self"` to your formula field `HYPERLINK` functions. If you specify something other than `target="_self"`, the link opens with standard browser navigation outside of Lightning Experience.

If you haven't enabled the critical update, relative links open in a new tab regardless of the `target` value.

 **Note** This critical update is automatically enabled in Summer '20. Both before and after enabling

the critical update, external links always open in a new tab, regardless of the `target` value.

Link Type	Target	Expected Behavior	Example
Relative Visualforce page or image	<code>_self</code>	<p>The Visualforce page or image opens in Lightning Experience or Console inside the same browser tab.</p> <p>Whether the link opens in a new subtab or in the same workspace tab depends on the Console app's navigation rules. By default, links from record detail pages open in a new subtab, and links from reports open in the same workspace tab.</p>	<code>HYPERLINK("/apex/VFPAGE", "Visualforce Page", "_self")</code>
Relative Visualforce page or image	<code>_blank</code>	<p>The Visualforce page or image opens outside of Lightning Experience or Console in a new browser tab.</p> <p>A critical update that is available in Winter '19 and automatically enabled in Summer '19 on May 17, 2019 controls this behavior. If you don't enable the critical update, these links open inside Lightning Experience or Console in a new browser tab.</p>	<code>HYPERLINK("/img/logo214", "Logo", "_blank")</code>
External website	<code>_self</code>	The website opens in a new browser tab.	<code>HYPERLINK("https://salesforce.com",</code>

Link Type	Target	Expected Behavior	Example
			"Salesforce", "_self")
External website	_blank	The website opens in a new browser tab.	HYPERLINK("https://salesforce.com", "Salesforce", "_blank")

Tips for Using Merge Fields in Formulas

Keep these tips in mind when using merge fields in formulas.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: All Editions

- Delegated administrators need to have access to custom objects to access the objects' merge fields from formulas.
- In account formulas, all business account fields are available as merge fields. However, account fields exclusive to person accounts such as Birthdate and Email are not available.
- You can't use formula fields that include related object merge fields in roll-up summary fields.
- Formulas and roll-up summary fields can't reference fields on external objects.
- Using `RecordType.Id` can make your formula less readable; when you do use it, write in-line comments into the formula to clarify.
- To determine if a record is a task or event, use the **IsTask** merge field. For example:

```
IF(IsTask, "This is a task", "This is an event")
```

- To reference the unique identifier for your Salesforce organization in a formula, insert the `$Organization.Id` merge field. This merge field can display anywhere formula fields can except in reports.
- Some merge fields display as radio buttons but function like picklist fields when referenced in a formula.

Use the values “Read,” “Edit,” and “None” in a formula when referencing:

- `$UserRole.CaseAccessForAccountOwner`
- `$UserRole.OpportunityAccessForAccountOwner`
- `CaseAccessLevel (on Territory)`
- `OpportunityAccessLevel (on Territory)`

Use the values “Read,” “Edit,” and “All” in a formula when referencing:

- `AccountAccessLevel (on Territory)`

- If you create a contacts formula field that references account merge fields, that field can be included in contact page layouts but should not be included in person accounts page layouts. The formula field

will display a value of `#Error` on the person accounts page.

See Also

[Tips for Building Formulas](#)

Tips for Working with Number Formula Fields

Useful tips for working with number fields.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

- Use the decimal version of a percent when working with percent fields in formulas. For example, `IF(Probability =1...)` for 100% probability or `IF(Probability =0.9...)` for 90% probability.
- Reference auto-number fields as text fields in formulas.
- The output of your formula must be less than 19 digits.
- Formulas can contain a mix of numbers, percents, and currencies as in this example: `AnnualRevenue / NumberOfEmployees`.
- Salesforce uses the round half up tie-breaking rule for numbers in formula fields. For example, 12.345 becomes 12.35 and -12.345 becomes -12.35.

See Also

[Tips for Building Formulas](#)

Tips for Working with Picklist and Multi-Select Picklist Formula Fields

Consider these tips when creating single- and multi-select picklist formula fields.

REQUIRED EDITIONS

Available in: both Lightning Experience and Salesforce Classic

Available in: all editions

- You can use special picklist fields in your formulas, such as **IsEscalated** for cases and **IsWon** for opportunities.
- Picklist fields can only be used in these functions.
 - **ISPICKVAL**—Compares the value of a picklist to a single value.
 - **CASE**—Compares the value of a picklist to multiple values.
 - **TEXT**—Returns a picklist value's API Name so that you can work with a reference to the value, even if the displayed value changes, in functions that support text values, such as **CONTAINS**. Available in only flow formula resources, formula fields, validation rules, and workflow field updates.

- Multi-select picklist fields can only be used in these functions.
 - CONTAINS, in Process Builder, in which the criteria for executing actions are set to **Conditions are met**.
 - INCLUDES
 - ISBLANK
 - ISNULL
 - ISCHANGED. Only in assignment rules, validation rules, workflow field updates, and workflow rules in which the evaluation criteria are set to **Evaluate the rule when a record is: created, and every time it's edited**.
 - PRIORVALUE. Only in assignment rules, validation rules, workflow field updates, and workflow rules in which the evaluation criteria are set to **Evaluate the rule when a record is: created, and every time it's edited**.

For more details about the functions, see [Formula and Operator Functions](#).

See Also

[Tips for Building Formulas](#)

Tips for Referencing Record Types in Formulas

Reference record types in formulas if you want different workflow rules, validation rules, and lookup filters to apply to different record types.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

Record types available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

For example, you can:

- Create a workflow rule on accounts that emails different teams based on the account record type the user selects when creating the account.
- Create a validation rule on opportunities that allows only members of the North American sales team to save opportunities with the Domestic record type.

When possible, use `RecordTypeId` instead of `RecordType.Name` to reference a specific record type.

While `RecordType.Name` makes a formula more readable, you must update the formula if the name of the record type changes, whereas the ID of a record type never changes. Also, `RecordType.Name` requires a cross-object reference to the record type, while `RecordTypeId` doesn't. However, if you are deploying formulas across organizations (for example, between sandbox and production), use `RecordType.Name` because IDs are not the same across organizations.

Avoid using `$RecordType` in formulas, except in default value formulas. Instead, use the `RecordType` merge field (for example, `Account.RecordType.Name`) or the `RecordTypeId` field on the object.

See Also

[Tips for Building Formulas](#)

Tips for Working with Text Formula Fields

Keep these tips in mind when working with text formula fields.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

- Before using the HYPERLINK function, consider the differences between hyperlinks and custom links.
 - Hyperlink formula fields are just like other custom fields that you can show in list views and reports.
 - Custom links appear on detail pages in a predefined section; hyperlink formula fields can appear on a detail page wherever you specify.
 - Using custom links, you can specify display properties such as window position and opening in a separate popup position. Hyperlink formula fields open in a new browser window by default or you can specify a different target window or frame.
 - Your formulas can reference custom links. Before deleting a custom link, make sure that it's not referenced in a formula field.
 - Hyperlink formula fields that contain relative URLs to Salesforce pages, such as `/rpt/reportwizard.jsp`, can be added to list views, reports, and related lists. However, use a complete URL, including the server name and `https://`, in your hyperlink formula before adding it to a search layout.
- To insert text in your formula field, surround the text with quotation marks. For example, to display "CASE: 123," use this formula `"CASE: "& CaseNumber__c`.
- If a formula field is used in a Classic email template, surround empty spaces with quotation marks. Otherwise, use `BR()`. For example, use this formula for Classic email templates: `TEXT(Amount)&" "& TEXT(CloseDate) &" "& TEXT(CreatedDate)`
- Use the backslash (\) character before a quote or backslash to insert it as a literal value in your output. For example, `"Trouble\\Case \"Ticket\": "` in your formula displays Trouble\Case "Ticket": on detail pages.
- In Salesforce Classic, Knowledge articles show URLs from Formula (Text) fields as plain text. In Lightning Experience, Knowledge articles show such URLs as clickable links.

See Also

[Tips for Building Formulas](#)

What Is a Cross-Object Formula?

A *Cross-object formula* is a formula that spans two related objects and references merge fields on those objects. A cross-object formula can reference merge fields from a master (“parent”) object if an object is on the detail side of a master-detail relationship. A cross-object formula also works with lookup

relationships.

REQUIRED EDITIONS

-  **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

You can reference fields from objects that are up to 10 relationships away. A cross-object formula is available anywhere formulas are used except when creating default values.

-  **Note** If you create a formula that references a field on another object and display that formula in your page layout, users can see the field on the object even if they don't have access to that object record. For example, if you create a formula field on the Case object that references an account field, and display that formula field in the case page layout, users can see this field even if they don't have access to the account record.

You can't include an object as the lookup field in a formula. To reference an object, reference the object's ID field or another field on the object. For example, `Account.Owner` is invalid because it references the object directly. `Account.Owner.FirstName` or `Account.OwnerId` are valid references for your formula.

[Building Cross-Object Formulas in the Simple Formula Tab](#)

To create a cross-object formula when building a formula in the Simple Formula tab, enter the relationship names of the objects to which you are spanning followed by the field you want to reference. Separate the relationship names of each object and the field with periods.

[Build Cross-Object Formulas in the Advanced Formula Tab](#)

Create a cross-object formula when building a formula in the Advanced Formula tab or for approvals or rules, such as workflow, validation, assignment, auto-response, or escalation rules.

[Tips for Building Cross-Object Formulas](#)

Keep these tips in mind when working with cross-object formulas.

See Also

[Build a Formula Field](#)

[Building Cross-Object Formulas in the Simple Formula Tab](#)

To create a cross-object formula when building a formula in the Simple Formula tab, enter the relationship names of the objects to which you are spanning followed by the field you want to reference. Separate the relationship names of each object and the field with periods.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To create or change cross-object formulas: Customize Application

 **Example** For example, enter `Contact.Account.Name` to reference the **Account Name** for a contact associated with a case in a formula field on the Case object. Be sure to use the relationship names of the objects, not the labels. Although the relationship name is often the same as the object name, it is technically the field name of the relationship field. To reference the parent account name from Account object, the syntax is `Parent.Name`, not `Account.Name`. When referencing a custom object, add two underscores and the letter *r* to its name. For example, `Position__r.title__c` references the **Job Title** field (`title__c`) on a Position custom object.

 **Note** If you create a formula that references a field on another object and display that formula in your page layout, users can see the field on the object even if they don't have access to that object record. For example, if you create a formula field on the Case object that references an account field, and display that formula field in the case page layout, users can see this field even if they don't have access to the account record.

See Also

[Build Cross-Object Formulas in the Advanced Formula Tab](#)

[What Is a Cross-Object Formula?](#)

Build Cross-Object Formulas in the Advanced Formula Tab

Create a cross-object formula when building a formula in the Advanced Formula tab or for approvals or rules, such as workflow, validation, assignment, auto-response, or escalation rules.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To create or change cross-object formulas: Customize Application

Related objects are denoted by a “>” sign.

 **Note** If you create a formula that references a field on another object and display that formula in your page layout, users can see the field on the object even if they don't have access to that object

record. For example, if you create a formula field on the Case object that references an account field, and display that formula field in the case page layout, users can see this field even if they don't have access to the account record.



Example

The value of the `Profile.Name` merge field differs depending on the context of the cross-object formula field that references it. On detail pages, the value is the profile name, as expected. In list views and reports, the value is the internal value of the associated profile instead. If you use `Profile.Name` in a formula, use it within an `OR` function to ensure that the formula always returns the intended result. For example:

```
IF  
  (OR  
    (LastModifiedBy.Profile.Name = "Standard User", LastModifiedBy.P  
     rofile.Name = "PT2"),  
    "Standard", "Not Standard")
```

None of the above applies to profile names referenced by the `$Profile` global variable.

See Also

[Building Cross-Object Formulas in the Simple Formula Tab](#)

[What Is a Cross-Object Formula?](#)

Tips for Building Cross-Object Formulas

Keep these tips in mind when working with cross-object formulas.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

- Cross-object formulas that reference currency fields convert the value to the currency of the record that contains the formula. If the referenced currency field is from a custom setting, the field value isn't converted to the record's currency.
- Salesforce allows a maximum of 10 unique relationships per object in cross-object formulas. The limit is cumulative across all formula fields, rules, and lookup filters. For example, if two different formulas on opportunities reference two different fields of an associated account, only one unique relationship exists (from opportunities to accounts).
- You can't reference cross-object formulas in roll-up summary fields.
- In cross-object formulas, you can't reference merge fields for objects related to activities. For example, merge fields for contacts and accounts aren't available in task and event formulas.
- In cross-object formulas, you can't reference fields from contacts through person accounts. For more information, see [Using custom formula fields with person accounts](#).

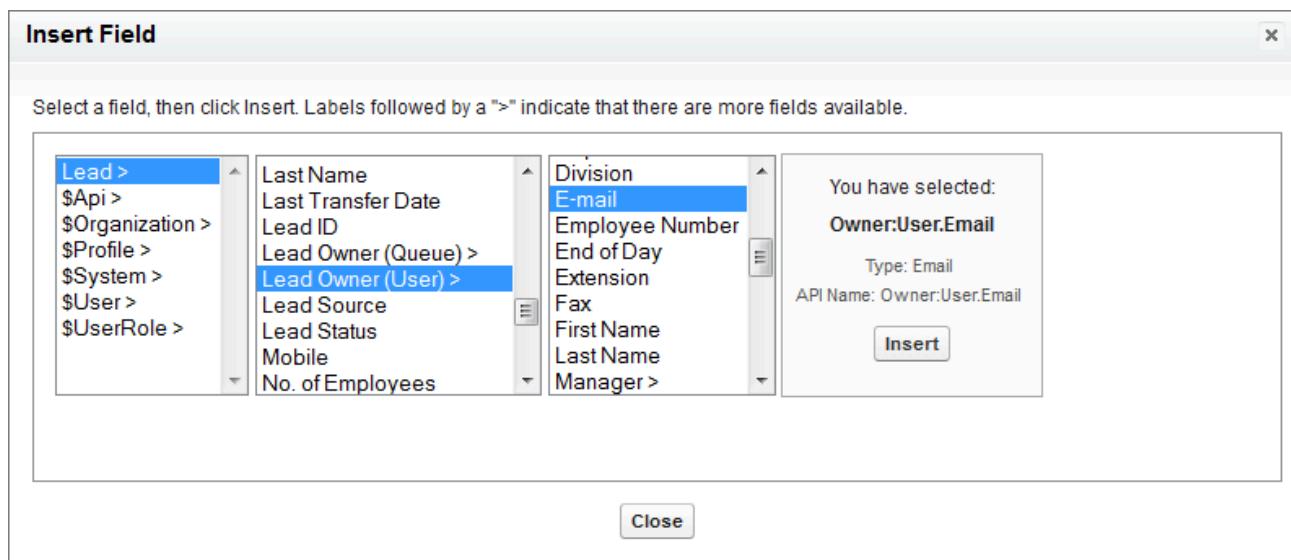
Using the Owner Field

Some objects support different object types for the Owner field, such as a User, Queue, or Calendar. When you create a cross-object formula using Owner on such objects, be explicit about the owner type that you reference.

For example, for owner email where you don't use queues, your formula is `Owner:User.Email`. If you do use queues, your formula can be

```
IF( ISBLANK( Owner:User.Id ), Owner:Queue.QueueEmail, Owner:User.Email )
```

Here's how to select Owner object fields on a Lead in the Advanced formula tab.



- Owner references aren't supported in Visualforce pages. For example, on a page with Case as a controller, you can't include `{ !Case.Owner:User.FirstName }`. However, you can include an existing spanning formula on a Visualforce page. For example, if you have a custom text formula `MyFormula__c` on a Case with value `Owner:User.FirstName`, you can include `{ !Case.MyFormula__c }` on your Visualforce page.
- Owner references aren't supported on the Queue object. For example, you can't reference `Owner:Queue.Owner.Email`.
- If your formula has `Owner:User.fieldname` and `Owner:Queue.fieldname`, they count against the limit of 10 unique relationships per object in cross-object formulas.
- On objects that don't support Queues, User is implicit when referencing Owner. Have your formula as `Owner.fieldname`, not `Owner:User.fieldname`.

Using `Profile.Name`

The value of the `Profile.Name` merge field differs depending on the context of the cross-object formula field that references it. On detail pages, the value is the profile name, as expected. In list views

and reports, the value is the internal value of the associated profile instead. If you use `Profile.Name` in a formula, use it within an `OR` function to ensure that the formula always returns the intended result. For example:

```
IF  
  (OR  
    (LastModifiedBy.Profile.Name = "Standard User", LastModifiedBy.Profi  
     le.Name = "PT2"),  
    "Standard", "Not Standard")
```

None of the above applies to profile names referenced by the `$Profile` global variable.

Fields That Can't Be Used In Formulas

Fields that are calculated from other fields, like Name fields

For example, the Account object has an `OwnerId` field that refers to users. The relationship name is Owner.

So you can use the `Owner.Email` field in a formula. But if you try to use `Owner.Name` field in a formula, you get this message: Error: Field name doesn't exist. Check spelling.

To work around this issue, you can use a formula that builds a name directly. For example, on an Account object, you can use `Owner.FirstName & " " & Owner.LastName`

Polymorphic relationship fields

For example, the Case object has a polymorphic relationship field called `OwnerId` that can refer to Groups or Users. The relationship name is Owner. If you try to use `Owner.Email` in a formula, you get this message: Error: Specify an object type for the Owner Field.

To work around this issue:

1. Create a custom field on the object that has a lookup relationship to the object that you want to use in the formula.

For example, create a `UserOwner` custom field on the Case object that has a lookup relationship to Users.

2. Set the custom field to refer to the object.

To continue the example, set the `UserOwner` field for a specific Case to refer to the required user.

3. Use the custom field in a formula.

For example, you can use `UserOwner__r.Email`, or `UserOwner__r.FirstName & " "`
`& UserOwner__r.LastName`

See Also

[Build Cross-Object Formulas in the Advanced Formula Tab](#)

[What Is a Cross-Object Formula?](#)

Formula Field Limits and Restrictions

Before you create formula fields, be aware of their limits and limitations.

REQUIRED EDITIONS

 **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

- Formula fields have these limits.
 - Character limit – Formula fields can contain up to 3,900 characters, including spaces, return characters, and comments. If your formula needs more characters, create separate formula fields and reference them in another formula field.
 -  **Note** The maximum number of displayed characters after an evaluation of a formula expression is 1,300.
 - Save size limit – Formula fields can't exceed 15,000 bytes when saved. If you use multi-byte characters in your formula, the save size is different from the number of characters
 - Compile size limit – Formula fields can't exceed 15,000 bytes when compiled. The compile size is the size of the formula (in bytes) including all of the fields, values, and formulas it references. There's no direct correlation between the compile size and the character limit. Some functions, such as TEXT, DATEVALUE, DATETIMEVALUE, and DATE significantly increase the compile size.
- Default value formulas for a record type can only reference fields for that record type. But formula fields and formulas for approvals or rules for a record type can reference fields for that record type and any records that are related through a lookup or master-detail relationship. For example, a formula for a validation rule on opportunities can reference merge fields for accounts, campaigns, and opportunities. A formula field on accounts can reference fields for cases.
- You can't use long text area, encrypted, or **Description** fields in formulas.
- The value of a field can't depend on another formula that references it.
- You can't delete fields referenced in formulas. Remove the field from the formula before deleting it.
- Campaign statistic fields can't be referenced in formulas for field updates, approval processes, workflow rules, or validation rules, but can be referenced in custom formula fields.
- The UI escapes HTML tags used in formula fields. To create an HTML element, replace your HTML with a function, like HYPERLINK or IMAGE.

- Custom formula fields from contacts can't be referenced through person accounts.
- The use of NULL as an expression isn't supported in a Checkbox formula field.
- You can't directly reference aggregate fields on parent objects in formula fields, such as **GrandTotal** on the **Quote** object. Consider using **Roll-Up Summary** fields to aggregate values from child records for these calculations.

See Also

[Tips for Building Formulas](#)
[Build a Formula Field](#)

Formula Best Practices

You can use the Formula Editor in Salesforce to construct a simple formula with a few clicks. But what if you want to build something more complex? Use these tips to help you map out formula logic and make it easier to troubleshoot errors.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

 **Thank you** Special thanks to [Trailblazer Chris Emmett](#) for contributing this content.

Tip 1: Put Every Function on a Separate Line

It's easy to fall into the habit of keeping an entire formula on a single line, especially when the formula is small. Putting each function on its own line makes the formula easier to read and troubleshoot. These examples show the same formula, first with no line breaks, and then with each function on a separate line.

```
IF(AND(ISBLANK(myDate_c),active_c=true),"Missing Date","Not Applicable")
```

```
IF(  
AND(  
ISBLANK(myDate_c),  
active_c=true  
)  
,  
"Missing Date",  
"Not Applicable"  
)
```

Tip 2: Indent Sections Within Parentheses

When your formula involves multiple functions, indentation helps visually isolate each function and makes it easier to identify errors, such as misplaced characters.

In this example, with indentation, you see that the bulk of the formula sits within a single `IF` statement and that the `AND` statement contains two functions. Inside the `AND` statement, the function `ISBLANK` is enclosed in parentheses.

```
IF (
    AND (
        ISBLANK(myDate_c),
        active_c=true
    ),
    "Missing Date",
    "Not Applicable"
)
```

Indentation can also help you zero in on mistakes. With a flat layout, it's difficult to see that an extra “`)`” is included after the `ISBLANK` statement, and there are no visual clues about how the formula is structured.

```
IF (
    AND (
        ISBLANK(myDate_c)
    ),
    active_c=true
),
"Missing Date",
"Not Applicable"
)
```

The indented layout makes it easy to see the formula's structure. You can quickly find and remove the extra character so that the `AND` statement is correctly formatted.

```
IF (
    AND (
        ISBLANK(myDate_c)
    ),
    active_c=true
),
"Missing Date",
"Not Applicable"
```

```
)
```

Tip 3: Write Statement and Function Names in Uppercase

All the examples here use uppercase letters for statement and function names, such as `IF`, `AND`, and `ISBLANK`. Using uppercase for these terms creates a clear distinction between functions and parameters and brings some visual clarity to a complex formula.

Tip 4: Handle Null and Required Input Field Values

These examples reference a field called `myDate__c` and use the `ISBLANK` check to confirm that the field is populated. It's important to verify the contents of any field in a formula. Without this verification, a formula can fail. For example, if you add a second date to the formula and perform a greater than operation, include the `ISBLANK` check for the second date to ensure that the formula executes correctly.

```
IF(  
    AND(  
        ISBLANK(myDate__c),  
        ISBLANK(mySecondDate__c),  
        active__c=true,  
        mySecondDate__c > myDate__c  
    ),  
    "Missing Date",  
    "Not Applicable"  
)
```

Tip 5: Use the HYPERLINK() Function to Create URLs

To create valid clickable links in Formula fields, use the `HYPERLINK()` function. This function is a TEXT function with syntax

```
HYPRELINK(URL, link_name [,target])
```

Examples of Advanced Formula Fields

Review examples of formula fields for various types of apps that you can use and modify for your own purposes.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To view formula field details: View Setup and Configuration

To create, change, or delete formula fields: Customize Application

This document contains custom formula samples for the following topics. For details about using the functions included in these samples, see [Formula Operators and Functions by Context](#).

[Sample Account Management Formulas](#)

Use these formulas to manage account details.

[Sample Account Media Service Formulas](#)

Formulas to link to specific search sites and media accounts.

[Sample Case Management Formulas](#)

Use these formulas to manage case details.

[Sample Commission Calculations Formulas](#)

Use these formulas to calculate commission amounts.

[Sample Contact Management Formulas](#)

Use these formulas to manage contact details.

[Sample Data Categorization Formulas](#)

Use these formulas for data categorizations.

[Sample Date Formulas](#)

Use the sample formulas in this topic to manipulate and perform calculations with date and time.

[Sample Discounting Formulas](#)

Use these formulas to calculate discount amounts.

[Sample Employee Services Formulas](#)

Use these formulas for employee services.

[Sample Expense Tracking Formulas](#)

Use these formulas for expense tracking.

[Sample Financial Calculations Formulas](#)

Use these formulas for financial calculations.

[Sample Image Link Formulas](#)

Use these formulas for image links.

[Sample Integration Link Formulas](#)

Use these formulas for integration links.

[Sample Lead Management Formulas](#)

Use these formulas to manage leads.

Sample Metric Formulas

Use these formulas for metric temperature and metric unit of measure conversion.

Sample Opportunity Management Formulas

Use these formulas for business expenses and earnings.

Sample Pricing Formulas

Use these formulas for total amounts and user pricing.

Sample Scoring Calculations Formulas

Use these formulas for lead scoring and customer success scoring.

See Also

[Formulas: How Do I ... ?](#)

[Tips for Building Formulas](#)

Sample Account Management Formulas

Use these formulas to manage account details.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

For details about using the functions included in these samples, see [Formula Operators and Functions by Context](#).

Account Rating

This formula evaluates **Annual Revenue**, **Billing Country**, and **Type**, and assigns a value of “Hot,” “Warm,” or “Cold.”

```
IF (AND (AnnualRevenue > 10000000,  
CONTAINS (CASE (BillingCountry, "United States", "US", "America", "US", "USA",  
"US", "NA"), "US")),  
IF(ISPICKVAL(Type, "Manufacturing Partner"), "Hot",  
IF(OR (ISPICKVAL (Type, "Channel Partner/Reseller"),  
ISPICKVAL(Type, "Installation Partner"))), "Warm", "Cold")),  
"Cold")
```

In addition, you can reference this Account Rating formula field from the contact object using cross-object formulas.

```
Account.Account_Rating__c
```

Account Region

This formula returns a text value of “North,” “South,” “East,” “West,” or “Central” based on the **Billing State/Province** of the account.

```
IF(ISBLANK(BillingState), "None",
IF(CONTAINS("AK:AZ:CA:HA:NV:NM:OR:UT:WA", BillingState), "West",
IF(CONTAINS("CO:ID:MT:KS:OK:TX:WY", BillingState), "Central",
IF(CONTAINS("CT:ME:MA:NH:NY:PA:RI:VT", BillingState), "East",
IF(CONTAINS("AL:AR:DC:DE:FL:GA:KY:LA:MD:MS:NC:NJ:SC:TN:VA:WV", BillingState),
"South",
IF(CONTAINS("IL:IN:IA:MI:MN:MO:NE:ND:OH:SD:WI", BillingState), "North", "Other")))))
```

Contract Aging

This formula calculates the number of days since a contract with an account was activated. If the contract **Status** isn’t “Activated,” this field is blank.

```
IF(ISPICKVAL(Contract_Status__c, "Activated"),
NOW() - Contract_Activated_Date__c, null)
```

Sample Account Media Service Formulas

Formulas to link to specific search sites and media accounts.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

For details about using the functions included in these samples, see [Formula Operators and Functions by Context](#).

BBC™ News Search

This formula creates a link to a BBC news search site based on the **Account Name**.

```
HYPERLINK(
"http://www.bbc.co.uk/search/news/?q=&Name",
"BBC News")
```

Bloomberg™ News Search

This formula creates a link to an account's ticker symbol on the Bloomberg website.

```
HYPERLINK(  
    "http://www.bloomberg.com/markets/symbolsearch?query=""&TickerSymbol,  
    "Bloomberg News")
```

CNN™ News Search

This formula creates a link to a CNN news search site using the **Account Name**.

```
HYPERLINK(  
    "http://www.cnn.com/search/?query=""&Name,  
    "CNN News")
```

MarketWatch™ Search

This formula creates a link to an account's ticker symbol on the Marketwatch.com website.

```
HYPERLINK(  
    "http://www.marketwatch.com/investing/stock/"&TickerSymbol,  
    "Marketwatch")
```

Google™ Search

This formula creates a link to a Google search site using the **Account Name**.

```
HYPERLINK(  
    "http://www.google.com/#q=""&Name,  
    "Google")
```

Google News Search

This formula creates a link to a Google news search site using the **Account Name**.

```
HYPERLINK(  
    "http://news.google.com/news/search?en&q=""&Name,  
    "Google News")
```

Yahoo!™ Search

This formula creates a link to a Yahoo! search site using the **Account Name**.

```
HYPERLINK(  
    "http://search.yahoo.com/search?p="&Name,  
    "Yahoo Search")
```

Yahoo! News Search

This formula creates a link to a Yahoo! news search site using the **Account Name**.

```
HYPERLINK(  
    "http://news.search.yahoo.com/search/news?p="&Name,  
    "Yahoo News")
```

Sample Case Management Formulas

Use these formulas to manage case details.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **All** Editions

For details about using the functions included in these samples, see [Formula Operators and Functions](#).

Autodial

This formula creates a linkable phone number field that automatically dials the phone number when clicked. In this example, replace `"servername"` and `"call"` with the name of your dialing tool and the command it uses to dial. The merge field, Id, inserts the identifier for the contact, lead, or account record. The first Phone merge field tells the dialing tool what number to call and the last Phone merge field uses the value of the Phone field as the linkable text the user clicks to dial.

```
HYPERLINK("http://servername/call?id=" & Id & "&phone=" &  
          Phone, Phone)
```

Case Categorization

This formula displays a text value of “RED,” “YELLOW,” or “GREEN,” depending on the value of a case age custom text field.

```
IF(DaysOpen__c > 20, "RED",
IF(DaysOpen__c > 10, "YELLOW",
"GREEN") )
```

Case Data Completeness Tracking

This formula calculates the percentage of specific custom fields that contain data. The formula checks the values of two custom number fields: **Problem Num** and **Severity Num**. If the fields are empty, the formula returns the value “0.” The formula returns a value of “1” for each field that contains a value and multiplies this total by fifty to give you the percentage of fields that contain data.

```
(IF(ISBLANK(Problem_Num__c), 0, 1) + IF(ISBLANK(Severity_Num__c ), 0,1)) * 50
```

Suggested Agent Prompts

This formula prompts an agent with cross-sell offers based on past purchases.

```
CASE(Product_Purch__c,
"Printer", "Extra toner cartridges", "Camera", "Memory cards",
"Special of the day")
```

Suggested Offers

This formula suggests a product based on the support history for a computer reseller. When the **Problem** custom field matches a field, the formula field returns a suggestion.

```
CASE(Problem__c,
"Memory", "Suggest new memory cards", "Hard Drive failure", "Suggest new hard
drive with tape backup",
 "")
```

Sample Commission Calculations Formulas

Use these formulas to calculate commission amounts.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

For details about using the functions included in these samples, see [Formula Operators and Functions](#).

Commission Amounts for Opportunities

The following is a simple formula where commission is based on a flat 2% of the opportunity **Amount**.

```
IF(ISPICKVAL(StageName, "Closed Won"),  
    ROUND(Amount *0.02, 2), 0)
```

This example calculates the commission amount for any opportunity that has a “Closed Won” stage. The value of this field is the amount times 0.02 for any closed or won opportunity. Open or lost opportunities have a zero commission value.

Commission Deal Size

This formula calculates a commission rate based on deal size, returning a 9% commission rate for deals over 100,000 and an 8% commission rate for smaller deals.

```
IF(Amount > 100000, 0.09, 0.08 )
```

Commission Greater Than or Equal To

This formula assigns the YES value with a commission greater than or equal to one million. Note, this field is a text formula field that uses a custom currency field called Commission.

```
IF(Commission__c >=  
    1000000, "YES", "NO")
```

Commission Maximum

This formula determines what commission to log for an asset based on which is greater: the user's commission percentage of the price, the price times the discount percent stored for the account or 100 dollars. This example assumes you have two custom percent fields on users and assets.

```
MAX($User.Commission_Percent__c * Price,
```

```
Price * Account_Discount__c, 100)
```

Sample Contact Management Formulas

Use these formulas to manage contact details.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

For details about using the functions included in these samples, see [Formula Operators and Functions](#).

Contact's Account Discount Percent

This percent formula displays the account's **Discount Percent** field on the contacts page.

```
Account.Discount_Percent__c
```

Contact's Account Name

This formula displays the standard **Account Name** field on the contacts page.

```
Account.Name
```

Contact's Account Phone

This formula displays the standard **Account Phone** field on the contacts page.

```
Account.Phone
```

Contact's Account Rating

Use this formula to display the **Account Rating** field on the contacts page.

```
CASE(Account.Rating, "Hot", "Hot", "Warm", "Warm", "Cold", "Cold", "Not Rate  
d")
```

Contact's Account Website

This formula displays the standard **Account Website** field on the contacts page.

```
Account.Website
```

If the account website URL is long, use the HYPERLINK function to display a label such as “Click Here” instead of the URL. For example:

```
IF(Account.Website="", "",  
    IF(  
        OR(LEFT(Account.Website, 7) = "http://", LEFT(Account.Website,  
8) = "https://"),  
        HYPERLINK( Account.Website , "Click Here" ),  
        HYPERLINK( "https://" & Account.Website , "Click Here" )  
    )  
)
```

This formula also adds the necessary “https://” before a URL if “http://” or “https://” wasn’t previously included in the URL field.

Contact's LinkedIn™ Profile

You can configure a link that appears on your contacts' profile page that sends you to their LinkedIn profile. To do so:

1. From the object management settings for contacts, go to Buttons, Links, and Actions.
2. Click **New Button or Link**.
3. Enter a **Label** for this link, like *LinkedInLink*.
4. Enter this formula in the content box:

```
https://www.linkedin.com/search/fpsearch?type=people&keywords  
={!Contact.FirstName}+{!Contact.LastName}
```

5. Click **Save**.

Remember to add this link to the Contact page layout in order for it to show up.

Contact Identification Numbering

This formula displays the first five characters of a name and the last four characters of a social security number separated by a dash. This example uses a text custom field called **SSN**.

```
TRIM(LEFT(LastName, 5)) &
      "-" & TRIM(RIGHT(SSN__c, 4))
```

Contact Preferred Phone

This formula displays the contact's preferred contact method in a contact related list—work phone, home phone, or mobile phone—based on a selected option in a **Preferred Phone** custom picklist.

```
CASE(Preferred_Phone__c,
"Work", "w. " & Phone,
"Home", "h. " & HomePhone,
"Mobile", "m. " & MobilePhone,
"No Preferred Phone")
```

Contact Priority

This formula assesses the importance of a contact based on the account rating and the contact's title. If the account rating is **Hot** or the title starts with **Executive**, then the priority is high (P1). If the account rating is **Warm** or the title starts with **VP** then the priority is medium (P2), and if the account rating is **Cold** then the priority is low (P3).

```
IF(OR(ISPICKVAL(Account.Rating, "Hot"), CONTAINS>Title, "Executive")), "P1",
IF(OR(ISPICKVAL(Account.Rating, "Warm"), CONTAINS>Title, "VP")), "P2",
IF(ISPICKVAL(Account.Rating, "Cold"), "P3",
"P3")
```

Contact Yahoo! ID

This formula displays a clickable Yahoo! Messenger icon indicating if the person is logged on to the service. Users can click the icon to launch a Yahoo! Messenger conversation with the person. This example uses a custom text field called **Yahoo Name** on contacts where you can store the contact's Yahoo! Messenger ID.

```
HYPERLINK("ymsgr:sendIM?" & Yahoo_Name__c,
IMAGE("https://opi.yahoo.com/online?u=" & Yahoo_Name__c &
```

```
"&m;=g&t;=0", "Yahoo"))
```

Dynamic Address Formatting

This formula field displays a formatted mailing address for a contact in standard format, including spaces and line breaks where appropriate depending on the country.

```
CASE(ShippingCountry,
"USA",
ShippingStreet & BR() &
ShippingCity & ",",
" & ShippingState & " " &
ShippingPostalCode & BR()
& ShippingCountry,
"France",
ShippingStreet & BR() &
ShippingPostalCode & " " &
ShippingCity & BR() &
ShippingCountry, "etc")
```

Phone Country Code

This formula determines the phone country code of a contact based on the **Mailing Country** of the mailing address.

```
CASE(MailingCountry,
"USA", "1",
"Canada", "1",
"France", "33",
"UK", "44",
"Australia", "61",
"Japan", "81",
"?")
```

Unformatted Phone Number

This formula removes the parentheses and dash characters from North American phone numbers. This formula is necessary for some auto-dialer software.

```
IF(Country_Code__c = "1", MID( Phone ,2, 3) & MID(Phone,7,3) & MID(Phone,1
1,4), Phone)
```

Sample Data Categorization Formulas

Use these formulas for data categorizations.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

For details about using the functions included in these samples, see [Formula Operators and Functions](#).

Deal Size Large and Small

This formula displays “Large Deal” for deals over one million dollars or “Small Deal” for deals under one million dollars.

```
IF(Sales_Price__c > 1000000,  
    "Large Deal",  
    "Small Deal")
```

Deal Size Small

This formula displays `Small` if the price and quantity are less than one. This field is blank if the asset has a price or quantity greater than one.

```
IF(AND(Price<1,Quantity<1),"Small", null)
```

Product Categorization

This formula checks the content of a custom text field named **Product_Type** and returns `Parts` for any product with the word “part” in it. Otherwise, it returns `Service`. The values are case-sensitive, so if a **Product_Type** field contains the text “Part” or “PART,” this formula returns `Services`.

```
IF(CONTAINS(Product_Type__c, "part"), "Parts", "Service")
```

Sample Date Formulas

Use the sample formulas in this topic to manipulate and perform calculations with date and time.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

Find the Day, Month, or Year from a Date

Use the functions `DAY(date)`, `MONTH(date)`, and `YEAR(date)` to return their numerical values. Replace `date` with a value of type Date (for example, `TODAY()`).

To use these functions with Date/Time values, first convert them to a date with the `DATEVALUE()` function. For example, `DAY(DATEVALUE(date/time))`.

Find Out If a Year Is a Leap Year

This formula determines whether a year is a leap year. A year is only a leap year if it's divisible by 400, or if it's divisible by four but not by 100.

```
OR(
    MOD( YEAR( date ), 400 ) = 0,
    AND(
        MOD( YEAR( date ), 4 ) = 0,
        MOD( YEAR( date ), 100 ) != 0
    )
)
```

Find Which Quarter a Date Is In

For standard quarters, you can determine which quarter a date falls in using this formula. This formula returns the number of the quarter that `date` falls in (1–4) by dividing the current month by three (the number of months in each quarter) and taking the ceiling.

```
CEILING( MONTH( date ) / 3 )
```

The formula for shifted quarters is similar, but shifts the month of the date by the number of months between January and the first quarter of the fiscal year. The following example shows how to find a date's quarter if Q1 starts in February instead of January.

```
CEILING( ( MONTH( date ) - 1 ) / 3 )
```

To check whether a date is in the current quarter, add a check to compare the date's year and quarter

with `TODAY()`'s year and quarter.

```
AND(
    CEILING( MONTH( date ) / 3 ) = CEILING( MONTH( TODAY() ) / 3 ),
    YEAR( date ) = YEAR( TODAY() )
)
```

Find the Week of the Year a Date Is In

To find the number of a date's week of the year, use this formula:

```
IF(
    CEILING( ( date - DATE( YEAR( date ), 1, 1 ) + 1 ) / 7 ) > 52,
    52,
    CEILING( ( date - DATE( YEAR( date ), 1, 1 ) + 1 ) / 7 )
)
```

To find the current week number, determine the days to date in the current year and divide that value by 7. The `IF()` statement ensures that the week number the formula returns doesn't exceed 52. So if the given date is December 31 of the given year, the formula returns 52, even though it's more than 52 weeks after the first week of January.

Find Whether Two Dates Are in the Same Month

To determine whether two Dates fall in the same month, say for a validation rule to determine whether an opportunity Close Date is in the current month, use this formula:

```
AND(
    MONTH( date_1 ) == MONTH( date_2 ),
    YEAR( date_1 ) == YEAR( date_2 )
)
```

Find the Last Day of the Month

The easiest way to find the last day of a month is to find the first day of the next month and subtract a day.

```
IF(
    MONTH( date ) = 12,
    DATE( YEAR( date ), 12, 31 ),
    DATE( YEAR( date ), MONTH( date ) + 1, 1 ) - 1
)
```

```
)
```

Display the Month as a String instead of a Number

To return the month as a text string instead of a number, use:

```
CASE(  
    MONTH( date ),  
    1, "January",  
    2, "February",  
    3, "March",  
    4, "April",  
    5, "May",  
    6, "June",  
    7, "July",  
    8, "August",  
    9, "September",  
    10, "October",  
    11, "November",  
    "December"  
)
```

If your organization uses multiple languages, you can replace the names of the month with a custom label:

```
CASE(  
    MONTH( date ),  
    1, $Label.Month_of_Year_1,  
    2, $Label.Month_of_Year_2,  
    3, $Label.Month_of_Year_3,  
    4, $Label.Month_of_Year_4,  
    5, $Label.Month_of_Year_5,  
    6, $Label.Month_of_Year_6,  
    7, $Label.Month_of_Year_7,  
    8, $Label.Month_of_Year_8,  
    9, $Label.Month_of_Year_9,  
    10, $Label.Month_of_Year_10,  
    11, $Label.Month_of_Year_11,  
    $Label.Month_of_Year_12  
)
```

Find and Display the Day of the Week from a Date

To find the day of the week from a Date value, use a known Sunday, for example, January 7, 1900, and subtract it from the date, for example, `TODAY()`, to get the difference in days. The `MOD()` function finds the remainder of this result when divided by 7 to give the numerical value of the day of the week between 0 (Sunday) and 6 (Saturday). This formula finds the result and then returns the text name of that day.

```
CASE(
    MOD( date - DATE( 1900, 1, 7 ), 7 ),
    0, "Sunday",
    1, "Monday",
    2, "Tuesday",
    3, "Wednesday",
    4, "Thursday",
    5, "Friday",
    "Saturday"
)
```

This formula only works for dates after 01/07/1900. If you work with older dates, use the same process with any Sunday before to your earliest date, for example, 01/05/1800.

You can adjust this formula if your week starts on a different day. For example, if your week starts on Monday, you can use January 8, 1900 in your condition. The new formula looks like this:

```
CASE(
    MOD( date - DATE( 1900, 1, 8 ), 7 ),
    0, "Monday",
    1, "Tuesday",
    2, "Wednesday",
    3, "Thursday",
    4, "Friday",
    5, "Saturday",
    "Sunday"
)
```

To get the formula for the name of the month, if your organization uses multiple languages, you can replace the names of the day of the week with a variable like `$Label.Day_of_Week_1`, and so on.

Find the Next Day of the Week After a Date

To find the date of the next occurrence of a particular day of the week following a given Date, get the difference in the number of days of the week between a `date` and a `day_of_week`, a number 0–6 where 0 = Sunday and 6 = Saturday. By adding this difference to the current date, you can find the date of the

day_of_week. The `IF()` statement in this formula handles cases where the *day_of_week* is before the day of the week of the *date* value (for example *date* is a Thursday and *day_of_week* is a Monday) by adding 7 to the difference.

```
date + ( day_of_week - MOD( date - DATE( 1900, 1, 7 ), 7 ) )
+
IF(
    MOD( date - DATE( 1900, 1, 7 ), 7 ) >= day_of_week,
    7,
    0
)
```

You can substitute either a constant or another field in for the *day_of_week* value based on your needs.

Find the Number of Days Between Two Dates

To find the number of days between two dates, *date_1*, and *date_2*, subtract the earlier date from the later date: `date_1 - date_2`

You can alter this formula slightly if you want to determine a date that's a certain number of days in the past. For example, to create a formula to return true if some date field is more than 30 days before the current date and false otherwise, use a formula such as the following:

```
TODAY() - 30 > date
```

Find the Number of Weekdays Between Two Dates

Calculating how many weekdays passed between two dates is slightly more complex than calculating total elapsed days. In this example, weekdays are Monday through Friday. The basic strategy is to choose a reference Monday from the past and find out how many full weeks and any additional portion of a week have passed between the reference date and your date. These values are multiplied by five for a five-day work week, and then the difference between them is taken to calculate weekdays.

```
(5 * ( FLOOR( ( date_1 - DATE( 1900, 1, 8 ) ) / 7 ) ) + MIN( 5, MOD( date_1 - DATE( 1900, 1, 8 ), 7 ) ) )
-
(5 * ( FLOOR( ( date_2 - DATE( 1900, 1, 8 ) ) / 7 ) ) + MIN( 5, MOD( date_2 - DATE( 1900, 1, 8 ), 7 ) ) )
```

In this formula, *date_1* is the more recent date and *date_2* is the earlier date. If your work week runs shorter or longer than five days, replace all fives in the formula with the length of your week.

Find the Number of Months Between Two Dates

To find the number of months between two dates, subtract the year of the earlier date from the year of the later date and multiply the difference by 12. Next, subtract the month of the earlier date from the month of the later date, and add that difference to the value of the first set of operations.

```
((YEAR(date_1) - YEAR(date_2))*12) + (MONTH(date_1) - MONTH(date_2))
```

Add Days, Months, and Years to a Date

If you want to add a certain number of days to a date, add that number to the date directly. For example, to add 5 days to a date, the formula is `date + 5`.

If you want to add a certain number of months to a date, use this function.

```
ADDMONTHS()
```

For example, if you want to add 4 months to a date, use this formula.

```
ADDMONTHS(date + 4)
```

If the date that you provide is the last of any month, this formula returns the last day of the resulting month.

To add a certain number of years to a date, use this formula.

```
ADDMONTHS(date, 12*num_years)
```

If the date that you provide is February 29, and the resulting year isn't a leap year, the formula returns the date as February 28. In this scenario, if you want the resulting date as March 1, use this formula.

```
IF( MOD((Year(ADDMONTHS(date, 12*num_years))-1960), 4)=0, ADDMONTHS(date, 12*num_years)+1, ADDMONTHS(date, 12*num_years))
```

Add Business Days to a Date

This formula finds three business days from a given *date*.

```
CASE(
    MOD(date - DATE(1900, 1, 7), 7),
    3, date + 2 + 3,
```

```

4, date + 2 + 3,
5, date + 2 + 3,
6, date + 1 + 3,
date + 3
)

```

This formula finds the day of the week of the *date* field value. If the date is a Wednesday, Thursday, or Friday, the formula adds five calendar days, two weekend days, three weekdays, to the date to account for the weekend. If *date* is a Saturday, you need four additional calendar days. For any other day of the week Sunday Tuesday, simply add three days. You can easily modify this formula to add more or fewer business days. The tip for getting the day of the week is useful to use to adjust this formula.

Find the Hour, Minute, or Second from a Date/Time

To get the hour, minute, and second from a Date/Time field as a numerical value, use the following formulas where *TZoffset* is the difference between the user's time zone and GMT. For hour in 24-hour format:

```
VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) )
```

For hour in 12-hour format:

```

IF(
OR(
    VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) = 0,
    VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) = 12
),
12,
    VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) )
-
    IF(
        VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) < 12,
        0,
        12
    )
)
)
```

For minutes:

```
VALUE( MID( TEXT( date/time - TZoffset ), 15, 2 ) )
```

For seconds:

```
VALUE( MID( TEXT( date/time - TZoffset ), 18, 2 ) )
```

And, to get “AM” or “PM” as a string, use:

```
IF(
    VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) < 12,
    "AM",
    "PM"
)
```

To return the time as a string in “HH:MM:SS A/PM” format, use the following formula:

```
IF(
    OR(
        VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) = 0,
        VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) = 12
    ),
    "12",
    TEXT( VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) )
        -
        IF(
            VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) < 12,
            0,
            12
        )
    )
)
& ":" &
MID( TEXT( date/time - TZoffset ), 15, 2 )
& ":" &
MID( TEXT( date/time - TZoffset ), 18, 2 )
& " "
IF(
    VALUE( MID( TEXT( date/time - TZoffset ), 12, 2 ) ) < 12,
    "AM",
    "PM"
)
```

When working with time in formula fields, always consider the time difference between your organization and GMT. See [A Note About Date/Time and Time Zones](#) for more information about the time zone offset used in this formula.

Find the Elapsed Time Between Date/Times

To find the difference between two Date values as a number, subtract one from the other like so:

`date_1 - date_2` to return the difference in days.

Finding the elapsed time between two Date/Time values is slightly more complex. This formula converts the difference between two Date/Time values, `datetime_1` and `datetime_2`, to days, hours, and minutes.

```
IF(
    datetime_1 - datetime_2 > 0 ,
    TEXT( FLOOR( datetime_1 - datetime_2 ) ) & " days "
    & TEXT( FLOOR( MOD( (datetime_1 - datetime_2 ) * 24, 24 ) ) ) & " hours "
    & TEXT( ROUND( MOD( (datetime_1 - datetime_2 ) * 24 * 60, 60 ), 0 ) ) & " mi
nutes",
    ""
)
```

Find the Number of Business Hours Between Two Date/Times

The formula to find business hours between two Date/Time values expands on the formula to find elapsed business days. It works on the same principle of using a reference Date/Time. In this case 1/8/1900 at 16:00 GMT, or 9:00 AM PDT, and then finding your Dates' respective distances from that reference. The formula rounds the value it finds to the nearest hour and assumes an 8-hour, 9:00 AM5:00 PM work day.

```
ROUND( 8 * (
    ( 5 * FLOOR( ( NOW() - DATETIMEVALUE( '1900-01-08 16:00:00' ) ) / 7) +
    MIN(5,
        FLOOR( ( MOD( ( NOW() - DATETIMEVALUE( '1900-01-08 16:00:00' ), 7) / 1) +
        MIN( 1, 24 / 8 * ( MOD( ( NOW() - DATETIMEVALUE( '1900-01-08 16:00:00' ),
        1 ) ) )
    )
)
-
( 5 * FLOOR( ( MQL_datetime_c - DATETIMEVALUE( '1900-01-08 16:00:00' ) ) /
7) +
MIN( 5,
    FLOOR( ( MOD( MQL_datetime_c - DATETIMEVALUE( '1900-01-08 16:00:00' ), 7) / 1) +
    MIN( 1, 24 / 8 * ( MOD( MQL_datetime_c - DATETIMEVALUE( '1900-01-08 16:00:00' ),
    1 ) ) )
)
```

```
)  
) ,  
2 )
```

You can change the eights in the formula to account for a longer or shorter work day. If you live in a different time zone or your work day doesn't start at 9:00 AM, change the reference time to the start of your work day in GMT. See [A Note About Date/Time and Time Zones](#) for more information.

See Also

[Using Date, Date/Time, and Time Values in Formulas](#)

[Examples of Advanced Formula Fields](#)

[Tips for Building Formulas](#)

Sample Discounting Formulas

Use these formulas to calculate discount amounts.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

For details about using the functions included in these samples, see [Formula and Operator Functions](#).

Maintenance and Services Discount

This formula field uses two custom currency fields: **Maintenance Amount** and **Services Amount**. It displays "Discounted" on an opportunity if its maintenance amount and services amount don't equal the opportunity **Amount** standard field value. Otherwise, it displays "Full Price."

```
IF(Maintenance__c + Services__c <> Amount,  
"Discounted",  
"Full Price")
```

Opportunity Discount Amount

This formula calculates the difference of the product Amount less the Discount Amount. Discount Amount is a custom currency field.

```
Amount -  
Discount_Amount__c
```

Opportunity Discount Rounded

Use this formula to calculate the discounted amount of an opportunity rounded off to two digits. This example is a number formula field on opportunities that uses a custom percent field called **Discount Percent**.

```
ROUND(Amount - Amount * Discount_Percent__c, 2)
```

Opportunity Discount with Approval

This formula adds a “Discount Approved” checkbox to an opportunity. It uses conditional logic to check the value of the approval flag before calculating the commission.

```
IF(Discount_Approved__c, ROUND(Amount - Amount * DiscountPercent__c, 2), Amount)
```

Sample Employee Services Formulas

Use these formulas for employee services.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

For details about using the functions included in these samples, see [Formula Operators and Functions](#).

Bonus Calculation

This example determines an employee's bonus amount based on the smallest of two amounts: the employee's gross income times a bonus percent or an equally divided amount of the company's performance amount among all employees. It assumes you have a custom number field for **Number of Employees**, a custom percent field for **Bonus Percent**, and currency custom fields for the employee's **Gross** and company's **Performance**.

```
MIN(Gross__c * Bonus_Percent__c,  
     Performance__c / Number_of_Employees__c)
```

Employee 401K

This example formula determines which amount to provide in employee 401K matching based on a matching program of half of the employee's contribution or \$250, whichever is less. It assumes you have a custom currency field for **Contribution**.

```
MIN(250, Contribution__c /2)
```

Hours Worked Per Week

This formula uses a custom tab to enable time tracking of hours worked per day. It uses a formula field to sum the hours per week.

```
MonHours__c + TuesHours__c + WedsHours__c + ThursHours__c + FriHours__c
```

Total Pay Amount

This formula determines total pay by calculating regular hours multiplied by a regular pay rate, plus overtime hours multiplied by an overtime pay rate.

```
Total Pay =  
IF(Total_Hours__c <= 40, Total_Hours__c * Hourly_Rate__c,  
40 * Hourly_Rate__c +  
(Total_Hours__c - 40) * Overtime_Rate__c)
```

Sample Expense Tracking Formulas

Use these formulas for expense tracking.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

For details about using the functions included in these samples, see [Formula and Operator Functions](#).

Expense Identifier

This formula displays the text `Expense-` followed by trip name and the expense number. This field a text formula field that uses an expense number custom field.

```
"Expense-" &  
    Trip_Name__c & "-" & ExpenseNum__c
```

Mileage Calculation

This formula calculates mileage expenses for visiting a customer site at 35 cents a mile.

```
Miles_Driven__c * 0.35
```

Sample Financial Calculations Formulas

Use these formulas for financial calculations.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

For details about using the functions included in these samples, see [Formula Operators and Functions](#).

Compound Interest

This formula calculates the interest that you have after T years, compounded M times per year.

```
Principal__c * ( 1 + Rate__c / M ) ^ ( T * M )
```

Compound Interest Continuous

This formula calculates the interest that will have accumulated after T years, if continuously compounded.

```
Principal__c * EXP(Rate__c * T)
```

Consultant Cost

This formula calculates the number of consulting days times 1200 given that this formula field is a currency data type and consulting charges a rate of \$1200 per day. **Consulting Days** is a custom field.

```
Consulting_Days__c *  
1200
```

Gross Margin

This formula provides a simple calculation of gross margin. In this formula example, **Total Sales** and **Cost of Goods Sold** are custom currency fields.

```
Total_Sales__c - Cost_of_Goods_Sold__c
```

Gross Margin Percent

This formula calculates the gross margin based on a margin percent.

```
Margin_percent__c * Items_Sold__c * Price_item__c
```

Payment Due Indicator

This formula returns the date five days after the contract start date whenever Payment Due Date is blank. Payment Due Date is a custom date field.

```
(BLANKVALUE(Payment_Due_Date__c, StartDate +5))
```

Payment Status

This formula determines if the payment due date is past and the payment status is “UNPAID.” If so, it returns the text “PAYMENT OVERDUE” and if not, it leaves the field blank. This example uses a custom date field called **Payment Due Date** and a text custom field called **Payment Status** on contracts.

```
IF(  
AND(Payment_Due_Date__c < TODAY(),  
ISPICKVAL(Payment_Status__c, "UNPAID")),  
"PAYMENT OVERDUE",  
null )
```

Sample Image Link Formulas

Use these formulas for image links.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

For details about using the functions included in these samples, see [Formula Operators and Functions by Context](#).

Yahoo! Instant Messenger™ Image

This formula displays an image that indicates whether a contact or user is logged in to Yahoo! Instant Messenger. Clicking the image launches the Yahoo! Instant Messenger window. This formula uses a custom text field called **Yahoo Name** to store the contact or user's Yahoo! ID.

```
IF(ISBLANK(Yahoo_Name__c), "", HYPERLINK("ymsgr:sendIM?" & Yahoo_Name__c, IMAGE("http://opi.yahoo.com/online?u=" & Yahoo_Name__c & "&m=g&t=0", " ")))
```

Flags for Case Priority

This formula displays a green, yellow, or red flag image to indicate case priority.

```
IMAGE(
CASE( Priority,
"Low", "/img/samples/flag_green.gif",
"Medium", "/img/samples/flag_yellow.gif",
"High", "/img/samples/flag_red.gif",
"/s.gif"),
"Priority Flag")
```

Color Squares for Case Age

This formula displays a 30 x 30 pixel image of a red, yellow, or green, depending on the value of a **Case Age** custom number field.

```
IF( Case_Age__c > 20,
IMAGE("/img/samples/color_red.gif", "red", 30, 30),
IF( Case_Age__c > 10,
IMAGE("/img/samples/color_yellow.gif", "yellow", 30, 30),
IMAGE("/img/samples/color_green.gif", "green", 30, 30)
))
```

Traffic Lights for Status

This formula displays a green, yellow, or red traffic light images to indicate status, using a custom picklist field called **Project Status**. Use this formula in list views and reports to create a “Status Summary” dashboard view.

```
IMAGE(
CASE(Project_Status__c,
"Green", "/img/samples/light_green.gif",
"Yellow", "/img/samples/light_yellow.gif",
"Red", "/img/samples/light_red.gif",
"/s.gif"),
"status color")
```

Stars for Ratings

This formula displays a set of one to five stars to indicate a rating or score.

```
IMAGE(
CASE(Rating__c,
"1", "/img/samples/stars_100.gif",
"2", "/img/samples/stars_200.gif",
"3", "/img/samples/stars_300.gif",
"4", "/img/samples/stars_400.gif",
"5", "/img/samples/stars_500.gif",
"/img/samples/stars_000.gif"),
"rating")
```

Consumer Reports™–Style Colored Circles for Ratings

This formula displays a colored circle to indicate a rating on a scale of one to five, where solid red is one, half red is two, black outline is three, half black is four, and solid black is five.

```
IMAGE(
CASE(Rating__c,
"1", "/img/samples/rating1.gif",
"2", "/img/samples/rating2.gif",
"3", "/img/samples/rating3.gif",
"4", "/img/samples/rating4.gif",
"5", "/img/samples/rating5.gif",
"/s.gif"),
"rating")
```

Horizontal Bars to Indicate Scoring

This formula displays a horizontal color bar (green on a white background) of a length that is proportional to a numeric score. In this example, the maximum length of the bar is 200 pixels.

```
IMAGE("/img/samples/color_green.gif", "green", 15, Industry_Score__c * 2) &
IMAGE("/s.gif", "white", 15,
200 - (Industry_Score__c * 2))
```

Sample Integration Link Formulas

Use these formulas for integration links.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

For details about using the functions included in these samples, see [Formula Operators and Functions by Context](#).

Application API Link

This formula creates a link to an application outside Salesforce, passing the parameters so that it can connect to Salesforce via SOAP API and create the necessary event.

```
HYPERLINK ("https://www.myintegration.com?sessionId=" & GETSESSIONID() & "?&rowID="
& Name & "action/CreateTask","Create a Meeting Request")
```

! **Important** `$Api.Session_ID` and `GETSESSIONID()` return the same value, an identifier for the current session in the current context. This context varies depending on where the global variable or function is evaluated. For example, if you use either in a custom formula field, and that field is displayed on a standard page layout in Salesforce Classic, the referenced session is a basic Salesforce session. That same field (or the underlying variable or formula result), when used in a Visualforce page, references a Visualforce session instead. Session contexts are based on the domain of the request. That is, the session context changes whenever you cross a hostname boundary, such as from `.salesforce.com` to `.vf.force.com` or `.lightning.force.com`. Session identifiers from different contexts, and the sessions themselves, are different. When you transition between contexts, the new session replaces the previous one, and the previous session is no longer valid. The session ID also changes at this time. Normally Salesforce transparently handles session hand-off between contexts, but if you're passing the session ID around yourself, you must reaccess `$Api.Session_ID` or `GETSESSIONID()` from the new context to ensure a valid session ID. Not

all sessions are created equal. In particular, sessions obtained in a Lightning Experience context have reduced privileges, and don't have API access. You can't use these session IDs to make API calls. `{ !$Api.Session_ID }` isn't generated for guest users.

Shipment Tracking Integration

This formula creates a link to FedEx, UPS, or DHL shipment tracking websites, depending on the value of a **Shipping Method** custom picklist field. The parameters shown in this example for FedEx, UPS, and DHL websites are illustrative and don't represent the correct parameters for all situations.

```
CASE(Shipping_Method__c,
  "Fedex",
  HYPERLINK("http://www.fedex.com/Tracking?ascend_header=1&clienttype
 =dotcom&cntry_code=us&language=english&tracknumbers= "& tracking_id__c,"Trac
 k"),
  "UPS",
  HYPERLINK("http://wwwapps.ups.com/WebTracking/processInputRequest?HTMLVersion
 =5.0&sort_by=status&loc=en_US&InquiryNumber1= "& tracking_id__c & "&track.x=3
 2&track.y=7", "Track"),
  "DHL",
  HYPERLINK("http://track.dhl-usa.com/TrackByNbr.asp?ShipmentNumber=" & trackin
 g_id__c,"Track"), "")
```

Skype™ Auto Dialer Integration

This formula creates a linkable phone number field that automatically dials the phone number via the Skype VOIP phone application. It requires installation of the Skype application (a third-party product not provided by Salesforce) on your desktop.

```
HYPERLINK("callto://+" & Country_Code__c & Phone_Unformatted__c, Phone)
```

Sample Lead Management Formulas

Use these formulas to manage leads.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: All Editions

For details about using the functions included in these samples, see [Formula Operators and Functions](#).

Lead Aging (for open leads)

This formula checks to see if a lead is open and if so, calculates the number of days it has been open by subtracting the date and time created from the current date and time. The result is the number of days open rounded to zero decimal places. If the lead isn't open, this field is blank.

```
IF(ISPICKVAL(Status,  
"Open") , ROUND(NOW()-CreatedDate, 0) , null)
```

Lead Data Completeness

This formula calculates the percent of certain lead fields that your sales personnel enter. The formula field checks the values of two custom number fields: **Phone** and **Email**. If the fields are empty, the formula returns the value “0.” The formula returns a value of “1” for each field that contains a value and multiplies this total by fifty to give you the percentage of fields that contain data.

```
(IF(Phone = "", 0, 1) + IF>Email = "", 0, 1) ) * 50
```

Lead Numbering

This formula returns a number value for the text value in the auto-number field **Lead Number**, which can be useful if you want to use the **Lead Number** field in a calculation, such as round-robin or other routing purposes. Auto-number fields are text fields and must be converted to a number for numeric calculations.

```
VALUE(Lead_Number__c)
```

Round-Robin Assignment of Cases or Leads

The following formula example for leads assumes you have three lead queues and you want to assign an equal number of incoming leads to each queue. You can also assign cases using a similar formula.

```
MOD(VALUE(Lead_Number__c) ,  
3)
```

This formula is for a custom formula field named Round_Robin_ID that assigns each lead a value of 0, 1, or 2. This formula uses a custom auto-number field called **Lead Number** that assigns each lead a unique number starting with 1. The MOD function divides the lead number by the number of lead queues available (three in this example) and returns a remainder of 0, 1, or 2. Use the value of this formula field in your lead assignment rules to assign lead records to different queues. For example:

- Round_Robin_ID = 0 is assigned to Queue A
- Round_Robin_ID = 1 is assigned to Queue B
- Round_Robin_ID = 2 is assigned to Queue C

Sample Metric Formulas

Use these formulas for metric temperature and metric unit of measure conversion.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

For details about using the functions included in these samples, see [Formula Operators and Functions by Context](#).

Temperature Conversion

This formula converts Celsius degrees to Fahrenheit.

```
1.8 * degrees_celsius__c + 32
```

Unit of Measure Conversion

This formula converts miles to kilometers.

```
Miles__c * 1.60934
```

Sample Opportunity Management Formulas

Use these formulas for business expenses and earnings.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

For details about using the functions included in these samples, see [Formula Operators and Functions](#).

Expected Product Revenue

This formula calculates total revenue from multiple products, each with a different probability of closing.

```
ProductA_probability__c * ProductA_revenue__c + ProductB_probability__c * ProductB_revenue__c
```

Maintenance Calculation

This formula calculates maintenance fees as 20% of license fees per year. **Maintenance Years** is a custom field on opportunities.

```
Amount * Maint_Years__c * 0.2
```

Monthly Subscription-Based Calculated Amounts

This formula calculates an opportunity amount based on a monthly subscription rate multiplied by the subscription period.

```
Monthly_Amount__c * Subscription_Months__c
```

Monthly Value

This formula divides total yearly value by 12 months.

```
Total_value__c / 12
```

Opportunity Additional Costs

This formula calculates the sum of the product **Amount**, maintenance amount, and services fees. **Maint amount** and **Service Fees** are custom currency fields.

```
Amount + Maint_Amount__c +  
Services_Amount__c
```

Opportunity Categorization

This formula uses conditional logic to populate an **Opportunity category** text field, based on the value of the **Amount** standard field. Opportunities with amounts less than \$1500 are “Category 1,” opportunities

with amounts between \$1500 and \$10,000 are “Category 2,” and the rest are “Category 3.” This example uses nested IF statements.

```
IF(Amount < 1500, "Category 1", IF(Amount > 10000, "Category 3", "Category 2"))
```

Opportunity Data Completeness

This formula takes a group of fields and calculates what percent of them are being used by your personnel. This formula field checks five fields to see if they’re blank. If so, a zero is counted for that field. A “1” is counted for any field that contains a value, and this total is divided by five (the number of fields evaluated). This formula requires you to select the **Treat blank fields as blanks** option under Blank Field Handling while the Advanced Formula subtab is showing.

```
(IF(ISBLANK(Maint_Amount__c), 0, 1) +  
 IF(ISBLANK(Services_Amount__c), 0, 1) +  
 IF(ISBLANK(Discount_Percent__c), 0, 1) +  
 IF(ISBLANK(Amount), 0, 1) +  
 IF(ISBLANK(Timeline__c), 0, 1)) / 5
```

Opportunity Expected License Revenue

This formula calculates expected revenue for licenses based on the probability of closing.

```
Expected_rev_licenses__c * Probability
```

Opportunity Revenue Text Display

This formula returns the expected revenue amount of an opportunity in text format without a dollar sign. For example, if the **Expected Revenue** of a campaign is “\$200,000,” this formula field displays “200000.”

```
TEXT(ExpectedRevenue)
```

Opportunity Total Deal Size

This formula calculates the sum of maintenance and services amounts.

```
Amount + Maint_Amount__c + Services_Amount__c
```

Opportunity Total Price Based on Units

This formula generates proposal pricing based on unit price and total volume.

```
Unit_price__c * Volume__c * 20
```

Professional Services Calculation

This formula estimates professional service fees at an average loaded rate of \$1200 per day. **Consulting Days** is a custom field on opportunities.

```
Consulting_Days__c * 1200
```

Stage-Based Sales Document Selection

This formula Identifies a relevant document in the Documents tab based on opportunity **Stage**. Use document IDs in the form of "00I3000000j7AO."

```
CASE(StageName,  
    "Prospecting", "Insert 1st Document ID",  
    "Qualification", "Insert 2nd Document ID",  
    "Needs Analysis", "Insert 3rd Document ID",  
    "Value Proposition", ...  
)  
)
```

Sales Coach

This formula creates a hyperlink that opens a stage-specific document stored in the Documents tab. It uses the previously defined custom formula field that identifies a document based on opportunity **Stage**. See [Stage-Based Sales Document Selection](#).

```
HYPERLINK("/servlet/servlet.FileDownload?file=" & Relevant_Document__c, "View  
Document in New Window")
```

Shipping Cost by Weight

This formula calculates postal charges based on weight.

```
package_weight__c * cost_lb__c
```

Shipping Cost Percentage

This formula calculates shipping cost as a fraction of **total amount**.

```
Ship_cost__c / total_amount__c
```

Tiered Commission Rates

This formula calculates the 2% commission amount of an opportunity that has a probability of 100%. All other opportunities have a commission value of zero.

```
IF(Probability = 1,  
ROUND(Amount * 0.02, 2),  
0)
```

Total Contract Value from Recurring and Non-Recurring Revenue

This formula calculates both recurring and non-recurring revenue streams over the lifetime of a contract.

```
Non_Recurring_Revenue__c + Contract_Length_Months__c * Recurring_Revenue__c
```

Sample Pricing Formulas

Use these formulas for total amounts and user pricing.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

For details about using the functions included in these samples, see [Formula Operators and Functions by Context](#).

Total Amount

This formula calculates a total amount based on unit pricing and total units.

```
Unit_price__c * Total_units__c
```

User Pricing

This formula calculates a price per user license.

```
Total_license_rev__c / Number_user_licenses__c
```

Sample Scoring Calculations Formulas

Use these formulas for lead scoring and customer success scoring.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

For details about using the functions included in these samples, see [Formula Operators and Functions by Context](#).

Lead Scoring

This formula scores leads, providing a higher score for phone calls than website requests.

```
CASE(LeadSource, "Phone", 2, "Web", 1, 0)
```

Here's a formula that scores a lead based on his or her rating:

```
CASE(1, IF(ISPICKVAL(Rating, "Hot"), 1, 0), 3, IF(ISPICKVAL(Rating, "Warm"), 1, 0), 2, IF(ISPICKVAL(Rating, "Cold"), 1, 0), 1))
```

Customer Success Scoring

This formula uses a simple scoring algorithm to rank customers a high score for positive survey results in Salesforce.

```
Survey_Question_1__c * 5 + Survey_Question_2__c *2
```

Formulas: How Do I ... ?

A collection of topics around formulas, including common math calculation, text functions, and more.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

Some How Do I's are not relevant to **Database.com**

USER PERMISSIONS NEEDED

To view formula field details:	View Setup and Configuration
To create, change, or delete formula fields:	Customize Application

Common Math Calculations

- Add numbers?
- Convert text into a number
- Divide numbers?
- Multiply numbers?
- Round numbers?
- Subtract numbers?

Common Text Functions

- Check if a field contains specified text?
- Check if a picklist contains a specified value?
- Combine first and last names?
- Convert numbers into text?
- Create a hyperlink field?

Advanced Formulas

- Calculate Commission Amounts for Opportunities?
- Set Up Round-Robin Assignment of Cases or Leads?
- Set Up Opportunity Discount Rounded?

Custom Summary Formulas for Reports

- Calculate the sum of all leads that have Email Opt Out and Do Not Call fields selected?

- Calculate the difference of all Amount fields and all Discounted Amount fields on opportunities?
- Calculate the average of all opportunities?
- Calculate what percent of all opportunities are closed won
- Calculate the number of active Salesforce users to the 2nd power for administration?
- Calculate the duration of all activities (minutes) times the number of records per 24 hours?
- Calculate the average percent margin on a product-by-product level across many opportunities?
- Calculate the percentage of one product compared to all products in closed opportunities?
- Calculate the change in revenue from opportunities between months?

Cross-Object Formulas

- Display a Percent field from a parent object?
- Display a text field from a parent object?
- Display a phone number field from a parent object?
- Display a picklist field from a parent object?
- Display a URL field from a parent object?

Common Formula Errors

Review common errors that can occur with formulas and how to fix them.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

- “#Error!” displays for a formula field whenever an error occurs while calculating the value of a formula. To resolve the error, check your formula.
 - Is the formula dividing by zero? If so, check if the denominator of your expression is zero and provide an alternative value. For example, the following campaign formula field is blank if the number of opportunities is zero:

```
IF (NumberOfOpportunities > 0,
    NumberOfWonOpportunities / NumberOfOpportunities, null)
```

- Is the formula calculating a value larger than the maximum value of the current type? If so, you can append L to numeric values to make them Long so the intermediate products will be Long and no overflow occurs. For example, the following example shows how to correctly compute the amount of milliseconds in a year by multiplying Long numeric values.

```
Long MillsPerYear = 365L * 24L * 60L * 60L * 1000L;
                                         Long ExpectedValue
= 31536000000L;
                                         System.assertEquals
```

```
s(MillsPerYear, ExpectedValue);
```

- Is the formula calculating the square root of a negative number? If so, use an IF function similar to the one above to check if the value is a positive number.
- Is the formula calculating the LOG of a negative number? If so, use an IF function similar to the one above to make sure that the number is positive.
- Is the formula using the VALUE function with text that contains special characters? For examples of special characters, see [Formula Operators and Functions](#).
- Make sure the formula does not contain a HYPERLINK function within a text function, such as `LEFT(HYPERLINK("http://MYCOMPANY.ORG ", "MYCOMPANY ") , 5)`.
- Is the formula disabled or referencing a disabled formula field? Salesforce disables formula fields when they are deleted and they remain disabled after they are restored. To enable disabled formula fields, edit and save the field. For more information on deleted custom fields and restoring them, see [Manage Deleted Custom Fields](#).
- “#Too Big!” displays if your formula output is over 18 digits. When this happens, check your formula for calculations that could result in more than 18 digits. Avoid multiplying large numbers, raising a large number to a power, or dividing by a very small number.
- CASE functions return an error whenever any of the expressions return an error, regardless of which one should be returned. For example, `CASE(Field__c,"Partner", "P", "Customer", "C", LEFT(Field__c, -5))` returns an error even if the value of the field is “Partner” or “Customer” because the last statement is illogical.
- Prevent division by zero errors by including an IF function that determines if the value of a field is zero. For example, `IF(Field__c =0,0, 25/Field__c)`.

Get an Explanation for a Formula and Fix Errors with Einstein for Formulas

Use Einstein for Formulas to get an explanation for a formula and fix syntax errors in a formula used in Formula fields, default field values, and record validation rules. As an admin, you can use Einstein’s assistance for existing formulas or for the new ones that you create.

REQUIRED EDITIONS

Einstein Generative AI is available in Lightning Experience. Setup for Einstein Generative AI is available in Lightning Experience.

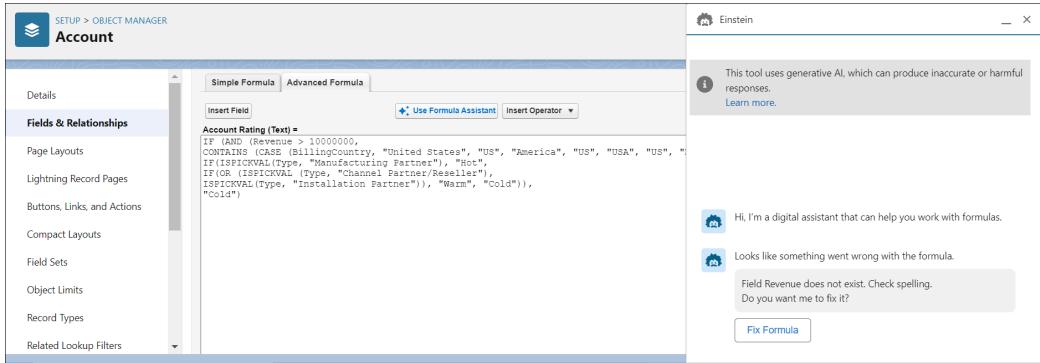
Available in: **Enterprise, Performance, and Unlimited** Editions

Your organization must license Einstein for Formulas to use the feature. Contact your Salesforce account executive for more information.

In the following scenario, see how Einstein for Formulas can help you fix a formula field.

1. Turn on Einstein Generative AI. See [Enable Einstein Generative AI](#).
2. From Setup, in the Quick Find box, enter *Object Manager*, and then select **Object Manager**. Next, click any object. For example, select **Account**.

3. Select **Fields and Relationships**, click any existing formula field, and then click **Edit**. You can view the existing formula in the Formula Editor.
 4. To get assistance for this formula, click **Use Formula Assistant** on top of the Formula Editor. The Einstein panel opens.
 5. If the formula is correct, Einstein asks if you want to receive an explanation for the formula. Click **Explain Formula** to proceed.
 6. If there are any syntax errors in the formula, Einstein identifies the error and suggests a fix for the formula. Click **Fix Formula** to view the fixed formula.



- a. To copy the updated formula in the Formula Editor, click **Use Formula**.

 **Note** The languages supported for the generative response include English, French, German, Italian, Japanese, and Spanish. If your user's language isn't one of the supported languages, responses are generated in English.

7. (Optional) To provide feedback to Einstein about the generated explanations and formula fixes, use the thumbs-up and thumbs-down icons in the panel. If the generated explanation or the suggested fix for a formula is accurate, use the thumbs-up icon, or use the thumbs-down icon to select and submit feedback.

Generate Emails From Records

A merge field is a field you can put in an email template, mail merge template, custom link, or formula to incorporate values from a record. For example, you can place a merge field in an email template so that the greeting includes the recipient's name rather than a generic "Hello!".

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

The available merge fields vary according to which Salesforce edition you have.

You can use merge fields within custom formula fields, s-controls, custom links, custom buttons, Visualforce pages, and when you create email or mail merge templates.

Merge field names are determined when you create a custom field or object. Field Name is automatically populated based on what you type into Field Label. You can customize this field if you want, but keep in mind that the name must:

- Only use underscores and alphanumeric characters
- Begin with a letter and end with a letter
- Not include spaces
- Not contain two consecutive underscores

 **Important** Ensure that the custom field name and label are unique for that object.

- If a standard and custom field have identical names or labels, the merge field displays the custom field value.
- If two custom fields have identical names or labels, the merge field can display an unexpected value.

If you create a field label called `Email` and a standard field labeled `Email` exists, the merge field is unable to distinguish between the fields. Adding a character to the custom field name makes it unique. For example, `Email12`.

To find the merge field name for an object or field in Salesforce, visit the object or field's detail page and refer to Field Name.

To incorporate merge fields, use the editor in the respective feature. Salesforce provides valid merge fields in each editor for all related standard and custom objects. If you're using the Connect for Office Word add-in to create mail merge templates, you see a complete list of valid merge fields to insert.

Merge Field Syntax

A merge field's syntax can vary depending on where you use the field. To make sure that you use the correct syntax, select merge fields from the dropdown list in the editor where you use the merge field.

Merge Fields for Validation Rules

A merge field is a field you can put in an email template, mail merge template, custom link, or formula to incorporate values from a record.

Merge Fields for Formulas

A merge field is a field you can put in an email template, mail merge template, custom link, or formula to incorporate values from a record.

Merge Fields for Cross-Object Formulas

A *Cross-object formula* is a formula that spans two related objects and references merge fields on those objects.

Merge Field Tips

Here are a few pointers for getting the most out of merge fields.

Merge Field Syntax

A merge field's syntax can vary depending on where you use the field. To make sure that you use the correct syntax, select merge fields from the dropdown list in the editor where you use the merge field.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

The available merge fields vary according to which Salesforce edition you have.

Custom objects and fields are always appended with `__c` when referenced. The object precedes field labels, and all spaces convert to underscores. For example, `Account.CreatedDate` references the **Created Date** standard field for the account object.

In standard relationships, the name of the relationship is the master object. For example, you can reference the account name from a contact validation rule using `Account.Name`. You can reference the phone number of the account creator from an opportunity product formula field using `Opportunity.Account.CreatedBy.Phone`. In custom relationships, the name of the relationship is the value specified in **Field Name** with `_r` appended to it. For example, you can reference contact email from a custom object validation rule using `Contact__r.Email`.

Merge Fields for Validation Rules

A merge field is a field you can put in an email template, mail merge template, custom link, or formula to incorporate values from a record.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Essentials, Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Syntax and Formatting

When you insert a merge field in a validation rule, the syntax consists of the object, a period, and the field name. For example, `$User.State` corresponds with a user's state or province.

A merge field's syntax can vary depending on where you're using the field. To make sure you're using the correct syntax, select merge fields from the drop-down list in the editor where you're using the merge field. The merge fields for validation rules correspond directly with the fields in your app.

For a list of fields in an object, from the management settings for the object, go to the fields section.

Important

- If two or more custom objects have matching names or labels, only one of the objects appears when you select from available merge fields. Make sure that all custom objects have unique

names and labels so that you can select merge fields from any of the objects.

Limitations

Validation rules can't reference merge fields for:

- Auto number fields, such as Requisition Number
 - Compound fields, such as addresses, first and last names, dependent picklists, and dependent lookups
-  **Note** Validation rules can reference merge fields individual address fields, such as **Billing City**.
- Campaign statistic fields, including statistics for individual campaigns and campaign hierarchies

Tips

- Some merge fields display as radio buttons but function like picklist fields when referenced in a formula.

Use the values "Read," "Edit," and "None" in a formula when referencing:

- \$UserRole.CaseAccessForAccountOwner
- \$UserRole.OpportunityAccessForAccountOwner
- CaseAccessLevel (on Territory)
- OpportunityAccessLevel (on Territory)

Use the values "Read," "Edit," and "All" in a formula when referencing:

- AccountAccessLevel (on Territory)

- Use the **RecordType.Id** merge field in your formula to apply different validations for different record types.

See Also

[Find Object Management Settings](#)

Merge Fields for Formulas

A merge field is a field you can put in an email template, mail merge template, custom link, or formula to incorporate values from a record.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

Syntax and Formatting

Merge fields for formulas aren't enclosed in curly braces or preceded by an exclamation point, nor are they preceded by the type of record. For example: `AccountNumber`. To ensure you're using the correct syntax, use the Insert Field button or the drop-down list in the formula editor.

See Also

- [Tips for Using Merge Fields in Formulas](#)
- [Build a Formula Field](#)

Merge Fields for Cross-Object Formulas

A *Cross-object formula* is a formula that spans two related objects and references merge fields on those objects.

REQUIRED EDITIONS

 **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: Salesforce Classic

Available in: all editions

A merge field is a field you can put in an email template, mail merge template, custom link, or formula to incorporate values from a record.

A cross-object formula can reference merge fields from a master ("parent") object if an object is on the detail side of a master-detail relationship. A cross-object formula also works with lookup relationships. For example, you can write a cross-object formula that references the Account Name for a contact associated with a case. In this example, you would type `Contact.Account.Name` in a formula on the Case object.

Syntax and Formatting

Merge fields for formulas aren't enclosed in curly braces or preceded by an exclamation point. Use the relationship names of the objects, not the labels. Although the relationship name is often the same as the object name, it is technically the field name of the relationship field.

To reference the parent account name from Account object, the syntax is `Parent.Name`, not `Account.Name`. When referencing a custom object, add two underscores and the letter *r* to its name. For example, `Position__r.title__c` references the **Job Title** field (`title__c`) on a Position custom object.

Limitations

You can't reference:

- Merge fields for objects related to activities. For example, merge fields for contacts and accounts are not available in task and event formulas.
- The `$RecordType` global variable—it only resolves to the record containing the formula, not the record to which the formula spans. Starting with the Spring '13 release, when you create a new formula the `$RecordType` global variable is only available for default value formulas.

The value of the `Profile.Name` merge field differs depending on the context of the cross-object formula field that references it. On detail pages, the value is the profile name, as expected. In list views and reports, the value is the internal value of the associated profile instead. If you use `Profile.Name` in a formula, use it within an `OR` function to ensure that the formula always returns the intended result. For example:

```
IF  
    (OR  
        (LastModifiedBy.Profile.Name = "Standard User", LastModifiedBy.Profi  
le.Name = "PT2"),  
        "Standard", "Not Standard")
```

None of the above applies to profile names referenced by the `$Profile` global variable.

Merge Field Tips

Here are a few pointers for getting the most out of merge fields.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

The available merge fields vary according to which Salesforce edition you have.

To make sure you're using the correct syntax, select merge fields from the merge field picker.

- To use a merge field as the destination of a link, insert the merge field after `http://`.
- Salesforce rounds decimal values referenced in merge fields on email templates to three digits regardless of locale.
- You can store the name of an account, contact, or lead in your organization's default language (the local name), in addition to the account or user's default language (the standard name). If the local name is blank, the standard merge field name is used.
- To reference a standalone file, use `$Resource.<resource_name>`, where `<resource_name>` is the name you specified when you uploaded the resource.

- If you're using the Translation Workbench to translate custom field names, users can look up merge fields in their chosen language.
- You can't use a lookup field as a merge field in an email template. You can, however, create a hidden formula field on the page layout that pulls the value from the lookup field. Then include the hidden field in the email template.

Build Your Own Salesforce App

An *app* is a collection of items that work together to serve a particular function. Salesforce apps come in two flavors: Classic and Lightning. Classic apps are created and managed in Salesforce Classic. Lightning apps are created and managed in Lightning Experience. You can customize both types of app to match the way your users work.

REQUIRED EDITIONS

Available in: Salesforce Classic, Lightning Experience, and the Salesforce mobile app

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To view apps: View Setup and Configuration

To manage apps: Customize Application

The platform includes innovative point-and-click app-building tools that give you the power to customize Salesforce to meet the needs of your business. You can also build your own apps to share and store information that is important to you. You don't need any programming knowledge to use these tools. You can find them in Salesforce Classic Setup by selecting **Create**, and in Lightning Experience Setup by entering *App* in the Quick Find box to get started.

The platform also includes app building tools that require some programming knowledge. You can find tools that require advanced programming knowledge in Salesforce Classic Setup by selecting **Develop**, and in Lightning Experience Setup by entering *Custom Code* in the Quick Find box.

Classic apps are a collection of standard and custom tabs, including:

- Most standard objects, including Home, the main Chatter feed, Groups, and People
- Your org's custom objects
- Visualforce tabs
- Lightning component tabs
- Canvas apps via Visualforce tabs
- Web tabs

Lightning apps are a collection of items that include everything from the Classic apps list, plus Lightning

page tabs, and utilities like Sales Dialer. In Lightning apps, you can customize the app's logo and enhance its branding by customizing the color of the navigation bar.

You can also upgrade Classic apps to Lightning apps in Lightning Experience, but the two versions of the app must then be managed separately in their own environments.

Salesforce provides standard apps such as Sales and Service.

You can also build your own on-demand apps by grouping items into new custom apps. A custom app consists of a label, a description, and an ordered list of items, which often includes tabs. You can also add custom logos and branding to your custom apps.

In Salesforce Classic, custom apps are listed in the Lightning Platform app menu, which is a dropdown list displayed at the top of every page.



In Lightning Experience and the Salesforce mobile app, you can find your available custom apps in the App Launcher (grid icon). In Lightning Experience, to see all your available Salesforce apps and items, click **View All**.



When you choose an app, your screen changes to reflect the contents of that app. For example, if you switch from an app that contains Opportunities to another app that doesn't, the Opportunities item disappears. In addition, the app might display a different *default landing* tab when selected.

Apps are associated with profiles. Profiles control which tabs you can see or hide, as well as which apps are available to you.

Lightning Apps

With apps in Lightning Experience, members of your org can work more efficiently by easily switching between apps. Users can open apps you've created from the App Launcher. What's most important to sales reps? Accounts, events, and organizations. How about sales managers? Reports and dashboards make the top of the list. Lightning apps take things to another level past Classic apps by letting you brand your apps with a custom color, logo, and utility bar.

Tips for Creating Apps in Lightning Experience

It's time for the fun part: deciding how to set up Lightning apps for your users. Here are some tips for planning Lightning apps for your org.

Create Lightning Apps

As in Salesforce Classic, you can create apps in Lightning Experience, but with even more bells and whistles. You can brand and customize Lightning apps to help your users work more efficiently. For example, you can create a Lightning app for your finance department that includes all important items, including tabs, for users to complete common tasks. You can customize the navigation bar color, brand it with a logo, and make the app available in the App Launcher for the user profiles associated with the finance department.

Customize Lightning Apps with the Lightning App Builder

When you edit a Lightning app from the App Manager in Setup, you're brought into the Lightning App Builder to manage the app's settings. Update app branding, navigation, and other options, and manage the Lightning pages assigned to that app all in one place.

Add a Utility Bar to Lightning Apps

The utility bar is a specialized type of Lightning page that gives your users quick access to common productivity tools, like Notes and Recent Items. It appears as a fixed footer that users can access to open utilities in docked panels. Some utilities support pop-out, which lets them open in a new browser window.

Lightning App Navigation Bar Items

Most of the items that appear in the App Launcher can appear in a Lightning app navigation bar. To add items to an app's navigation bar, you can use the Lightning app creation wizard, which lets you choose from a list of available items.

Upgrade Classic Apps to Lightning Apps

You can upgrade a Classic app to a Lightning app in Lightning Experience, enhancing it for your Lightning Experience users with a customized color, logo, utility bar, and more items like Lightning pages supported in the navigation bar.

Salesforce App Considerations

Keep these considerations in mind when working with apps in either Lightning Experience or Salesforce Classic.

Create Custom Apps for Salesforce Classic

Create custom apps to give your Salesforce Classic users access to everything they need all in one place.

Subtab Apps in Salesforce Classic

An *app* is a group of tabs that work as a unit to provide application functionality. Similarly, a *subtab app* is a collection of tabs that appears on the Chatter profile page. A subtab app can include both default and custom tabs.

Lightning Apps

With apps in Lightning Experience, members of your org can work more efficiently by easily switching between apps. Users can open apps you've created from the App Launcher. What's most important to sales reps? Accounts, events, and organizations. How about sales managers? Reports and dashboards make the top of the list. Lightning apps take things to another level past Classic apps by letting you brand your apps with a custom color, logo, and utility bar.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Lightning apps contain everything you expect from a custom app, such as custom and standard objects,

and custom tabs. But Lightning apps can also include Lightning page tabs and utilities like Lightning Voice. The navigation model of Lightning apps is optimized for efficiency, with actions on certain items like Opportunities in the navigation bar. You can even append navigation items to a Lightning app that you found on AppExchange and installed from a managed package.

Custom apps from Salesforce Classic automatically work in Lightning Experience and can be upgraded to Lightning apps. Classic apps appear in the list of apps in Setup alongside your Lightning apps, and are available from the App Launcher as long as their **Show in Lightning Experience** attribute is enabled.

However, the reverse isn't true for Lightning apps. Lightning apps aren't available in Salesforce Classic.

You can assign multiple user profiles to multiple apps. Also, you can assign as many user profiles to one app as you need to. For instance, you have several groups involved with inside sales. Assign all the groups to your inside sales app, and they all have access to it.

To switch between apps, users can use the App Launcher. This makes it easy for users to switch contexts and still have access to the items, objects, and pages they need most.

You can view all the apps in your org from the App Manager. In Lightning Experience Setup, enter *App* in the Quick Find box, then select **App Manager**.

See Also

[Create Lightning Apps](#)

[Upgrade Classic Apps to Lightning Apps](#)

[Salesforce App Considerations](#)

Tips for Creating Apps in Lightning Experience

It's time for the fun part: deciding how to set up Lightning apps for your users. Here are some tips for planning Lightning apps for your org.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Professional, Performance, Unlimited, and Developer** Editions

The best time to create Lightning apps is when you're rolling out Lightning Experience. So make creating Lightning apps a part of your rollout strategy. Check out the Trailhead module "[Lightning Experience Rollout](#)" for many great ideas to help you make a smooth transition.

Talk to your users. Ask them what their priorities are. Customizing tabs in apps gives you a unique opportunity to engage with your users. Each group of users has its own priorities. Find out which objects and items represent their highest priorities.

- Ask users to post feedback to a Chatter group.
- Publish polls.

- Schedule lunch sessions. Everyone likes a free lunch, and nearly everybody is happy to express their opinion.

Create a master list of objects that everyone in your org wants. Then trim down the list for each group—sales reps, sales managers, execs, and so on. The menus for every user group share some common objects, like Home, Tasks, and Feed. Keep the high-priority items for each group at the top. Put low-priority items at the bottom, or remove them altogether. Users can always go to the App Launcher to get the items they use less often.

Create Lightning Apps

As in Salesforce Classic, you can create apps in Lightning Experience, but with even more bells and whistles. You can brand and customize Lightning apps to help your users work more efficiently. For example, you can create a Lightning app for your finance department that includes all important items, including tabs, for users to complete common tasks. You can customize the navigation bar color, brand it with a logo, and make the app available in the App Launcher for the user profiles associated with the finance department.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To view apps: View Setup and Configuration

To manage apps: Modify All Data

1. From the Home tab in Setup, in the Quick Find box, enter *App*, and then select **App Manager**.
2. To use an existing custom app as the basis for a new custom app, find the source app in the list, select **Clone** from its action menu, and then go through the Lightning app creation wizard.
This feature is only available for custom Lightning apps. It isn't available for standard, connected, managed, community, or classic apps.
3. To create an app from scratch, click **New Lightning App**, and walk through the New Lightning App wizard.

Here are some things you can do in the wizard.

- Give your app a name, set its primary color, and give it a logo. The app description displays alongside the icon in the App Launcher. Make sure that the description is meaningful to your users.
- Choose whether to override a custom theme's brand image and navigation bar color with the brand image and color from the app.
- Choose which type of navigation to use—standard or console.
- Choose which form factors your app is available for.
- Add a utility bar for common processes and tools, like Recent Items, Notes, Dialer, and Open CTI.

- Customize which items appear in the app's navigation bar.
 - Assign the app to user profiles.



Note If you add more than 50 default navigation items to an app, your users can't personalize the app's navigation bar.

When organizing the navigation bar, the item at the top of the list becomes your app's landing page on desktop and mobile.

The order of items in the navigation bar also determines the default objects shown in the Top Results page on the search results page. After the user interacts with the app for 15 days, Top Results reflects the user's most frequently used objects. If the user doesn't have access or permissions to the app, Top Results includes the Account, Contact, Opportunity, Case, Lead, People (User), and Group objects until the user's most frequently used objects are determined.

See Also

Lightning Apps

Create and Edit a Custom Lightning Console App

Add a Utility Bar to Lightning Apps

Salesforce App Considerations

Customize Lightning Apps with the Lightning App Builder

When you edit a Lightning app from the App Manager in Setup, you're brought into the Lightning App Builder to manage the app's settings. Update app branding, navigation, and other options, and manage the Lightning pages assigned to that app all in one place.

REQUIRED EDITIONS

Lightning App Builder is available in: both Salesforce Classic and Lightning Experience

Lightning apps are available in: Lightning Experience

Available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS NEEDED

To manage apps:	Modify All Data
To create and save Lightning pages in the Lightning App Builder:	Customize Application
To view apps:	View Setup and Configuration

1. From the Home tab in Setup, enter *App* in the Quick Find box, then select **App Manager**.
 2. Click on a Lightning app's row, and select **Edit**.

Here are some things you can do from the App Settings tab.

- Change your app's name, primary color, and logo.
- Override the org theme with your app's brand image and nav bar color.
- Add a utility bar for common processes and tools, like Recent Items, Notes, Dialer, and Open CTI.
- Manage which default items appear in the app's navigation bar.
- Manage user profile assignments.

On the Navigation Items tab, you can control which objects and other items—like Visualforce pages or Lightning components—are included in your app. To create a custom object by importing data from a spreadsheet, click **Create** at the top of the Available Items section.

3. Click the **Pages** tab in the Lightning App Builder header to see all the active Lightning pages assigned to the app, create pages, and open existing pages, even ones not associated with the app.

 **Tip** If you create pages for your app, when finished, don't forget to click **Activation** to make them active for your users and assign them to the app.

See Also

- [Lightning Apps](#)
- [Create Lightning Apps](#)
- [Get Help for Lightning App Builder](#)

Add a Utility Bar to Lightning Apps

The utility bar is a specialized type of Lightning page that gives your users quick access to common productivity tools, like Notes and Recent Items. It appears as a fixed footer that users can access to open utilities in docked panels. Some utilities support pop-out, which lets them open in a new browser window.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To view apps: View Setup and Configuration

To manage apps: Modify All Data

Utilities harness the power of Lightning components. When you set up a utility bar, you select which Lightning components to use as utilities.

Background utility items are added the same way as normal utility items, but don't appear in the utility bar. The  icon appears next to background utility items on the utility item list. If you have only background utility items in your utility bar, the utility bar doesn't appear in your app. You need at least

one non-background utility item in your utility bar for it to appear.

You can add or edit a utility bar at any time.

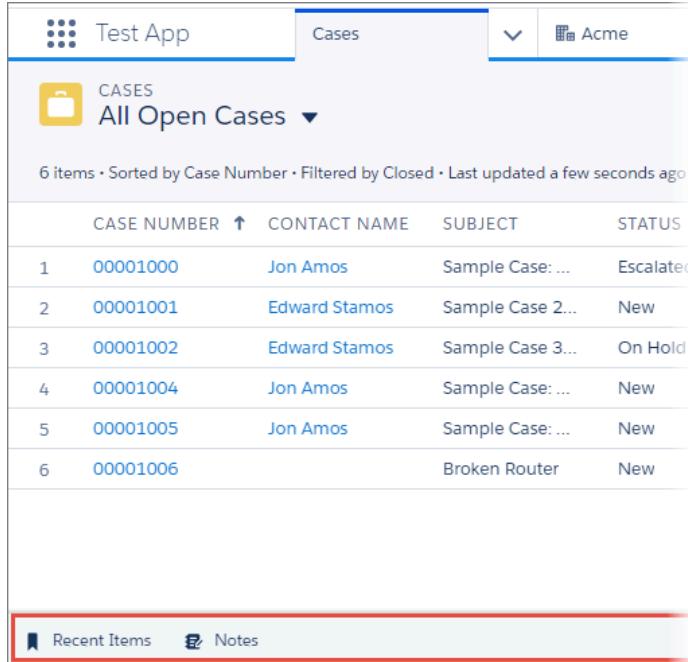
1. From the Home tab in Setup, enter *App* in the Quick Find box, then select **App Manager**.
2. To edit or add a utility bar to an existing app, click **Edit** in the dropdown menu next to your app. To create a Lightning app with a utility bar, click **New Lightning App**.
3. Click the **Utility Items** tab and add the utilities you want.

Specify component and utility properties, like the height and width of the utility panel, and what label and icon to display in the utility bar. Some utilities have properties that can't be changed.

 **Tip** A Lightning page region can contain up to 100 components. We recommend adding no more than 10 utilities, and that you keep the utility labels short and sweet. You want your users to quickly find the tools and processes they need most.



Example Here's a utility bar with the Recent Items and Notes utilities.



The screenshot shows a Salesforce Lightning App interface. At the top, there are three tabs: 'Test App' (selected), 'Cases' (disabled), and 'Acme' (disabled). Below the tabs is a section titled 'CASES' with a sub-section 'All Open Cases'. It displays a table with 6 items, sorted by Case Number and filtered by Closed status. The table columns are CASE NUMBER, CONTACT NAME, SUBJECT, and STATUS. The data is as follows:

CASE NUMBER	CONTACT NAME	SUBJECT	STATUS
1 00001000	Jon Amos	Sample Case: ...	Escalated
2 00001001	Edward Stamos	Sample Case 2...	New
3 00001002	Edward Stamos	Sample Case 3...	On Hold
4 00001004	Jon Amos	Sample Case: ...	New
5 00001005	Jon Amos	Sample Case: ...	New
6 00001006		Broken Router	New

At the bottom of the screen, there is a utility bar with two items: 'Recent Items' and 'Notes'. The 'Recent Items' button is highlighted with a red border.

When creating a utility bar for your app, keep these things in mind:

- Utility bars created using the Lightning App Wizard or in the Lightning App Builder can be assigned to only one Lightning app. However, utility bars created using the API can be assigned to multiple Lightning apps.
- The utility bar doesn't support Visualforce pages or components.
- The utility bar doesn't fully support the Chatter Publisher and Feed components.
- The History utility works in Lightning console apps only.
- The Omni-Channel utility works in the Lightning Service Console app only.
- The default utility bar alignment matches the user's language setting alignment. For example, English is read left to right. If you select Default and a user's language is set to English, the utility bar appears at the bottom of the left side of the screen. If you select Mirrored, the utility bar appears at the bottom

of the right side of the screen.

See Also

[Customize Your Lightning Console App with Utilities](#)

[Create Lightning Apps](#)

[Lightning Web Components Dev Guide: Configure a Component for the Utility Bar](#)

[Salesforce Console Developer Guide: Using Background Utility Items](#)

[Salesforce Console Developer Guide: Using Pop-Out Utilities](#)

Lightning App Navigation Bar Items

Most of the items that appear in the App Launcher can appear in a Lightning app navigation bar. To add items to an app's navigation bar, you can use the Lightning app creation wizard, which lets you choose from a list of available items.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Professional, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To view navigation menus: View Setup and Configuration

To create navigation menus: Customize Application

The list of available items contains only those items in your org that are eligible for Lightning app navigation bars, which includes:

- Most standard objects, including Home, the main Chatter feed, Groups, and People
- Your org's custom objects and apps
- Visualforce tabs
- Lightning component tabs
- Lightning page tabs
- Canvas apps via Visualforce tabs
- Web tabs

If the Lightning app was installed from a managed package, you can append navigation items below the original set of navigation items included in the Lightning app, which are locked.

 **Note** You can't add Connected apps like Gmail™ and Microsoft® Office 365™ to the navigation bar. Users can continue to access them from the App Launcher.

Upgrade Classic Apps to Lightning Apps

You can upgrade a Classic app to a Lightning app in Lightning Experience, enhancing it for your Lightning Experience users with a customized color, logo, utility bar, and more items like Lightning pages supported in the navigation bar.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To view apps:	View Setup and Configuration
---------------	------------------------------

To manage apps:	Modify All Data
-----------------	-----------------

 **Note** You can't upgrade a Salesforce Classic console app to Lightning Experience. You can choose to display or hide the app in the Lightning Experience App Launcher, but you can't edit the app from the App Manager page in Lightning Experience Setup. To get started in Lightning Experience, customize these Salesforce-provided Lightning console apps: Service Console and Sales Console. You can also recreate your Salesforce Classic console app in Lightning Experience, but using the Salesforce out-of-the-box app is faster and easier.

1. From the Home tab in Setup, enter *App* in the Quick Find box, then select **App Manager**.
2. Find the Classic app that you want to upgrade in the apps list. A checkmark in the Visible in Lightning Experience column means that the app is accessible in Lightning Experience via the App Launcher and is fully functional. Even though a Classic app works in Lightning Experience, it doesn't take advantage of all the benefits of being a Lightning app. That's why we recommend that you upgrade it.
3. Click , and select **Upgrade**.
4. Review the app properties and update them if necessary.
If you have custom Lightning Home pages assigned to profiles in your org, and the app you're upgrading is visible in Lightning Experience, you see a checkbox that lets you apply those Home page profile assignments to the upgraded app. Selecting the checkbox ensures that users see the custom Home page assigned to their profile when they're working in the upgraded app, and that you can modify those Home page assignments if you need to.
5. Click **Upgrade**.
Your Classic app is copied and upgraded for Lightning Experience. You now have two versions of the app: a Classic version, and a Lightning version. After you upgrade it, the Classic app is no longer accessible in Lightning Experience via the App Launcher. You still see the Classic app in the apps list, but with the Visible in Lightning column deselected.

The two versions of your app now must be managed separately. Future changes you make to the Classic app won't be reflected in the Lightning version of the app, and vice versa. You can toggle the availability

of your Classic apps in Lightning Experience by selecting or deselecting **Show in Lightning Experience** on the Classic app's detail page.

See Also

[Lightning Apps](#)

[Salesforce App Considerations](#)

Salesforce App Considerations

Keep these considerations in mind when working with apps in either Lightning Experience or Salesforce Classic.

REQUIRED EDITIONS

Available in: Salesforce Classic, Lightning Experience, and the Salesforce mobile app

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

General

- You can delete custom apps, but not standard apps. Deleting a custom app removes it from the apps menu and the App Launcher, but doesn't delete any associated objects. If you created objects for an app, consider deleting them as well.
- Salesforce Classic console apps are custom apps.
- You can't change an app's type—such as standard to connected or vice versa—after you create it.
- For Salesforce Platform and Salesforce Platform One license users, the Platform standard app is the only app listed in the Lightning Platform app menu and the Lightning Experience App Launcher. For details about specifying a unique label for the Platform standard app in Salesforce Classic, see [Create Custom Apps for Salesforce Classic](#).
- To assign apps to user profiles in Professional Edition, you must have user profiles enabled for your org.
- You must prefix a namespace to a custom app API that is retrieved from a managed package before you deploy it.

Classic Apps

Consider these requirements when choosing a custom app logo for a Classic app from the document library:

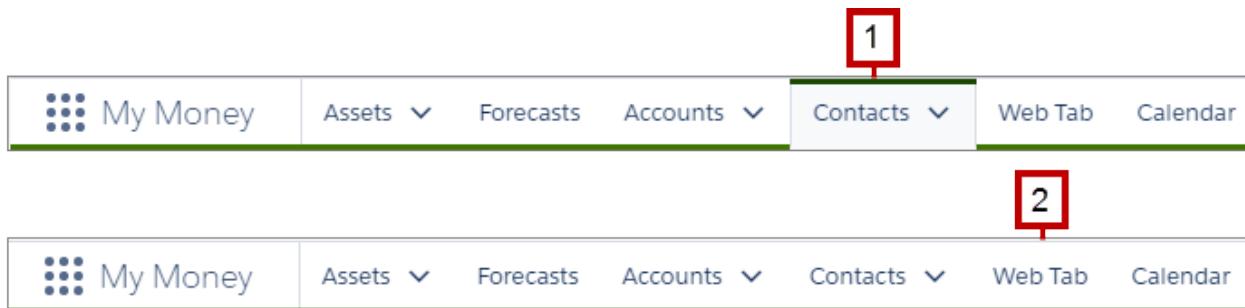
- The image must be in GIF or JPEG format and less than 20 KB.
- If the image is larger than 300 pixels wide by 55 pixels high, then it is scaled to fit.
- For the best on-screen display, we recommend that you use an image with a transparent background.
- The Externally Available checkbox must be selected on the document's properties so that users can view the image.

Lightning Apps

- The number of Lightning Apps you can create in an org varies by edition.

Edition	Lightning Apps Limit
Professional Edition	10
Enterprise Edition	260
Unlimited Edition	Unlimited

- A Lightning app's description displays in the App Launcher, so we recommend that you keep the description concise.
- Users can't remove the items you include in the navigation bar, and they can't personalize the navigation bar when it contains more than 50 items. For example, if you include 32 items in an app's navigation bar, users can add 18 more personal items.
- Consider these requirements when choosing a custom app image for apps in Lightning Experience:
 - App images represent your app in both Lightning Experience and the Salesforce app.
 - Choose a JPG, PNG, BMP, or GIF image that's smaller than 5 MB.
 - For best results, upload an image that's 128 by 128 pixels. Images larger than the maximum display of 128 by 128 pixels are automatically resized.
- Even if you haven't selected the option to override themes in the App Manager, your app's brand image and color always override the Lightning Lite and Lightning Blue themes.
- If you restrict an app to one type of device, only users viewing the app on that device can access it. For example, if you assign your app to the Phone form factor, your desktop users don't see the app in the App Launcher. Only your Salesforce mobile app users can see it.
- Not all objects that appear in the App Launcher can appear in an app, but it's easy to figure out which ones can. When you start the wizard from the Lightning Experience App Manager, you see all available items for a navigation bar.
- Navigation items in a Lightning app installed from a managed package are locked. You can't remove or reorder them. However, you can append other navigation items so that they're accessible in the Lightning app.
- You can create records and access recent records and lists for certain items directly from the navigation bar. Items with  next to their name support this feature, with a few exceptions. Tasks and Notes allow you to create a record but you can't access recent records or lists. Reports and Dashboards allow you to see recent records but you can't see recent lists or create a record.
- Some tabs, such as web tabs and Visualforce tabs, aren't highlighted when you select them on the navigation bar. For example, when you select **Contacts**, the tab is highlighted (1). However, when you select a web tab, the page displays but the tab isn't highlighted (2).



Create Custom Apps for Salesforce Classic

Create custom apps to give your Salesforce Classic users access to everything they need all in one place.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To view apps: View Setup and Configuration

To manage apps: Modify All Data

If you're new to custom apps, we recommend using [Lightning Platform quick start](#) to create an app. With this tool, you can generate a basic working app in just one step.

If you've already created the objects, tabs, and fields you need for your app, follow these steps. With this option, you create an app label and logo, add items to the app, and assign the app to profiles.

1. From Setup, enter *Apps* in the Quick Find box, then select **Apps**.
2. Click **New**.
3. If the Salesforce console is available, select whether you want to define a custom app or a Salesforce console.
4. Give the app a name and description.
An app name can have a maximum of 40 characters, including spaces.
5. Optionally, brand your app by giving it a custom logo.
6. Select which items to include in the app.
7. Optionally, set the default landing tab for your new app using the Default Landing Tab dropdown menu below the list of selected tabs.
This determines the first tab a user sees when logging into this app.
8. Choose which profiles the app will be visible to.
9. Select the **Default** box to set the app as that profile's default app, meaning that new users with the profile see this app the first time they log in.

Profiles with limits are excluded from this list.

10. Click **Save**.

[Create Apps in Salesforce Classic with App Quick Start](#)

App quick start is a fast way to create a basic Classic app in just one step.

[App Quick Start: Next Steps for Configuring Apps in Salesforce Classic](#)

After you've created a basic working app with app quick start in Salesforce Classic, build out the app with more objects and fields, define its access settings, and add users to share your app with them.

See Also

[Build Your Own Salesforce App](#)

[Salesforce App Considerations](#)

Create Apps in Salesforce Classic with App Quick Start

App quick start is a fast way to create a basic Classic app in just one step.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create apps:

Modify All Data

AND

Manage Profiles and Permission Sets

- From Setup, enter *Apps* in the Quick Find box, then select **Apps**, and click **Quick Start**. Alternatively, from the Lightning Platform Home page, click **Add App** under **Getting Started**, or **App Quick Start** under **Quick Links**.
- Enter the information needed for your app.

Field Name	Description
App Label	The app's name that appears in the Lightning Platform app menu. The label can have a maximum of 40 characters, including spaces.
Plural Label	The plural name of the object. This name appears on the tab.
Singular Label	A name used to refer to the object in any user

Field Name	Description
	interface pages.
Gender	If it's appropriate for your org's default language, specify the gender of the label. This field appears if the org-wide default language expects gender.
Starts with a vowel sound	If it's appropriate for your org's default language, enable this option if your label should be preceded by "an" instead of "a".

3. Click **Create**.
4. On the **You're All Set!** page, click **here** to add new fields to your app.
5. To see your app as it will appear to users, click **Go To My App**.

The app quick start:

- Generates an app label and API name (a unique name that's used to refer to the object when using the Lightning Platform API).
- Generates an object label and API name.
- Generates a tab label, and associates the tab with the object.
- Enables feed tracking for the object. Feed tracking lets people follow records of that object type and see Chatter feed updates.
- Enables access to the app and tab in your user profile. Any users who have the Modify All Data permission can also access the object.
- Generates a permission set that grants access to the new custom object.
- Assigns the permission set to the user who creates the app.

 **Note** If you're in a custom app, only the tabs included in the app appear and include the Create button.

After you've created an app, you can extend it with more components, specify access settings, and add users to your org.

See Also

- [App Quick Start: Next Steps for Configuring Apps in Salesforce Classic](#)
- [Salesforce App Considerations](#)

App Quick Start: Next Steps for Configuring Apps in Salesforce Classic

After you've created a basic working app with app quick start in Salesforce Classic, build out the app with more objects and fields, define its access settings, and add users to share your app with them.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create objects, tabs, fields, and validation rules: Customize Application

To create users: Manage Internal Users

To create profiles and permission sets: Manage Profiles and Permission Sets

1. Build out your app with the basic components used in apps.

- Create objects, which are custom database tables that allow you to store information specific to your app.
- Create tabs that are associated with the objects you've created.
- For each object, create fields to store the information that's important to your org.
- Create validation rules, which verify that the data users enter meets the standards you specify before they save a record.

For quick shortcuts to these tools, use the [Lightning Platform quick access menu](#), which is available from object list view pages and record detail pages.

2. Create user profiles or permission sets. These are collections of settings and permissions that determine what users can do in an app.
3. Specify the types of access that users will have to the app.
 - a. Make your app visible using profiles or permission sets.
 - b. Make your object tabs visible.
 - c. Set the object permissions for the objects you created.
4. Add users to your org. When adding users, be sure to assign them the appropriate profiles or permission sets you created so they can access your app.

See Also

[Create Apps in Salesforce Classic with App Quick Start](#)

[Salesforce App Considerations](#)

Subtab Apps in Salesforce Classic

An *app* is a group of tabs that work as a unit to provide application functionality. Similarly, a *subtab app* is a collection of tabs that appears on the Chatter profile page. A subtab app can include both default and custom tabs.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Users can see different sets of tabs on the profile page depending on their context. Subtab apps are the various sets of tabs available on specific pages, such as users' profile pages.

These default subtab apps determine which tabs display, depending on the user's context.

Subtab App	Displayed to the user when viewing...
Profile (Others)	Another user inside their internal org
Profile (Self)	Their own profile inside their internal org
Profile in Communities (Others)	Another user while inside a community. It's shown only if Communities is enabled.
Profile in Communities (Self)	Their own profile inside a community. It's shown only if Communities is enabled.

End users can't customize the display of subtab apps. Administrators can hide tabs within subtab apps using the Tab Hidden option in Tab Settings. Users can see tabs set to Default Off and Default On.

[Manage Subtab Apps in Salesforce Classic](#)

You can view and customize the subtab apps on users' profile pages.

[Control Subtab App Visibility in Salesforce Classic](#)

After you configure subtab apps, you can specify which users can see specific tabs on the profile page.

Manage Subtab Apps in Salesforce Classic

You can view and customize the subtab apps on users' profile pages.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To view apps: View Setup and Configuration

To manage apps: Modify All Data

From Setup, in the Quick Find box, enter *Apps*, and then select **Apps** to show your org's subtab apps.

You can do the following:

- To view details for a subtab app, click the name in the **Subtab Apps** section. This section displays the subtab app properties, such as which tabs are part of the app, including any tabs that aren't yet deployed. To view details, click custom tabs in the Included Tabs list.
 - To change the properties of a subtab app, click **Edit** to choose the tabs to include in the subtab app, change their display order, and set the Default Landing Tab.
-  **Note** Administrators can change permission sets or profile settings to limit users' access to each tab. This way, administrators can make specific tabs available to some users but not to others.

See Also

[Subtab Apps in Salesforce Classic](#)

[Control Subtab App Visibility in Salesforce Classic](#)

Control Subtab App Visibility in Salesforce Classic

After you configure subtab apps, you can specify which users can see specific tabs on the profile page.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To view apps:	View Setup and Configuration
To manage apps:	Customize Application

To control the visibility of tabs within a subtab app:

1. From Setup, in the Quick Find box, enter *Profiles*, and then select **Profiles**.
2. Do one of the following:
 - Original profile user interface—Click **Edit** next to the profile you want to modify and scroll to the Tab Settings section.
 - Enhanced profile user interface—Click the profile you want to modify and click **Object Settings**. Click the object you want to modify and click **Edit**.

 **Note** Some profiles, including Chatter External and Chatter Free users, don't have the permissions to view subtab apps.

3. Change the tab settings.
End users can't customize the display of subtab apps. Administrators can hide tabs within subtab apps using the Tab Hidden option in Tab Settings. Users can see tabs set to Default Off and Default On.
4. (Original profile user interface only) To reset users' tab customizations to the tab visibility settings that you specify, select **Overwrite users' personal tab customizations**.
5. Click **Save**.

See Also

- [Subtab Apps in Salesforce Classic](#)
- [Manage Subtab Apps in Salesforce Classic](#)

Get Help for Lightning App Builder

The Lightning App Builder is a point-and-click tool that makes it easy to create custom pages for the Salesforce mobile app and Lightning Experience, giving your users what they need all in one place. The Lightning App Builder is also a one-stop shop for configuring Lightning apps.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

You can access the [Lightning App Builder](#) from Setup by entering *Lightning App Builder* in the Quick Find box and then selecting **Lightning App Builder**.

With the Lightning App Builder, you can build:

- Single-page apps that drill down into standard pages
- Dashboard-style apps, such as apps to track top sales prospects or key leads for the quarter
- “Point” apps to solve a particular task, such as an expense app for users to enter expenses and monitor expenses they’ve submitted
- Custom record pages for your objects, tailored to the needs of your users
- Custom Home pages containing the components and features that your users use most
- Custom forecasts pages containing components and features that give sales leaders information about planned revenue.



But that's not all. When you edit a Lightning app from the App Manager in Setup, you're brought into the Lightning App Builder to manage the app's settings. You can update the app's branding, navigation, app options, and manage the Lightning pages assigned to that app all inside the Lightning App Builder.

The Lightning App Builder supports the same browsers as Lightning Experience and isn't supported on mobile browsers. The minimum recommended resolution for the Lightning App Builder is 1280x1024.

See Also

- [Create an App Home Page with the Lightning App Builder](#)
- [Create and Configure Lightning Experience Record Pages](#)
- [Supported Browsers and Devices for Lightning Experience](#)
- [Lightning Aura Components Developer Guide](#)

[Lightning Web Components Developer Guide](#)

Deciding When to Use Lightning App Builder or the Page Layout Editor

The Lightning App Builder helps you customize Lightning Experience pages. The page layout editor helps you customize Salesforce Classic record page layouts and some aspects of Lightning pages, depending on the page's configuration. Each tool has its own unique capabilities. To help you decide when to use one tool instead of the other, review the uses each one supports.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Page layouts are available in: All Editions

Lightning App Builder is available in:

Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions

To determine which tool is better suited for you, consider whether you're working in Salesforce Classic or Lightning Experience. If the answer is Salesforce Classic, then the page layout editor is the right tool for you. Lightning App Builder can customize only Lightning record pages, not Classic record pages.

If you're in Lightning Experience, the next thing to consider is how the fields on the record page are being delivered. For example, let's say you're on an object record page, and you want to reorder or change the fields that appear on the page. You go to Setup () and select **Edit Page**, and the record page opens in the Lightning App Builder. Take a close look at where the record fields are located on the page.

- If the page uses the Record Detail component to show record fields, those fields come from the object's page layout in one big block. The only way to adjust the presence or placement of the fields inside a Record Detail component is by using the [page layout editor](#).



Note If you select the Record Detail component on the Lightning App Builder canvas, you see an option in the component properties to upgrade the page to use [Dynamic Forms](#) instead. By upgrading to Dynamic Forms you can manage the record fields right inside the Lightning App Builder.

- If the page uses Dynamic Forms, where the record fields are present as individual components on the canvas, then you can adjust them right in the Lightning App Builder without having to go to the page layout.

When You Want To...	Page Layouts	Lightning Record Pages
Get real-time page performance feedback	-	

When You Want To...	Page Layouts	Lightning Record Pages
Support non-LWC-enabled objects	✓	✓ Some dynamic features of Lightning record pages, such as Dynamic Forms, aren't supported on non-LWC-enabled object record pages. For a list of LWC-enabled objects, see LWC Migration for Record Home Pages .
Set page visibility based on record type or user profile	✓	✓
Set page visibility based on app	-	✓
Set page visibility based on form factor (desktop or phone)	-	✓
Customize page structure using standard templates Some dynamic features of Lightning record pages, such as Dynamic Forms, aren't supported with custom page templates.	-	✓
Put page elements such as related lists, record fields, and highlights anywhere on a page	-	✓
Add custom Lightning components (LWC and Aura)	-	✓
Spread page elements across multiple tabs	-	✓
Hide section headers and collapse the space between two sections	✓	-
Show or hide individual fields and page elements based on criteria you set	-	✓

When You Want To...	Page Layouts	Lightning Record Pages
Add blank spaces to align fields	✓	✓
Make fields required or read-only	✓	✓
Add cross-object fields	–	✓
Add conditional formatting to record fields	–	✓
Display fields in two columns in narrow regions	✓	–
Indicate which fields can be inline-edited in object home list views	✓	–
To support inline edit, a field must be present on at least one page layout, and that page layout must be assigned to the user's profile.		
Customize actions and fields in the Highlights Panel	✓	✓
Customize the buttons and actions that appear on the page	✓	✓
Add or remove related lists	✓	✓
Customize related list columns and buttons	✓	<p>✓</p> <p>You can use the Dynamic Related Lists component in the Lightning App Builder to add, remove, and reorder related list columns, and to add related list buttons. Not all related lists are supported for use with Dynamic Related Lists, however. See Create Dynamic Related Lists in Lightning App Builder.</p>
Add custom links	✓	–

See Also

[Dynamic Highlights Panel](#)

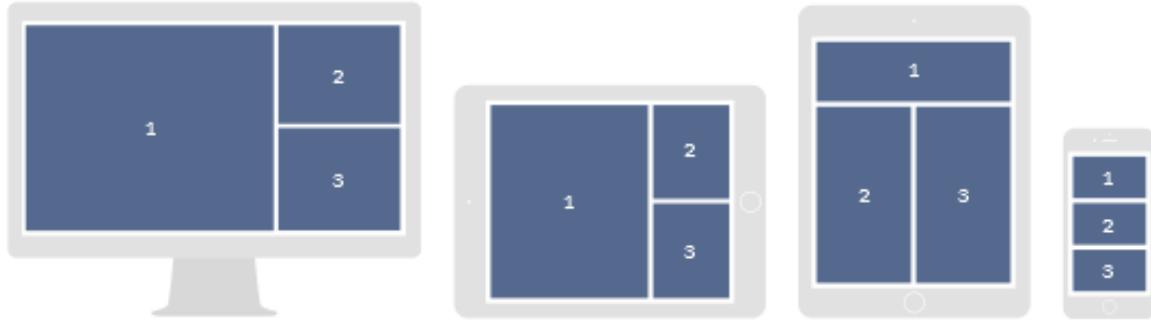
[Align Fields Horizontally in Dynamic Form Field Sections](#)

Lightning Pages

A Lightning page is a custom layout that lets you design pages for use in the Salesforce mobile app or Lightning Experience.

Lightning pages occupy a middle ground between page layouts and Visualforce pages. Like a page layout, Lightning pages allow you to add custom items to a page. However, these items, instead of being fields or Visualforce components, are Lightning components, which allow more flexibility.

The structure of a Lightning page adapts for the device it's viewed on. The template you choose when creating the page controls how it displays on a given device. The Lightning page's template divides the page into regions.

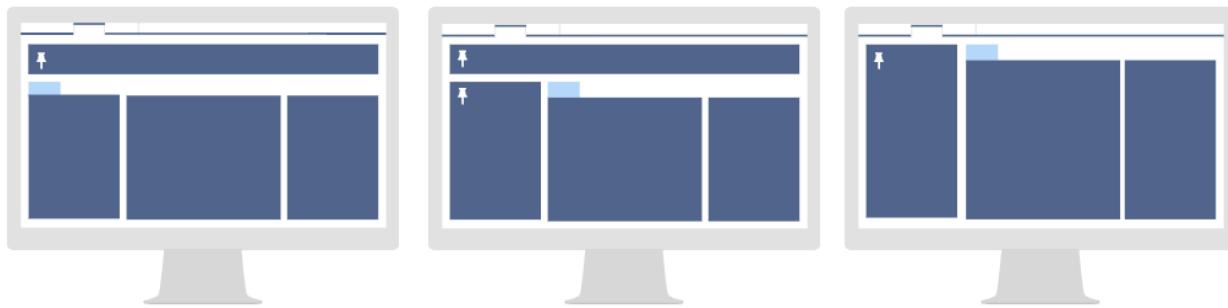


Lightning pages are built using Lightning components—compact, configurable, and reusable elements that you can drop into regions of the page in the Lightning App Builder.

You can use a Lightning page to create an app page that you can add to the navigation bar of a Lightning app, which makes it appear when that app is viewed in both Lightning Experience and the Salesforce mobile apps. An app page gives your users quick access to the objects and items that are most important in that app.

You can also use a Lightning page to create a customized Home page for Lightning Experience, or a custom record page for Lightning Experience and the Salesforce mobile app. And if you've integrated Salesforce with Microsoft® Outlook® or Gmail™, you can create a custom Email Application pane.

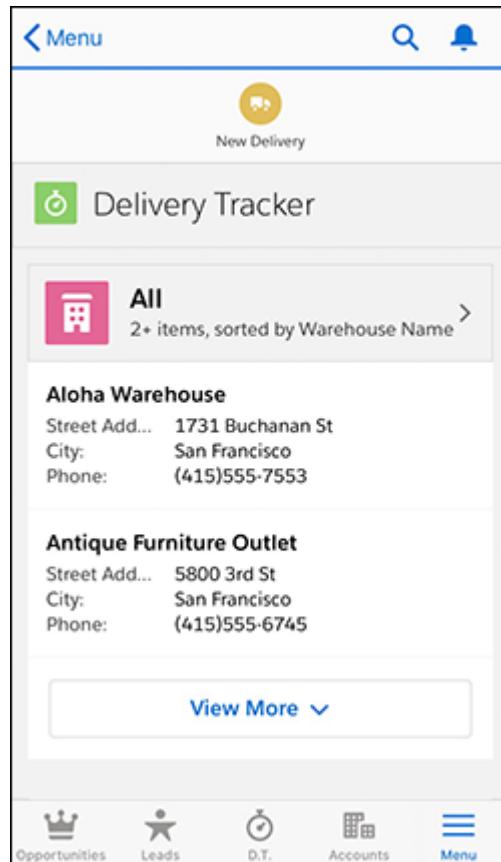
If you have a console app, you can create a Lightning page with pinned regions to let your users view and work with records while navigating between subtabs.



Lightning pages support these components:

- Standard Components—Standard components are Lightning components built by Salesforce.
- Custom Components—Custom components are Lightning components that you or someone else have created. With some configuration, custom Lightning components can work in the Lightning App Builder.
- Third-Party Components on AppExchange—The AppExchange provides a marketplace for Lightning components. You can find packages containing components already configured and ready to use in the Lightning App Builder.

 **Example** This Lightning app page in the Salesforce mobile app has a list view component, a recent items component, and one global action.



Lightning Page Types

You can create different types of Lightning pages with the Lightning App Builder. After you set a Lightning page's type, you can't change it.

App Page

App pages are supported both in the Salesforce mobile app and Lightning Experience.

Use an app page to create a home page for a third-party app that you can add directly into the Salesforce mobile app and Lightning Experience navigation menus. Your users then have an app home page where they can quickly access the most important objects and items.

Add global actions to an app page to enhance its functionality. Global actions allow a user to do things from your app page, such as add call details, create and update records, send email, and create a task. When a user visits a Lightning page in the Salesforce mobile app, the page's actions appear in its action bar. In Lightning Experience, actions appear in the highlights panel at the top of the page.

 **Important** Create actions in the full Salesforce site before adding them to your app page.

App pages support only global actions. Standard Chatter actions, such as Post, File, Link, and Poll, aren't supported.

When a user visits an app page in the Salesforce mobile app or Lightning Experience, only the actions that you specify for the page are displayed. If you haven't specified any actions, no actions appear.

Home Page

Create Home pages with features relevant to specific types of users, and assign the customized pages to different apps or app-and-user-profile combinations. Custom Home pages are supported in Lightning Experience only.

Forecasts Page

Create custom forecasts pages to include all the information that sales leaders require to drive accurate forecasts that support your unique business. The Forecasts Page type is available when Salesforce Forecasting is enabled, and is supported for desktop in Lightning Experience only.

Omni Supervisor Page

Create custom Omni Supervisor tabs to present customized information to supervisors. Configure tabs, including custom tabs, on the Supervisor Configurations page. Custom Omni Supervisor tabs are supported in Lightning Experience and with Enhanced Omni-Channel only.

Record Page

With a record page, you can create a customized version of an object's record page, tailoring it to your users' needs. Custom record pages are supported in Lightning Experience and the Salesforce mobile app.

-  **Note** Actions you see on Lightning Experience record pages and the Home page are taken from object and global page layouts. You can't add, edit, or remove actions on these pages using the Lightning App Builder.

Email Application Pane

Create custom email application panes to let users work with Salesforce content that's most relevant to them in Microsoft® Outlook® and Gmail™. Custom email application panes are supported in Salesforce Classic and Lightning Experience.

See Also

- [Create an App Home Page with the Lightning App Builder](#)
[Create and Configure Lightning Experience Record Pages](#)

Lightning Page Templates

Lightning page templates are Lightning components that have been configured to serve as templates for custom Lightning pages. Page templates can support different form factors, such as desktop or phone. When you create a Lightning page in the Lightning App Builder, you can select a page template that matches the device you're designing the page for.

REQUIRED EDITIONS

Lightning App Builder available in: both Salesforce Classic and Lightning Experience

Lightning Home and utility bar pages available in: Lightning Experience

Lightning app and record pages available in: both the Salesforce mobile app and Lightning Experience

Email application pane pages available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Home page templates are desktop-only. Standard app page templates support both desktop and phone. Standard record page templates support both desktop and phone, except the pinned region templates, which are desktop only. If a form factor isn't on the options list when you try to assign a Lightning page, then the page template doesn't support it.

When working on a page in the Lightning App Builder, you can use the form factor switcher to preview

what the page looks like on different devices that the template supports.

Custom Lightning page template components are supported for record pages, app pages, and Home pages. You can create custom page templates only in Aura. See [Create a Custom Lightning Page Template Component](#) in the *Lightning Aura Components Developer Guide*.

When switching your Lightning record pages to a different template, keep these considerations in mind.

- You can't switch to a template that supports a smaller scope of devices than the original template. For example, if you have a page using a template that supports both desktop and phone, you can't switch the page to use a template that supports only phone or only desktop. When you choose to switch, the list of available templates reflects this rule and shows only the templates that you can switch to.
- If you switch to a page template that doesn't support the form factor of a component on your page, that component is dropped from the page at run time.
- You can't switch from a non-pinned region template to a pinned region template.

Standard Lightning Page Components

Standard components are Lightning components built by Salesforce. Several standard components are available when creating Lightning pages.

REQUIRED EDITIONS

Lightning App Builder available in: both Salesforce Classic and Lightning Experience

Lightning Home and utility bar pages available in: Lightning Experience

Lightning app and record pages available in: both the Salesforce mobile app and Lightning Experience

Email application pane pages available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

 **Note** The components listed here are supported across all page types, except where otherwise indicated. This list doesn't show all standard components available in the Lightning App Builder, only components with special considerations. Other standard components are available and vary based on the type of page you create and which object the page is associated with.

Some standard components have required properties that you must configure for the component to work on the page. When you add a component to a Lightning page in the Lightning App Builder, its required properties are marked with an asterisk.

A Lightning page region can contain up to 100 components.

If you have a page to be used in the Salesforce mobile app, use only the Lightning components that the mobile app supports.

Accordion

Use the Accordion component to organize your components into collapsible sections. You can put multiple components in each section and customize the section name. You can have up to 25 sections, but we recommend no more than 10.

The Accordion component is supported for record and Home pages.

-  **Note** Programmatic versions of the accordion components don't provide the same functionality as their App Builder counterparts. For example, `lightning-accordion` and `lightning:accordion` components don't currently support lazy loading.

Actions & Recommendations

Give your users a to-do list in the Actions & Recommendations component. Show a variety of steps, including screen flows, field service mobile flows, quick actions, and recommendations from your Next Best Action strategies. When a user opens a flow in a console app, it starts in a subtab. In a navigation app, it starts in a window. You can add this component to supported object pages in Lightning Experience.

For more information, see [Lightning Flow for Service and the Actions & Recommendations Component](#).

Activities

Display the timeline of the activities of the record. This component appears only if the Default Activities View is set to Activity Timeline. The component can contain Create a Record quick actions that point to the Event and Task objects. It contains Log A Call and Send Email actions for each of your org's accounts, contacts, contracts, leads, opportunities, and activity-enabled custom object records.

-  **Note** On the Case object, the component only shows the timeline of the activities. Actions appear in a Chatter Publisher component.

For more information, see [Activities View and Actions in Lightning Experience](#).

After Conversation Work (Beta)

Display a countdown after a customer conversation ends. During the After Conversation Work (ACW) countdown period, support agents can complete closing tasks like sending follow-up emails, updating a case, or finalizing their notes. Agents can exit the countdown early by closing the call record or let it run its course. When the time runs out, the agent is considered available to help the next customer whether they've closed the call record.

After Conversation Work is available only for Voice Call channels. To learn more, see [Configure After Conversation Work \(Beta\)](#).

App Launcher

The App Launcher displays a user's available Salesforce apps and the connected apps the administrator has configured. Users navigate between apps using the App Launcher.

This component is supported in API version 35.0 and later.

Contact Salesforce to enable the App Launcher component for the Lightning App Builder in your organization.

Blank Space

Use this component to help align fields in field sections on Lightning record pages that use Dynamic Forms. Find this component under the Fields tab in the component palette.

You can use a blank space on its own to separate fields vertically inside a field section. To further enhance the alignment of your field section content, you can also use a blank space in conjunction with the field section property Align fields horizontally.

For example, these two versions of the same field section contain account fields, with a blank space added below the Phone field. The left image shows the normal behavior without the Align fields horizontally checkbox selected. The blank space appears with a single row height.

On the right, the horizontal alignment setting was selected, causing the Website field to stay aligned horizontally with Parent Account instead of collapsing upward. The blank space height is now expanded to align with the row height of its neighbor, Billing Address.

The image shows two side-by-side screenshots of a Salesforce account record page. Both screenshots have a header labeled 'Account Information' with a downward arrow. Below the header, there are several field sections separated by vertical lines. In the first screenshot (left), the 'Website' field (containing 'http://edgecomm.com') is positioned directly below the 'Billing Address' field. In the second screenshot (right), the 'Website' field is aligned horizontally with the 'Parent Account' field, and the blank space between them is larger, matching the height of the 'Billing Address' field.

Account Information	
Account Owner Madison Rigsby	Rating Hot
Account Name Edge Communications	Phone (512) 757-6000
Billing Address 312 Constitution Place Austin, TX 78767 USA Austin, TX	Website http://edgecomm.com
Parent Account	Ownership Public
Account Number CD451796	Type Customer - Direct
Account Site	Industry Electronics
Annual Revenue \$139,000,000	

Account Information	
Account Owner Madison Rigsby	Rating Hot
Account Name Edge Communications	Phone (512) 757-6000
Billing Address 312 Constitution Place Austin, TX 78767 USA Austin, TX	Website http://edgecomm.com
Parent Account	Ownership Public
Account Number CD451796	Type Customer - Direct
Account Site	Industry Electronics
Annual Revenue \$139,000,000	

A blank space component is considered a field and counts against the limit of 100 fields per region. It doesn't count against the component page limit.

Blank spaces aren't supported in the Dynamic Highlights Panel component.

Call Recording Player

Use this component to listen to audio recordings of calls. The Call Recording Player component is available only for the Voice Call object. This component is available in the Lightning App Builder in orgs where Service Cloud Voice is turned on. To view and use this component, users need the Contact Center Agent permission set.

Because the Call Recording Player doesn't have an alternative text-based transcript, we recommend adding the Conversation Body component underneath the Call Recording Player to support accessibility.

Chatter

As of API version 38.0, the Feed component has been renamed Chatter in the Lightning App Builder. Use it to place a publisher and feed combo on a record page.

Chatter Feed

Use the Chatter Feed component to place a feed anywhere on a record page. The feed gives you a way to view posts, comments, and questions. Its attribute, Feed Type, takes one of these values.

- Bookmarked—Shows a feed of all items the current user has bookmarked
- What I Follow—Shows a feed of all items the current user has followed
- To Me—Shows a feed of all items where the current user is mentioned

No coding is required to join a feed to a publisher. The connection is made automatically with the publisher and any feed on the page. The Chatter Feed component is available in API version 38.0 and later.

Chatter Publisher

Use the Chatter Publisher component to place a feed publisher anywhere on a record page. The publisher gives you a way to post, poll, or ask a question in a feed. Its attribute, Type, has the default value *Global*. Use the value *Record* when you want to associate the publisher with an object's feed. Then posts that are created with the publisher are associated with the record rather than the global Chatter feed. Use the Chatter Publisher component with the Chatter Feed component to get a full feed experience. No coding is required to join a publisher to a feed. The connection is made automatically with the publisher and any feed on the page. The Chatter Publisher component is available in API version 38.0 and later.

Conversation Body

Use this component to let agents view call transcripts in Service Cloud Voice. The Conversation Body component is available only for the Voice Call object. Transcripts are generated in real time during the call and the complete transcript is stored on the call record. As the call progresses, the transcript is

generated and displayed in the component. Configure transcription data streams in Amazon Web Services to generate call transcripts. To view and use this component, users need the Contact Center Agent permission set.

CRM Analytics Collection

The CRM Analytics Collection component enables you to insert and display collections of curated dashboards and lenses to your Lightning pages. The component intuitively displays collections in a concise panel so users can view relevant insights in the place where they work.

Available in API version 54.0 and later. Users must have access to the selected collection used within the component.

LWC CRM Analytics Dashboard

The LWC CRM Analytics Dashboard component surfaces an entire dashboard right where people work. The dashboard is interactive and works seamlessly in Lightning Experience pages and LWR sites. Users can refresh a dashboard, apply filters, and click chart segments to drill into filtered reports.

A dashboard needs space to display charts and tables. If an embedded dashboard is placed in too small a space, a collapsed version is displayed. Users can click **View Dashboard** to expand a collapsed dashboard.

Available in API version 60.0 and later. Private dashboards aren't available.

For more information, see [Embed CRM Analytics Dashboards in Lightning Pages](#).

CRM Analytics Dashboard

The CRM Analytics Dashboard Aura component surfaces an entire dashboard right where people work. The dashboard is fully interactive. Users can refresh a dashboard, apply filters, and click chart segments to drill into filtered reports.

A dashboard needs space to display charts and tables. If an embedded dashboard is placed into too small a space, then a collapsed version is displayed. Users can click **View Dashboard** to expand a collapsed dashboard.

Available in API version 41.0 and later. Private dashboards aren't available.

For more information, see [Embed CRM Analytics Dashboards in Lightning Pages](#).

Dynamic Actions Bar (Pilot)

With this component you can add action buttons anywhere on your Lightning record and app pages. Add, delete, and change the order of actions in the Dynamic Actions Bar. You can also control the visibility of the component and of individual actions in the component based on record field, permission,

or user.

 **Note** We provide the Dynamic Actions Bar component to selected customers through a pilot program that requires agreement to specific terms and conditions. To be nominated to participate in the program, contact Salesforce. Pilot programs are subject to change, and we can't guarantee acceptance. The Dynamic Actions Bar component isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. We can't guarantee general availability within any particular time frame or at all. Make your purchase decisions only based on generally available products and features.

To set the visibility of individual actions in the Dynamic Actions Bar component at runtime:

1. In the Dynamic Actions Bar properties pane, click an action name to open the Action dialog.
2. Under Set Action Visibility in the Action dialog, click **Add Filter**.
3. To control action visibility for a record field, select **Record Field**. To control action visibility for a record, permission, or user, select **Advanced**.
4. For Record Field, click the Field name field and then select a field from the dropdown list. For Advanced, select a field type and then select a field.
5. Select an operator and enter a value, then click **Done**.

Keep these considerations in mind when using the Dynamic Actions Bar component.

- This component is supported only for Lightning Experience desktop on record and app pages.
- The Dynamic Actions Bar supports all standard and custom global actions.
- This component is supported as a source for Dynamic Interactions on app pages only. See [Dynamic Interactions in the Lightning App Builder](#).
- Email quick action and `FeedItem.MobileSmartActions` actions aren't supported for the Dynamic Actions Bar.
- If no actions are supported for the current object, a message appears in the Lightning App Builder properties pane.

Dynamic Highlights Panel

See [Dynamic Highlights Panel](#).

Dynamic Related List–Single

Use this component to add or upgrade a single related list on your Lightning page and customize it directly in the Lightning App Builder instead of in the page layout editor. In the component properties, choose the list's fields and sort order, apply filters, add actions, and give the list a descriptive name. To see the most relevant records, set up two or more related lists with different filters on the same object.

The Dynamic Related List–Single component is available for custom objects and for Salesforce record home objects that are enabled for LWC. The Dynamic Related List–Single component isn't available for external objects. If you don't see Dynamic Related List–Single in the list of standard components, it's not

currently available for the object associated with the record page. You can use this component in record pages only.

Keep these considerations in mind when using the Dynamic Related List-Single component.

- Only the Related List-Single component can be upgraded to the Dynamic Related List-Single component. If you don't see the option to upgrade, Dynamic Related List-Single isn't supported for the object associated with the record page or for the related list selected in the properties pane.
- You can add standard actions, custom buttons, and Create a Record and Update a Record quick actions on dynamic related lists. Quick actions are available as mass actions on the related list.
- Custom actions aren't supported on dynamic related lists.
- In the Lightning App Builder, the dynamic related list preview shows a maximum of six records. Depending on permissions, the number of records that users see in the related list on the Lightning page doesn't always match the number of records in the list preview.
- The sort order that you define is the related list's initial sort behavior. If a user sorts the list on a different field or in a different order, their sort order overrides yours. The next time the user views the related list, they see the records in the order they selected.
- Only the List and Tile related list types are available for dynamic related lists. To display more than four fields in a dynamic related list, use the List type in a main region.
- To filter on a currency field in a dynamic related list, enter only numbers and an optional decimal point. For multiple currencies, the filter uses the default currency, even if you enter a currency symbol. For example, your default currency is in euros but you enter the filter *Price greater than or equal \$15000*. The related list shows records with prices greater than or equal to €15,000.
- Dynamic related lists support most relative date filters so you can filter on Date and DateTime fields using human-speech-inspired syntax. These relative date filters aren't supported: *n DAYS AGO*, *n WEEKS AGO*, *n MONTHS AGO*, *n QUARTERS AGO*, *n YEARS AGO*, *n FISCAL QUARTERS AGO*, *n FISCAL YEARS AGO*.
- Related list results are truncated at 2,000 records, so a filter set that returns over 2,000 records spends extra time querying records it can't display. To improve your list's performance, filter it to return a maximum of 2,000 records.
- When users go to a dynamic related list from a bookmark or shared URL, if they haven't viewed the list before, they're redirected to the record detail page. To see the entire list, they can click **View All** again.
- In Salesforce Console apps, a new subtab opens every time a user clicks **View All** on a dynamic related list. If a user clicks **View All** more than once, multiple subtabs open.
- The URL associated with the View All page is dynamically generated and is supported only on the record home page. Don't embed the View All URL in a custom component.
- Related list avatars aren't displayed on cards in the Dynamic Related List-Single component when viewed on mobile devices.

Einstein Field Recommendations

The Einstein Field Recommendations component recommends values for picklist, checkbox, and lookup fields on cases. It's used in Einstein Case Classification and Einstein Case Wrap-Up and is available only if you enabled one or both of those features. Einstein's field value recommendations are based on data from recently closed cases. The component has two types: Case Classification and Case Wrap-Up.

To learn more about using this component with Einstein Case Classification, see [Display Recommendations in the Service Console](#).

Available in API version 49.0 and later.

Einstein GPT Sales Emails

If Sales Emails and Gmail or Outlook integrations are enabled, the Einstein GPT Sales Emails component appears on the default email application pane. The component can be added to custom email application pane layouts. To learn more about using Sales Emails in Outlook and Gmail, see [Understand How Einstein Generative AI Creates Sales Emails](#).

Einstein Next Best Action

The Einstein Next Best Action component displays suggested recommendations and actions on a record page. Use strategies to apply your org's business rules to display context-specific suggestions to users.

See [Einstein Next Best Action Component](#).

Einstein Predictions

The Einstein Predictions component displays predictions and recommendations on a Lightning record page for a standard or custom object. See [Add Einstein Predictions to a Lightning Page](#).

Einstein Replies

The Einstein Replies component recommends stock replies that support agents can insert into chat and messaging sessions. It's used in Einstein Reply Recommendations and is available only if that feature is enabled. The replies that Einstein recommends are based on past closed chats. See [Einstein Reply Recommendations](#).

If Einstein Reply Recommendations is enabled, this component appears automatically on the Chat and Messaging console tabs for any users with the View and Act on Einstein Reply Recommendations user permission.

Available in API version 49.0 and later.

Flow Orchestration Work Guide

The Flow Orchestration Work Guide component displays orchestration work items on a Lightning record page. Use this component to ensure that users can see work items assigned to them from orchestration runs. To learn more about Flow Orchestration see [Flow Orchestration](#).

Forecasts Header

The Forecasts Header component contains key page-level actions and selections for a forecasts page, such as selecting the forecast type to show and the forecast period. We recommend that each forecasts page include a Forecasts Header component so that viewers can select their preferences and a forecast type. Those selections allow other forecasts components on the page to work correctly. Selections made in this component on a forecasts page determine what data is shown in other forecast-related components on the page.

This component is supported for Forecasts pages.

Forecasts Opportunity List

The Forecasts Opportunity List works in conjunction with the Forecasts Header and Forecasts Summary components and shows the opportunities that contribute to a selected forecast line in the summary.

This component is supported for Forecasts pages.

Forecasts Summary

The Forecasts Summary is the main part of any forecast page and shows the actual forecast numbers based on the selections made in the Forecasts Header on the page. The summary is where sales leaders can view forecasting data, make forecast adjustments, and update quotas. Pieces contained in the summary are configurable in Forecasts Settings in Setup.

This component is supported for Forecasts pages and works best in a wide region of the page.

If Einstein Forecasting is enabled, the summary also includes AI-powered intelligence to improve forecasting accuracy and predictions.

Highlights Panel

The Highlights Panel component displays key record fields along with page-level actions. The fields that appear in the highlights panel come from the compact layout assigned to the object. The actions in the highlights panel come from the Salesforce Mobile and Lightning Experience Actions section of the page layout. Record highlights show up to the first 7 fields on desktop and up to the first 10 fields in the Salesforce mobile app.

To streamline your highlights panel by not showing the second row of information and reducing the size of the first row, select the **Show as collapsed (desktop only)** checkbox. To show fewer action buttons, reduce the number using the **Number of Visible Action Buttons (desktop only)** attribute.

To display the highlights horizontally or vertically (desktop only), drag the Highlights Panel component into a region with the horizontal or vertical dimensions you want. The highlights panel adjusts to fit the region's space. For example, if you drag it into a narrow column, the highlights display vertically. If you

drag it to a full-page width column, the highlights display horizontally.

When a field with an image larger than 50x50 pixels is placed as the first field in the Highlights Panel, the image is cropped. In any other position, the image appears as expected. If the image is in a formula field, you can get around this issue by setting the height and width values of the image in the formula to 50 pixels each. For more information, see [IMAGE](#) in Salesforce Help.

If you configure a highlights panel with a formula field that contains an image, the first time the image renders on the page, its width is set to a smaller value. After navigating away and coming back to the page, updating the record, or performing an action on the page, the highlights panel renders the field's image the correct size. To avoid this behavior, set the size of the image in the formula field.

List View

The List View component points to a list view and displays the first few records from that view. It supports all public and shared list views that are associated with standard and custom objects, except:

- Activity
- ContentVersion (Files)
- Recently Viewed
- User
- UserProfile

In addition to the Recently Viewed list view, some standard objects include another list view with a similar name that shows the same records. The name of this view includes the name of the object, for example, Recently Viewed Accounts. The List View component supports these list views.

Keep these considerations in mind when using the List View component.

- By default, a List View component displays the first three records in the list, but you can set it to show a maximum of 30.
- You can't give a List View component a custom name. The component's name is derived from the name of the list view filter you select when you configure the component.
- In the Lightning App Builder, the Object dropdown list for this component displays only those objects that have list views associated with them.
- Adding too many List View components can cause page performance issues. Use them sparingly.

Order Product Summaries by Recipient

Use the Order Product Summaries by Recipient component to display order product details on an Order Summary record page. This component is available in Salesforce Order Management.

The Order Product Summaries by Recipient component displays information about the order delivery group summaries associated with the order summary, including the order product summaries associated with them.

The Order Product Summaries related list on the OrderDeliveryGroup object page layout defines the displayed order product summary fields. To modify the columns in this component on the order summary details page, edit the related list on the Order Delivery Group page layout, not on the Order Summary page layout.

You can create a custom filter to control which order delivery group summary records are displayed.

Order Summary Totals

Use the Order Summary Totals component to display order financial totals on an Order Summary record page. This component is available in Salesforce Order Management.

You can customize the panel title and which values to display.

Phone

Use this component to give agents easy access to the Service Cloud Voice softphone call controls so they can mute, hold, and end calls. This component is displayed when an agent accepts a call and is hidden when the call ends. The Phone component is available for the Account, Case, Contact, and Voice Call objects and for custom objects. To view and use this component, users need the Contact Center Agent permission set.

Quip Document

Use the Quip Document component to embed Quip documents directly in records. Set up any Quip document as a template so that your users can quickly create documents on Salesforce records.

When you set up the Quip Document component in Lightning App Builder, you can choose different modes.

- Allow different documents on each record. Let users attach different documents to different records, or create and embed new documents from scratch. Admins can programmatically attach preselected documents to different records.
- Use the same document for every record. Choose an existing document to embed in each record associated with a given object.
- Use a template to create documents for each record. Specify a template where users can create and embed documents on a per-record basis. You can use any Quip document as a template. You can set up the template to add record data to documents.
- Use different templates for different records. Specify different templates for different records on the same object. This option requires you to programmatically pre-populate the component field with the URLs of different templates rather than manually choosing a single template URL.

Rebate Types Panel

Use the Rebate Types Panel component to select and apply eligible rebate types and incentives to a

mapped object. This component is supported in API version 52.0 and later.

 **Note** To view and use this component, users need the Rebate Management license.

Rebate Types Tab

The Rebate Types component is used in combination with the Rebate Types Panel component to view and modify the benefit tiers associated with the applied rebate types and incentives. This component is supported in API version 52.0 and later.

 **Note** To view and use this component, users need the Rebate Management license.

Recent Items

The Recent Items component displays a list of the most recently used items. The default number of displayed items is three, but you can set it to show a maximum of 30. In the Lightning App Builder, you can specify which objects' records appear in the recent items list.

The Recent Items component supports these objects, based on the specified properties:

- All custom objects.
- All standard objects for which both of these conditions are true:
 - A compact layout is defined for the object.
 - The object is tracked in the most recently used objects list.

 **Note** The object isn't supported for this component if it meets both of these criteria but isn't in the list of available objects when you configure the component. Although they appear in the available objects list, the Task, Report, KnowledgeArticle, and Article objects aren't supported for this component.

If an object is tracked in the most recently used objects list, one or both of the **LastViewedDate** or **LastReferencedDate** fields are present.

Record Detail

The Record Detail component displays fields and sections from the page layout associated with the object. When users view the Lightning record page, they see different fields and sections based on their profile and page layout assignments.

You can't add, remove, or move the fields and sections when you're viewing the component in the Lightning App Builder. You can only make field and section changes on the page layout.

Related List-Single

Use the Related List-Single component to include a *single* related list for a record in your Lightning page.

You can show a related list for the record associated with your page, or you can show information for the parent record. Make sure that the page layout for your users includes the related list that you want to use. If you're using a parent record, update the parent record's page layout. Using a parent record is optional.

To customize how the list appears, update the list type. The Basic List type displays only the first four fields of a related list. With the Enhanced List type, you can show up to 10 fields and show or hide actions, and users can resize and sort columns, perform mass actions, and wrap text.

This component is supported in API version 39.0 and later. You can use it in record pages only.

Keep these considerations in mind when using the Related List–Single component.

- The action bar always appears on enhanced related lists that are in a narrow sidebar region on the record page.
- In the Related List field in the properties pane, you can see all of the related lists that are available for all of an object's record types. For example, in orgs with Person Accounts enabled, you can see all of the related lists for both the Account and Person Account record types. If the same related list exists on more than one record type, you see duplicates when you select a list.
- Duplicated Related List–Single components of the same relationship share preferences, like column widths, text wrapping and clipping, and sort order. For example, if you add two Related List–Single components and select Cases as the Related List on each component, then changes to the preferences for one of the components affect both components. A Related List–Single component and a Dynamic Related List–Single component of the same relationship don't share preferences.

Related List Quick Links

The Related List Quick Links component displays a set of links. Users can hover over the links to see all the related list columns without opening the View All page. Header actions, mass actions, row actions, and text wrapping are all available on the hover pane.

The component displays two rows of related list links in large or medium page regions, and six rows in small regions. Users can view the remaining related list links by clicking **Show All**, which expands the component. When a user hovers over a related list quick link, it displays the first 10 items in the related list.

The content of this component is based on the set of related lists on the object's page layout plus the user's preferences. Users can customize the order of the quick links. They can also exclude the ones they don't want in their personal settings. To exclude, in the Quick Find box, enter *Customize My Pages*, then select **Customize My Pages**, and then click the object.

The Open Activities and Activity History related lists aren't supported for this component. You can use this component in record pages only.

Related Lists

Use the Related Lists component to include *all* related lists for a record in your Lightning page. Make sure

that the page layout for your users includes the related lists that you want to use.

To customize how the lists appear, update the list type. The Basic List type displays only the first four fields of a related list. With the Enhanced List type, you can show up to 10 fields and show or hide actions, and users can resize and sort columns, perform mass actions, and wrap text. The action bar always appears on enhanced related lists that are in a narrow sidebar region on the record page.

You can use this component in record pages only.

Related Record

Use the Related Record component to display the details of a related record, including the details of a parent record, in your Lightning page. This component provides your users with built-in record creation, inline edit, and the ability to unlink a record and link a new one. This functionality is possible because the component uses actions.

This component is supported in API version 39.0 and later. You can use it only in record pages.

Keep these considerations in mind when using the Related Record component.

- To use the component, an object must have an associated quick action to update the records. Some lookup fields have default actions. If no actions are available for your lookup, follow the links in the Lightning App Builder property editor to create the actions.
- To change the displayed fields for the Related Record component, configure different lookup fields, and customize the associated action in Setup. If you don't see the action or can't modify it, create one. Also, make sure that the lookup field to the related object is included on the page layout of the main object. Otherwise the component can't be refreshed.
- To let users look up two levels of record relationships, specify the first-level lookup and then the second-level lookup. You must specify a first-level lookup before you can add a second-level lookup.
- To let users look up polymorphic fields, select a polymorphic lookup field type on the first-level lookup or on the second-level lookup.
- To use the Parent Case, Asset, and Case Source lookup fields on cases, change the field-level security to visible instead of hidden. Otherwise, your users see an error.
- Users without Read access to the value of a lookup field see an error.
- Person account records that display in contact Related Record components are read only.
- Cases are linked to default accounts that can't be removed (unlinked) from the component unless the contact is also removed at the same time.
- The Related Record component typically uses quick action metadata to determine which fields to show. Before Spring '24, read-only users saw a compact layout of fields, but now they see the same layout as other users when they view the component.

Report Chart

Use the Report Chart component to include a chart from a report in your Lightning page. If you leave the component's Label field blank, the component's label is derived from the report's label.

The chart refreshes if its report data is more than 1 day old. In the component properties, you can choose to display a refresh button to enable users to refresh the chart. Saving the report's definitions also updates the chart data in the component.

Setting a filter on the report chart data is supported only for record pages. If you set a filter option, the Report Chart component displays only that filtered data to users.

This component is supported in API version 32.0 and later. It doesn't work with reports in the My Personal Custom Reports folder. Report Chart components that refer to reports in the Unfiled Public Reports folder aren't deployable when you include them in a package.

Rich Text

Use the Rich Text component to add text and simple HTML markup to your Lightning page.

 **Note** JavaScript, CSS, iframes, and other advanced markup aren't supported. To use advanced HTML elements in a component, we recommend using a Visualforce page component or a custom Lightning component.

The Rich Text component uses Quill as its text editor. Keep these considerations in mind when using the text editor.

- Quill wraps each line of text with `<p> </p>` tags. The tags can increase the number of characters in the rich text API value when you save the text. If you get a maximum-length warning, break up the text into two Rich Text components.
- In Quill, the default color for text is gray in the editor, but the text renders black in the output.
- Selecting text using keyboard shortcuts, such as with Cmd+A, and then typing something new resets the formatting of the existing text.

You can include up to 4,000 characters in the Rich Text component. This component is supported in API version 32.0 and later.

Salesforce Surveys

The Salesforce Surveys component adds an active survey to your Lightning page, so you can collect data from your users while they work on Salesforce records. This component is supported in API version 42.0 and later.

 **Note** To create surveys and add them to Lightning pages, you must enable Salesforce Surveys in your org.

Send Email Later – Pending List

The Send Email Later – Pending List component shows the list of scheduled emails. From this list, your

users can see pending emails, cancel an email, edit its scheduled time, or edit its content.

Tabs

Use the Tabs component to add tabs to a region of your Lightning page. Choose from a set of standard tabs or create custom tabs to enhance record and Home pages for your Lightning Experience users. The Tabs component is supported only for Lightning Experience record and Home pages.

You can place up to 100 tabs in a Tabs component. This component is supported in API version 36.0 and later.

Twitter

Social Accounts and Contacts must be enabled for your organization before you can add the Twitter component to a Lightning page.

Visualforce Page

Use the Visualforce Page component to include a Visualforce page in your Lightning page.

If you leave the component's Label field blank, the label is taken from the Visualforce page that you assign to it.

If you leave the Height field blank, the Visualforce page's height defaults to 300 pixels when it is displayed in the Salesforce mobile app.

This component is supported in API version 32.0 and later.

To appear in the Salesforce mobile app or Lightning Experience, the Visualforce page must have the Available for Salesforce mobile apps and Lightning pages option selected. This option is available for pages that are set to API version 27.0 and later.

Voice Status

Use the Voice Status component to include tests in your Lightning page that help service agents check whether their workspaces are ready for Service Cloud Voice. When agents click **Run Test** from the Voice Status component, their browser and microphone statuses are verified. Additionally, agents using Service Cloud Voice with Amazon Connect run the Amazon Connect Endpoint Test Utility to verify their connection. If verification fails, messages instruct agents how to debug their Voice setup to connect successfully. This component is supported in API version 57.0 and later.

See Also

[Custom Lightning Page Components](#)

Custom Lightning Page Components

The Lightning App Builder supports custom Lightning components that you create either with Aura or Lightning web components (LWC).

REQUIRED EDITIONS

Lightning App Builder available in: both Salesforce Classic and Lightning Experience

Lightning Home and utility bar pages available in: Lightning Experience

Lightning app and record pages available in: both the Salesforce mobile app and Lightning Experience

Email application pane pages available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Custom components in your org that are configured for use in the Lightning App Builder appear in the Lightning Components pane.



If you have a page to be used in the Salesforce mobile app, use only Lightning components that the mobile app supports.

Your custom Lightning components don't automatically work on Lightning pages or in the Lightning App Builder. To make a custom component usable in both, configure the component and its component bundle so that they're compatible with the Lightning App Builder and Lightning pages.

For Aura components, see the [Lightning Aura Components Developer Guide](#). For Lightning web components (LWC), see the [Lightning Web Components Developer Guide](#).

-  **Note** Custom components that serve as containers, such as custom Tabs or Accordion components, aren't supported in Lightning App Builder. They display on the canvas, but you can't interact with them or put any components inside them.

You can configure custom components to support different devices. For instance, you have a record page whose template supports both phone and desktop, and you activate the page for both. Then you add a phone-only component to the page. When the page is viewed on a phone, users see the component. When the page is viewed on a desktop, the phone-only component doesn't appear.

-  **Note** Pull to refresh doesn't work for custom Lightning web components in the Salesforce mobile app.

See Also

[Standard Lightning Page Components](#)

Visibility Rules on Lightning Pages

Control when a component appears on a Lightning page by adding filter conditions and logic to its properties in the Lightning App Builder. For example, you can construct a filter that causes a rich text component on an opportunity page to display when the opportunity's amount is greater than or equal to US\$1 million.

REQUIRED EDITIONS

Lightning App Builder available in: Salesforce Classic and Lightning Experience

Lightning pages available in: Lightning Experience and the Salesforce mobile app

Available in: **Group Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Visibility properties appear when you select a component on a record, app, or Home page in the Lightning App Builder or on a field on a Dynamic Forms-enabled record page. You can also set visibility conditions to dynamically show or hide an individual tab in the Tabs component. This behavior applies to standard components, custom components, and components from AppExchange. No need to do anything to your custom components. The Lightning App Builder handles all of it.

On record pages, you can filter on record fields or advanced fields, such as fields from related objects or from a global object like User. Field values in visibility filters can't span more than five fields. For example, `Record.Account.Owner.Manager.Manager.Manager.LastName` has six spans, so it isn't supported.



App and Home pages aren't associated with an object, so the filters you can use are limited to other contexts, such as User, User Permission, or Device. But that doesn't mean that they're less powerful.

If you don't define a filter, the component displays on the Lightning page as usual. When you define one or more filters and set the filter logic for a component, it's hidden until the filter logic criteria are met.

In the Lightning App Builder, items that have at least one filter assigned are indicated with an icon (👁️).

Visibility Rules on Dynamic Forms Fields and Field Sections

You can make your Lightning record pages even more dynamic by setting visibility filters on Field and Field Section components. For example, you can have a field or set of fields hidden until a person with a certain profile, permission, or viewing on a certain device visits the page.

Be careful when setting up visibility rules on multiple components in the same region. If your rules cause all the components in a region to be invisible at run time, the region is empty.

If a field is set to Required in the Lightning App Builder, it's hidden by a visibility rule at run time, and users can save the record even if that field isn't populated.

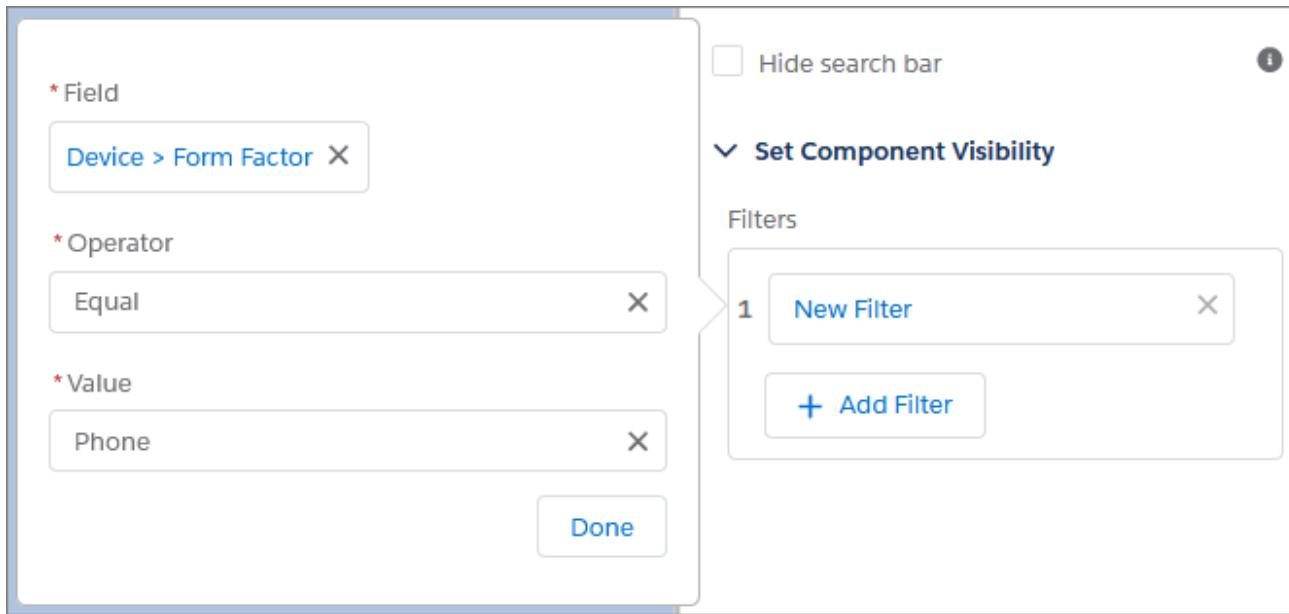
Visibility rules on fields are respected in the edit, clone, inline edit, and new record screens. Component visibility rules on field sections behave differently than they do on fields. Visibility rules on fields are assessed dynamically. Changes a user makes while editing a record can make fields appear and disappear as visibility rules are evaluated. Visibility rules on field sections aren't dynamic and don't react to what a user does while editing. Field section visibility rules are evaluated only after the record is saved.



Note Field and Field Section components are supported on desktop only. Therefore, the visibility rules you set on them are respected only in the desktop view, not on mobile. However, Dynamic Forms on Mobile gives your mobile users the same experience that your desktop users have. To avoid this issue, from Salesforce Mobile App Setup, enable **Dynamic Forms on Mobile**.

Component Visibility Based on Form Factor

With a filter using the Device context, you can set a component to display exclusively when its page is viewed in a specific experience, such as a phone or a desktop.



Custom Lightning components can also be set to support different form factors. For Lightning web components, see [Configure Your Component for Different Form Factors](#). For Aura components, see [Aura Component Bundle Design Resources](#).

Supported Objects, Fields, Field Types, and Operators

Two objects aren't supported for visibility filters: `ProcessInstanceStep` and `ProcessInstanceWorkItem`.

On record pages, visibility filters rely on the data captured in fields associated with the page's object. Not all fields, field types, and operators are supported.

These field types are supported:

- String type fields: Autonumber, Currency, Email, Number, Percent, Phone, Text, Text Area, URL
- ID
- Checkbox (boolean)
- Geolocation
- Picklist
- Formula fields that resolve to one of these preceding types
- Roll-up summary fields that resolve to one of these preceding types
- Lookup fields are supported, however parent lookup fields on Dynamic Forms-enabled pages aren't supported

These operators are supported.

- CONTAINS
- = and == (Equal)
- <> or != (Not Equal)
- > (Greater Than)
- >= (Greater Than or Equal)

- < (Less Than)
- <= (Less Than or Equal)

See Also

[Formula Operators and Functions by Context](#)

[Standard Lightning Page Components](#)

[Dynamic Forms and Mobile Using the Record Detail - Mobile Component](#)

Lightning Page Visibility Rule Considerations and Limitations

Keep these considerations and limitations in mind when working with visibility rules in the Lightning App Builder.

REQUIRED EDITIONS

Lightning App Builder available in: Salesforce Classic and Lightning Experience

Lightning pages available in: Lightning Experience and the Salesforce mobile app

Available in: **Group Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

General

- Visibility filters support a single value at a time, not comma-separated values. To add more than one value, add a new filter.
- Lookup fields that you want to use as a filter for a visibility rule must always have a value.
- If a user doesn't have access to a field used as part of a component visibility filter's criteria, the criteria evaluate to false.
- For visibility rules on dynamic actions in the Highlights Panel or Dynamic Highlights Panel, if a user doesn't have access to a field used as part of the action's visibility filter criteria, the criteria don't apply, and therefore evaluate to true.
- If all of the components inside an individual tab on a Lightning page are hidden due to visibility rules, then Salesforce hides that tab. For example, if all components inside the Related tab are hidden, then Salesforce hides the Related tab. Also, if all of the individual tabs inside a Tabs component are hidden due to visibility rules, then Salesforce hides the entire Tabs component.
- If the default tab becomes hidden after being selected because its components were hidden due to visibility rules, then the default tab isn't reselected if one of its components becomes visible again later.
- If you set visibility criteria based on a lookup field and that lookup field resolves to a null value at run time, then the page won't load properly.
- When creating a visibility rule, sometimes an error can occur when you enter text in the Field field and then backspace. If this happens, close the visibility rule window and try again.
- Visibility rules that reference a picklist field can sometimes not work as expected for users without

access to that field. This can occur when the picklist field is from the same object or from a different object via lookups. The field or component governed by the visibility rule can be incorrectly shown or hidden even when the rule criteria are met.

Limitations

- A component can have up to 10 filters.
- Some person account fields, including Email and Mobile, can't be used in filters.

Visibility Rules and Form Factors

The Salesforce mobile app on a tablet uses the phone form factor. Salesforce viewed in a browser on a tablet uses the desktop form factor. These form factors affect component visibility on record pages differently.

If the visibility filter is “Device not equal to desktop” or “Device equals phone,” the component appears on pages in the Salesforce mobile app on a phone and a tablet. It doesn’t appear on pages viewed in a browser on a tablet or on desktop.

If the visibility filter is “Device not equal to phone” or “Device equals desktop”:

- The component appears on pages in Lightning Experience on a desktop and on pages viewed in a browser on a tablet.
- In orgs without Dynamic Forms and Dynamic Highlights on Mobile enabled, the component also appears on pages viewed in the Salesforce mobile app on a tablet. It doesn’t appear in the Salesforce mobile app on a phone.
- In orgs with Dynamic Forms and Dynamic Highlights on Mobile enabled, the component doesn’t appear in the Salesforce mobile app on a phone or a tablet.

Visibility Rules and Dynamic Forms

- Field and Field Section components are supported on desktop only. Therefore, the visibility rules you set on them are respected only in the desktop view, not on mobile. However, Dynamic Forms on Mobile gives your mobile users the same experience that your desktop users have. To avoid this issue, from Salesforce Mobile App Setup, enable **Dynamic Forms and Dynamic Highlights on Mobile**.
- If a field is hidden on a page due to a visibility rule, and a user edits the value of the field, its new or changed value isn’t saved.
- Dependent picklist fields don’t respect visibility rules. Even if they have visibility rules assigned to them, dependent picklist fields still appear in the View All Dependencies list.
- Component visibility rules on field sections behave differently than they do on fields. Visibility rules on fields are assessed dynamically. Changes a user makes while editing a record can make fields appear and disappear as visibility rules are evaluated. Visibility rules on field sections aren’t dynamic and don’t react to what a user does while editing. Field section visibility rules are evaluated only after the record is saved.
- If you hide a field with a visibility rule, users don’t lose access to the field’s data. Values in hidden fields

- remain intact and are visible outside of the Dynamic Forms-based page in places like reports, dashboards, and list views. Hidden fields are for user convenience, not for field-level security.
- When a tab is hidden by a visibility rule, any field sections within that tab are hidden in the Create and Edit windows. If all field sections on a page are contained in the hidden tab, then the Create and Edit windows are blank.

Build Localized Component Labels and Attribute Values on Lightning Pages with Custom Labels

When you specify labels or attributes in the Lightning App Builder such as for components, field section headers, or custom tabs, you can use the `{ !$Label.customLabelName }` expression to help define the label or attribute's value.

REQUIRED EDITIONS

Lightning App Builder available in: both Salesforce Classic and Lightning Experience

Lightning Home pages available in: Lightning Experience

Lightning app and record pages available in: both the Salesforce mobile app and Lightning Experience

Available in: **Group Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create and save Lightning pages in the Lightning App Builder: Customize Application

On a Lightning page, custom Tabs component labels, field section header labels, and other component attribute values aren't localized when they're entered as plain text in the Lightning App Builder. For example, if you have an org whose default language is English, and you have a Tabs component with three custom tabs that you entered as Cars, Trucks, and Planes, the non-English users in your org see those tab label values as Cars, Trucks, and Planes when they view the page. The tab label values aren't translated into your users' languages.

But configuring custom label values in the Lightning App Builder using the `{ !$Label.customLabelName }` expression lets users see labels in their chosen language if you created a translation for that label in their language.

The `{ !$Label.customLabelName }` expression works with any custom labels that you create in Setup using the custom label feature. The text that you define in the Value field for your custom label displays as the label value when the component renders on a Lightning page. And, if you create a translated value for the label, users using that language in your org see the translated value.

Expressions are supported only in String type fields in component labels and attributes on app, Home,

and record pages. You can't use an expression in page-level properties.

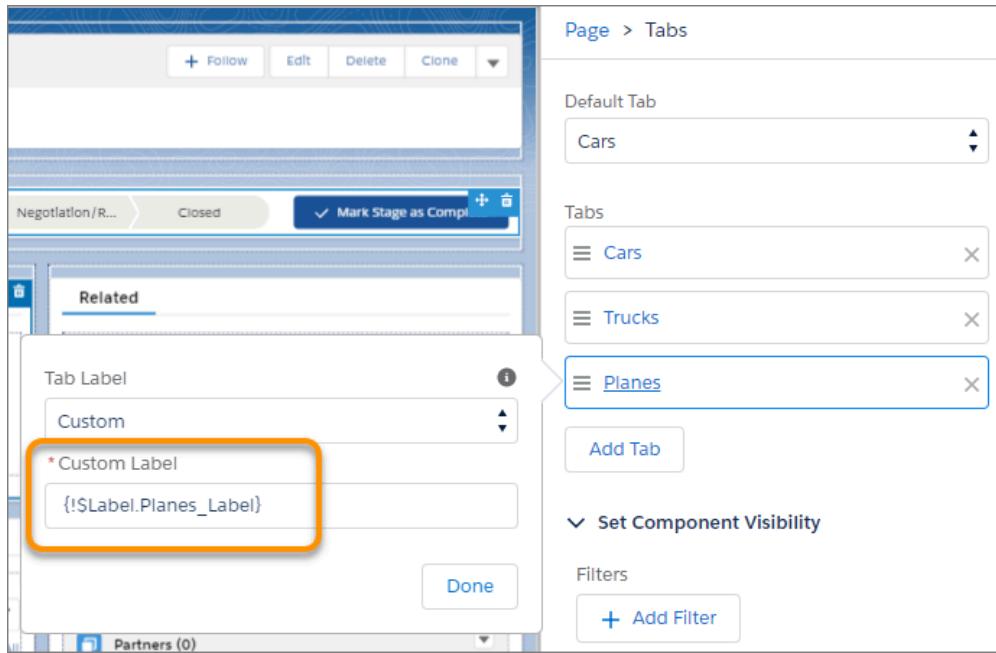
1. Create a translated custom label in Setup.
For more information, see [Translate Custom Labels](#).
2. Navigate to the Lightning App Builder. From Setup, in the Quick Find box, enter *App Builder*, then select **Lightning App Builder**.
3. Open an app, record, or Home Lightning page.
4. Select the component whose label or property you want to replace with your translated custom label.
5. Enter `{ !$Label.customLabelName }` in the label or attribute field, replacing “*customLabelName*” with the name of your custom label.
Label and property fields that support using this expression have an info bubble that says so.
6. Save your changes.
Users in your org whose language is set to the language of your translated label see the label or attribute in its translated value.

 **Note** When a field in a component contains an expression, Salesforce can't validate the value it resolves to. If the expression resolves to a value that's invalid for the field, sometimes the component doesn't work as expected. We recommend that you test the page in the translated languages before you make the page with the expression available to users.

Example: Create German Translation for Planes Tab

Your org's default language is English, but you want the custom Planes tab on your Lightning page to show a translated custom label for your German users.

1. Create a custom label in Setup with the name *Planes_Label*, and enter the Value as *Planes*. That's what your English users see.
2. From the Translations related list on the custom label detail page, create a translation for the label with the language set to German and the Translation Text value as *Flugzeuge*.
3. Save the translation.
4. In the Lightning App Builder, open your page and click the Tabs component on the canvas.
5. In the properties pane, click the Planes tab, and in the Custom Label field, replace the Planes plain text with `{ !$Label.Planes_Label }`.



6. Click **Done** and save your page.

When viewing the page, your English users see the tab as Planes, and your German users see the tab as Flugzeuge.

Create and Configure Lightning Experience Record Pages

Use the Lightning App Builder to add, remove, or reorder components on a record page to give users a customized view for each object's records.

REQUIRED EDITIONS

Lightning App Builder available in: both Salesforce Classic and Lightning Experience

Lightning Home and utility bar pages available in: Lightning Experience

Lightning app and record pages available in: both the Salesforce mobile app and Lightning Experience

Email application pane pages available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create and save Lightning pages in the Lightning App Builder:

Customize Application

1. Create a record page for Lightning Experience in one of these ways.
 - From the Setup menu on a record page, select **Edit Page**.



When you select **Edit Page** for the first time, Salesforce makes a copy of the standard page. This copy is what you edit in the Lightning App Builder. If a customized page exists and is active, selecting Edit Page opens that page to edit.

- Create a page from the Lightning App Builder list page in Setup. Enter *App Builder* in the Quick Find box, then select **Lightning App Builder**, click **New**, and step through the page creation wizard. To create an empty page, select a page template. To create a page prepopulated with standard components, clone the system default page.
- Clone an existing custom Lightning page from its detail page or from the Lightning page list in Setup.
- Click **New Page** from the Pages list inside the Lightning App Builder.

2. In the Lightning App Builder, add, edit, or remove components to change the page's layout.

To add a component, drag it from the list of components onto the canvas. To remove a component from the canvas, select it and then click or press Delete on your keyboard. Reorder components by dragging them around the canvas.

3. In the page properties, give your customized page a unique, descriptive label.

To get to the page properties, click **Page** from the breadcrumb at the top of the properties pane.

4. Save your page.

Hang on, you're not done yet! To make your customized record page available to your Lightning Experience and mobile users, you must activate it. You can activate the page from the Save dialog when you save it for the first time or later using the Activation button.

See Also

[Add and Customize Tabs on Lightning Pages Using the Lightning App Builder](#)

[Lightning App Builder Considerations](#)

[Lightning App Builder Limits and Limitations](#)

Activate Lightning Experience Record Pages

Is your custom record page ready for your users? Use the Activation function inside the Lightning App Builder to get the page out to them. You can make your custom record page the default record page for all users, assign it to specific Lightning apps, or assign it to record types and profiles. Assigning the page to a Lightning app, record type, or profile gives your users access to a record page customized for the context they're working in.

REQUIRED EDITIONS

Lightning App Builder available in: both Salesforce Classic and Lightning Experience

Lightning Home and utility bar pages available in: Lightning Experience

Lightning app and record pages available in: both the Salesforce mobile app and Lightning Experience

Email application pane pages available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create and save Lightning pages in the Lightning App Builder: Customize Application

1. Create a record page or open an existing one in the Lightning App Builder.
2. If you opened a record page that is ready for your users and doesn't need editing, click **Activation**.
3. If you created a page or opened a page that needs adjustment, make the necessary changes, click **Save**, and then click **Activate**.
4. Choose how you want to activate the page.
 - Make the page the org default for the object.
After you activate the Lightning page as the org default, the page is selected as the Lightning Experience override for the View action in the Override Properties panel.
 - Make the page the default object record page for specific Lightning apps.
If you activate the Lightning record page for specific Lightning apps only, the page takes precedence over the Lightning Experience Override setting for the View action on the object in those apps.
 - Assign the page to a combination of Lightning apps, record types, and profiles.
 - Assign the page to a form factor, such as a desktop or phone.
5. On the activation screen, click the tab for the option you've chosen, and follow the steps to activate the page.

See Also

[Create and Configure Lightning Experience Record Pages](#)

[Lightning App Builder Considerations](#)

Create a Custom App Page: The Big Picture

With just a few steps, you can create an app page that lets your Lightning Experience and Salesforce mobile app users access the most important objects and items in your custom app. Custom app pages are built using Lightning pages.

1. Before creating your page, determine which components you want to include and the global actions your users need.
2. Create the Lightning page.
3. Add actions to your page.
4. Activate your Lightning page in the Lightning App Builder.

Activation lets you create a custom tab for your app page, set its visibility, and add it to the Salesforce "Mobile Only" app navigation and Lightning app navigation bar all in one place.

Create an App Home Page with the Lightning App Builder

Create a home page for an app that you can add directly into Lightning apps and the Salesforce mobile app. Your users can then easily access the objects and items that are most important in that app.

REQUIRED EDITIONS

Lightning App Builder available in: both Salesforce Classic and Lightning Experience

Lightning app pages available in: both the Salesforce mobile app and Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create and save Lightning pages in the Lightning App Builder: Customize Application

1. From Setup, enter *App Builder* in the Quick Find box, then select **Lightning App Builder**.
2. Click **New**.
3. Select **App Page** and then click **Next**.
4. Give your app page a label, and then click **Next**.
The label can be up to 80 characters.
5. Select a page template and click **Done**.
6. Drag the components that you want onto the page.
Drag components up and down to rearrange them. You can also click a component's top or bottom border to create an insertion point () for the next component.
7. Click each component on the page to configure its properties.
8. Click in the empty area of the canvas or on the **Page** link in the breadcrumb to configure the page properties.
The Description field is limited to 255 characters. The API Name field is limited to 80 characters, but if you have a namespaced org, we recommend using fewer than 65 characters. When you create a Lightning page, the API name is derived from the first 40 characters of the label that you give the page.
9. Optionally add global actions to your page.
 - a. In the page properties, click **Select**.
 - b. Add, remove, or reorder the actions for your page.
Actions you select appear on the page's action bar and in the highlights panel at the top of the page in Lightning Experience. An app page is the only type of Lightning page that you can add actions to in this way. Other Lightning pages derive their actions from the object and global page layouts.
 - c. Click **OK**.
10. Click **Save** when you're done editing your page.

To give your users access to the app page, you must [activate it](#).

See Also

- [Activate Your Lightning App Page](#)
- [Get Help for Lightning App Builder](#)

Create a Mobile App Page with the Lightning App Builder

Create a mobile home page for the Salesforce mobile app. Your users can see this page right when they log in.

REQUIRED EDITIONS

Lightning App Builder available in: both Salesforce Classic and Lightning Experience

Lightning app pages available in: both the Salesforce mobile app and Lightning Experience

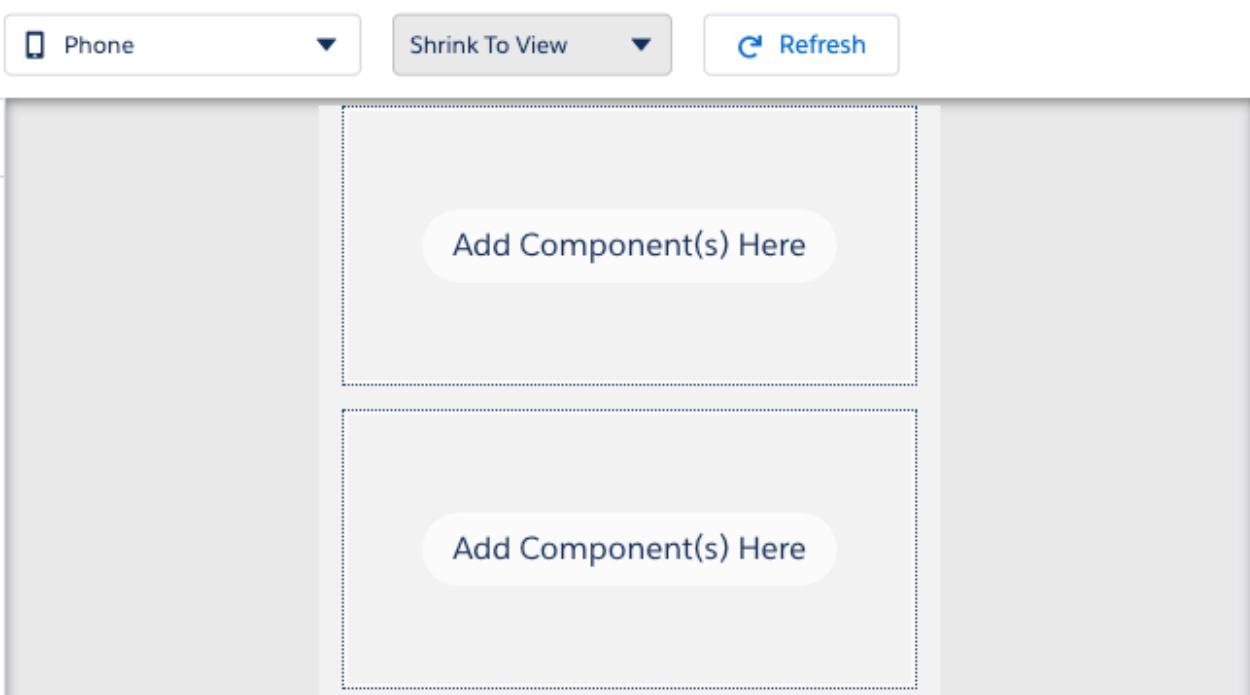
Available in: **Group, Essentials Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create and save Lightning pages in the Lightning App Builder: Customize Application

 **Note** These steps are similar to [Create an App Home Page with the Lightning App Builder](#). That's because when you create an app page or record page, it can be made available on both Lightning Experience and the Salesforce mobile app. The following steps focus on creating an app page for the mobile app only.

1. From Setup, enter *App Builder* in the Quick Find box, then select **Lightning App Builder**.
2. Click **New**.
3. Select **App Page** and then click **Next**.
4. Give your app page a label, and then click **Next**.
The label can be up to 80 characters.
5. Select the first template (**Header and Left Sidebar**) and click **Done**.
If this is a mobile-only page, it doesn't matter which template you pick. If you plan to share this page between mobile and desktop users, choose the template most appropriate for desktop use. On mobile devices, the page responsively collapses into a single column.
6. Make sure you're viewing the template preview using the Phone form factor. The regions on the template are displayed in a one column layout.



7. Drag the components that you want onto the page.

Drag components up and down to rearrange them. You can also click a component's top or bottom border to create an insertion point () for the next component.

To increase mobile adoption, we recommend making available the top features that your users use most frequently. For example, the Recent Items can take them back into the records that they were working with. Also, the List View standard components can be set to their open tasks to improve productivity in the mobile app.

Remember that screen space and bandwidth are precious on mobile, so try to limit each page to 4 or 5 components at most. If you overload the page with features, it can be slow and harder to use.

See [Preview Mobile App Pages in Lightning App Builder](#).

8. Click each component on the page to configure its properties.

9. Click in the empty area of the canvas or on the **Page** link in the breadcrumb to configure the page properties.

The Description field is limited to 255 characters. The API Name field is limited to 80 characters, but if you have a namespaced org, we recommend using fewer than 65 characters. When you create a Lightning page, the API name is derived from the first 40 characters of the label that you give the page.

10. Optionally add global actions to your page.

Global actions help your users perform tasks easily from the app page, like log a call, create a new case or event etc.

a. In the page properties, click **Select**.

b. Add, remove, or reorder the actions for your page.

Actions you select appear on the page's action bar and in the highlights panel at the top of the page in Lightning Experience. An app page is the only type of Lightning page that you can add actions to in this way. Other Lightning pages derive their actions from the object and global page layouts.

c. Click **OK**.

11. Click **Save** when you're done editing your page.

To give your users access to the mobile app page, you must [activate it](#). You can limit the page to mobile users during the page activation process. After activation, you can add this to any Lightning App in the App Manager for your users to personalize. If your Lightning App is available on both desktop and mobile, you can use the same page across both. Or create a Lightning App specifically for your mobile users, and add it to the navigation there.

Preview Mobile App Pages in Lightning App Builder

For a consistent experience across devices, use the Lightning App builder to preview your record and app pages.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

The Salesforce mobile web experience is no longer available after Summer '20. To customize and preview your apps and pages for Lightning Experience or the mobile app, use the App Builder. You can also specify certain components as mobile- or desktop-only.

If you need more than a preview, run the Lightning apps and components in the Salesforce mobile app. Testing in the mobile app helps validate that users have the best experience on mobile.

When you build a mobile app page, the steps include:

- [Build a new app page with the App Builder](#)
- [Activate the page and add the page to your mobile navigation](#)
- Create a Lightning app in the App Manager for your users to personalize

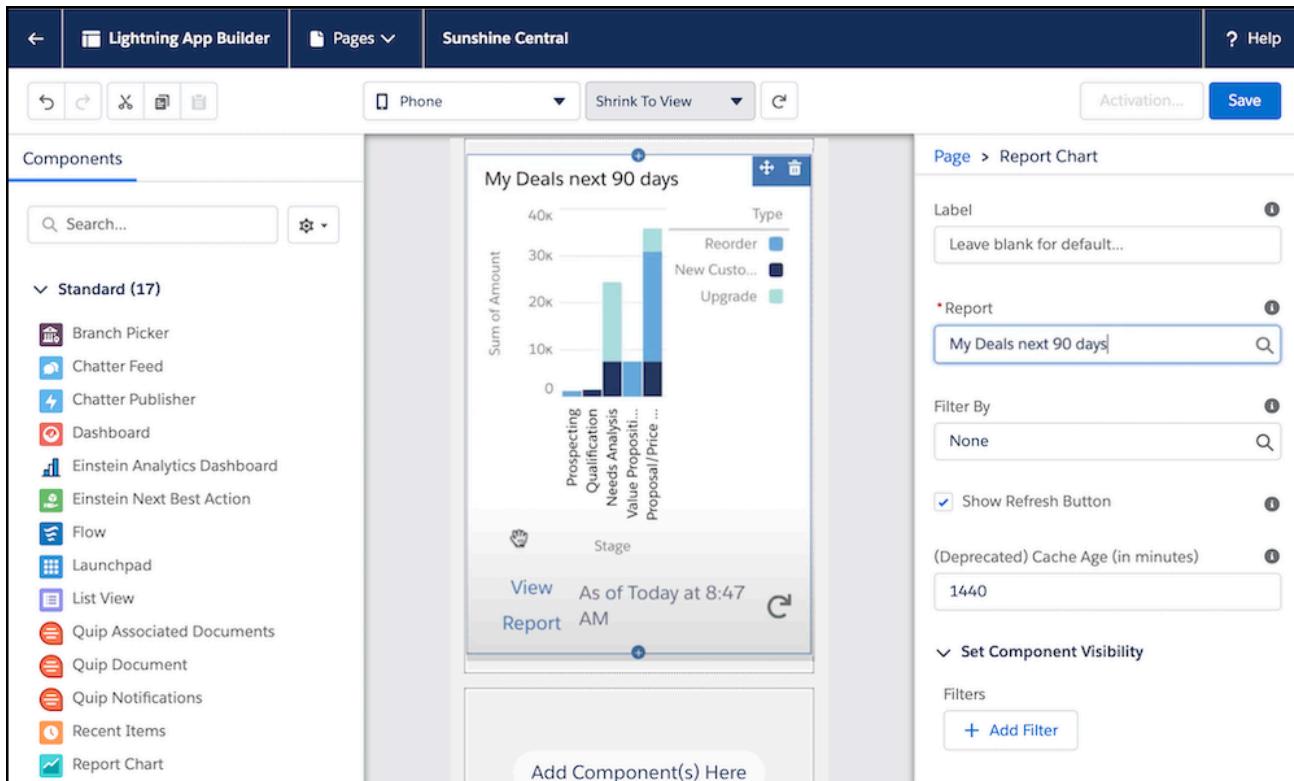


[Watch a Demo](#) (9 minutes)

The following sections discuss best practices on building a mobile app page using the App Builder.

Display Mobile-Friendly Components

When you create a [Lightning app page or record page](#) via the App Builder, the template you choose controls how it displays across devices.



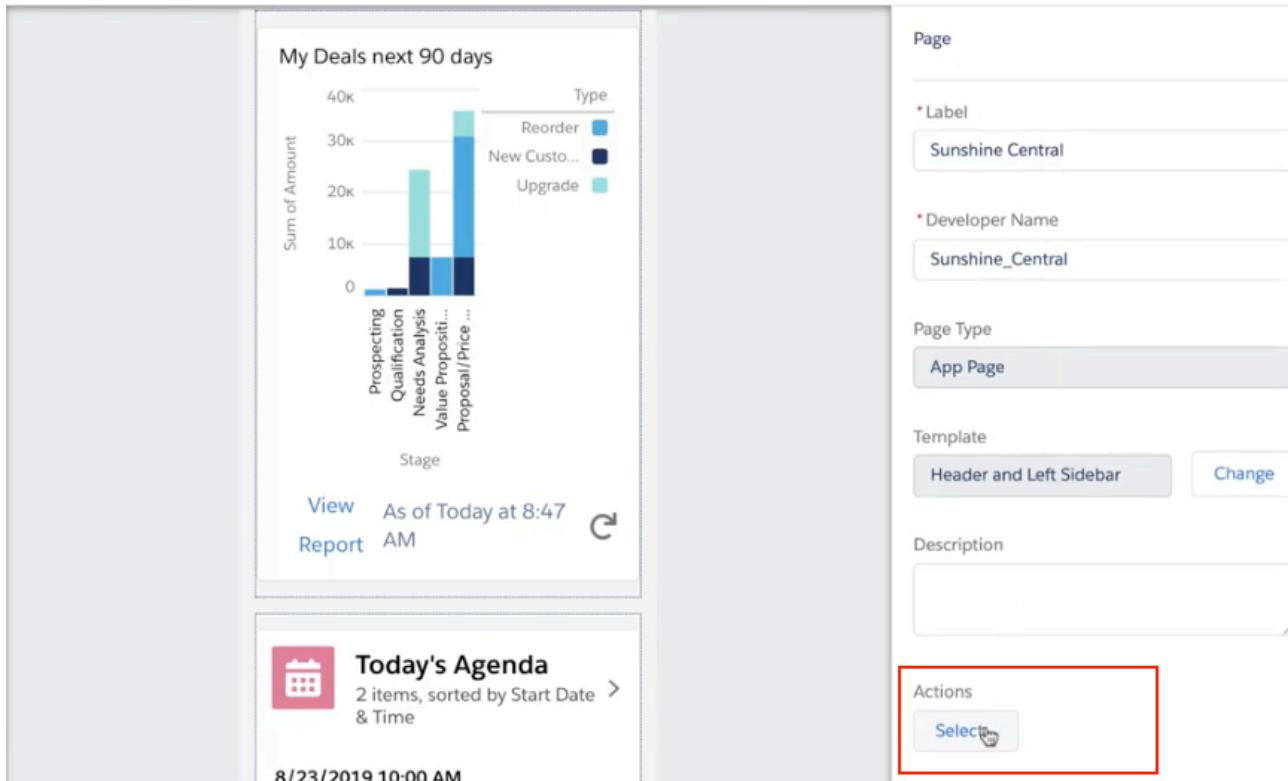
Here are several ideas on including [standard components](#) for mobile users. The list can vary depending on your user's role.

- The Report Chart component displays helpful data, such as your top deals, based on a filter.
- The List View component can display your events, such as today's agenda.
- The Task List component can display your open tasks, overdue tasks, or another list based on a filter.
- The Recent Items component lets you get back to the records you're working with recently.
- The Launchpad component lets you add a grid of navigation items, such as your contacts or a custom Visualforce page, which users can access with one tap.

If standard components don't meet your requirements, create [custom components](#).

Customize Actions for the Mobile Page

If you're creating an app page, you can also customize the actions available for the page. On the **Page** link, click **Select** under the Actions label.



Here are several popular actions for mobile users that make them more productive. Again, the list can vary depending on your user's role.

- Log a call
- New case
- New event
- New task

Activate the Page for Mobile Users

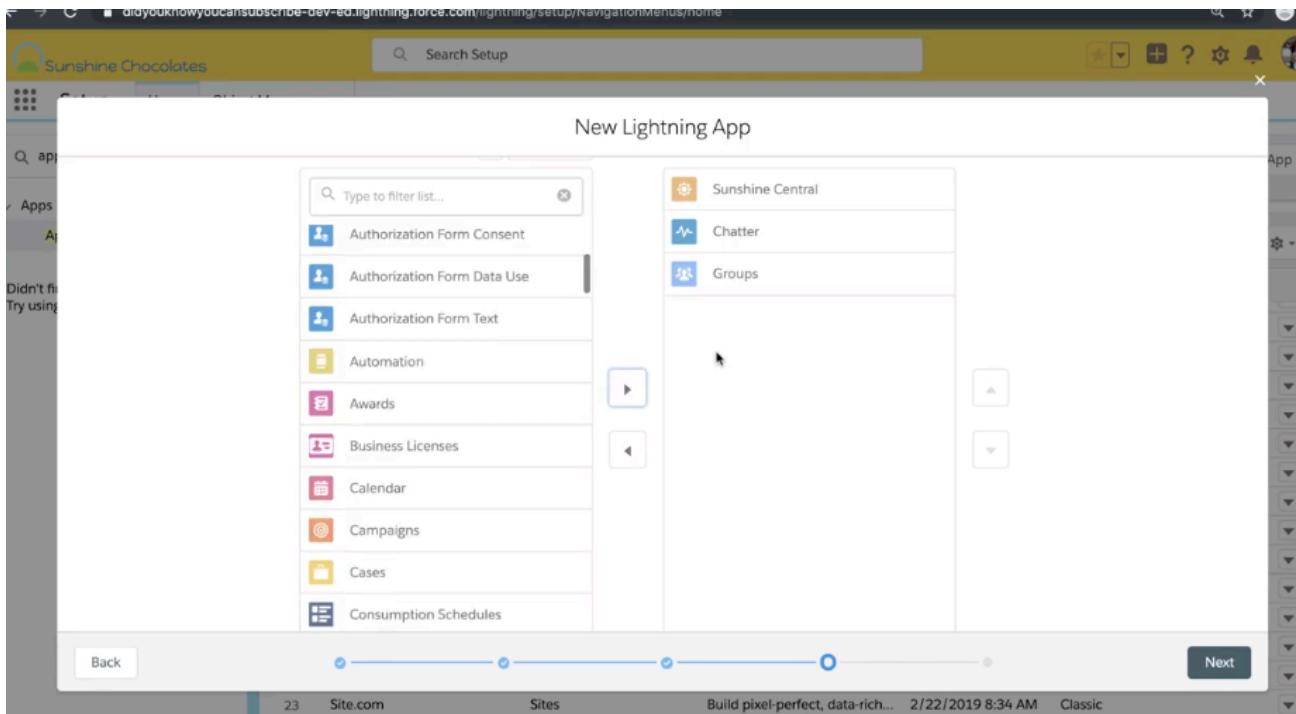
When you [activate the page](#), you have the option to add the page to both Lightning Experience and mobile app. You can add the page to any mobile-enabled Lightning app. Adding your app page to Lightning Experience makes it available to users viewing the app on desktop and in the Salesforce mobile app.

Enable Users to Personalize Tabs for the Mobile App

If you haven't already, enable your Lightning apps for mobile in the App Manager so your users can access the same navigation on all devices. Or create a Lightning app if you want separate navigation items on mobile.

To customize your mobile experience, personalize the app on desktop with your favorite tabs, and then use them on the mobile app. This guideline is optional but highly recommended to optimize your users' mobile experience. For this use case, choose the **Desktop and phone** form factors.

Add navigation items, including the app page you created and several basic tabs like Chatter and Groups. Keep the app lightweight since users can personalize their own tabs using this app.



Add the profiles who need mobile access to this app. For example, the mobile app is ideal for your sales and marketing users who need access to the mobile app on the go. Both profiles can then use the mobile app to personalize the tabs they need for their role.

If you have multiple Lightning apps available to different users' profiles, the profiles also get access to the mobile app.

Personalize Tabs for the Mobile App

In Lightning Experience on desktop, users can access the Lightning app you created. Go to a list view and open it in a new tab. This adds the list view to the mobile navigation and enable users to be productive in the mobile app.



After adding tabs, users can edit the mobile navigation items, edit item names, and reorder the items.



When users log in to the mobile app, the App Launcher lets them switch to the customized Lightning app. For more information, see [Customize the Salesforce Mobile App](#).



The actions you added are displayed on the home page screen with the components you added to the

template in the App Builder. Actions are also visible in preview mode.



Activate Your Lightning App Page

To make your custom app page available to users, you must activate it. You can also rename the Lightning page tab, adjust its visibility, and set its position in the navigation list. The Lightning App Builder's activation feature makes this process easy.

REQUIRED EDITIONS

Lightning App Builder available in: both Salesforce Classic and Lightning Experience

Lightning Home and utility bar pages available in: Lightning Experience

Lightning app and record pages available in: both the Salesforce mobile app and Lightning Experience

Email application pane pages available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create and save Lightning pages in the
Lightning App Builder:

Customize Application

1. To open your app page, from Setup, enter *Lightning App Builder* in the Quick Find box, select **Lightning App Builder**, and then click **Edit** next to the page.
2. In the Lightning App Builder, click **Activation**.
3. Update the activation properties, if desired.
 - Change the page's custom tab label. By default, the label that you give the Lightning page is used as the label for its custom tab.
 - Change the custom tab's icon. The icon that you choose here is used as the icon for the page in the mobile app and in Lightning Experience.
 - Adjust the app page custom tab's visibility.

If you select **System Administrators only**, the tab is set to **Default On** for System Administrators. For all other users, it's set to **Default Off** and doesn't show up in the navigation of the apps it has been assigned to, but users can see the tab in the App Launcher in Lightning Experience and on the All Tabs page in Salesforce Classic.

If you select **Active for all users**, the tab is set to **Default On** for all users. It appears in the visible tabs for its associated app, in the App Launcher in Lightning Experience, and on the All Tabs page in Salesforce Classic.

4. Add the page to one or more Lightning apps.

Adding your app page to Lightning Experience makes it available to users viewing the app on the

desktop and in the Salesforce mobile app.

5. Set the page's position in the Salesforce "Mobile Only" app navigation.

The first item that you put in the mobile navigation list becomes your users' landing page.

6. Click **Activate**.

Your Lightning page is now ready for your mobile and Lightning Experience users!

To enable your users to personalize the app with their favorite tabs, create a Lightning App in the App Manager. They can then use the tabs in the Salesforce mobile app.

 **Tip** You can give your app page a custom icon image by editing the style of the page's custom tab in Setup.

See Also

[Create Lightning Page Tabs](#)

[Tab Settings](#)

Activate Lightning Experience Home Pages

Is your custom home page ready for your users? Use the Activation function inside the Lightning App Builder to get the page out to them. You have three activation options. You can make your custom home page the default home page for all users, assign it to specific Lightning apps, or assign it to app-and-profile combinations. Assigning the page to a Lightning app or app-and-profile combination gives your users access to a home page customized for the context they're working in.

REQUIRED EDITIONS

Lightning App Builder available in: both Salesforce Classic and Lightning Experience

Lightning Home and utility bar pages available in: Lightning Experience

Lightning app and record pages available in: both the Salesforce mobile app and Lightning Experience

Email application pane pages available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create and save Lightning pages in the Lightning App Builder:

Customize Application

1. Create a home page or open an existing one in the Lightning App Builder.
2. If you opened a home page that is ready for your users and doesn't need editing, click **Activation**.
3. If you created a page or opened a page that needs adjustment, make the necessary changes, click **Save**, and then click **Activate**.

4. Choose how you want to activate the page.
 - Make the page the org default.
 - Make the page the default home page for specific Lightning apps.
 - Assign the page to a combination of Lightning apps and profiles.
5. On the activation screen, click the tab for the option you've chosen, and follow the steps to activate the page.



Tip You can also assign home pages from Setup. Enter *Home* in the Quick Find box, and then select **Home**.

- Classic apps can be viewed in Lightning Experience, but you can't display different home pages assigned for specific apps and profiles. Upgrade Classic apps to Lightning apps from the App Manager to make home page assignments.
- If you no longer want a page to be an app or org default, redo the activation process for the page, and select the option to remove it as the default.
- If you activate a page and then return to make changes, you don't have to activate it again. Clicking **Save** after you make your edits pushes the changes to your users.

Add and Customize Tabs on Lightning Pages Using the Lightning App Builder

With the Lightning App Builder, you can create, update, delete, and change the order of tabs and tab sets on record, app, and Home pages in Lightning Experience. Configure the tabs that your users see, name them whatever you like, and add components to each tab.

REQUIRED EDITIONS

Lightning App Builder available in: both Salesforce Classic and Lightning Experience

Lightning Home and utility bar pages available in: Lightning Experience

Lightning app and record pages available in: both the Salesforce mobile app and Lightning Experience

Email application pane pages available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create and save Lightning pages in the Lightning App Builder:

Customize Application

1. Open a Home, app, or record page for Lightning Experience in one of these ways.

- From the Setup menu, select **Edit Page**.



When you select **Edit Page** for the first time, Salesforce makes a copy of the standard page. This

copy is what you edit in the Lightning App Builder. If a customized page exists and is active, selecting Edit Page opens that page to edit.

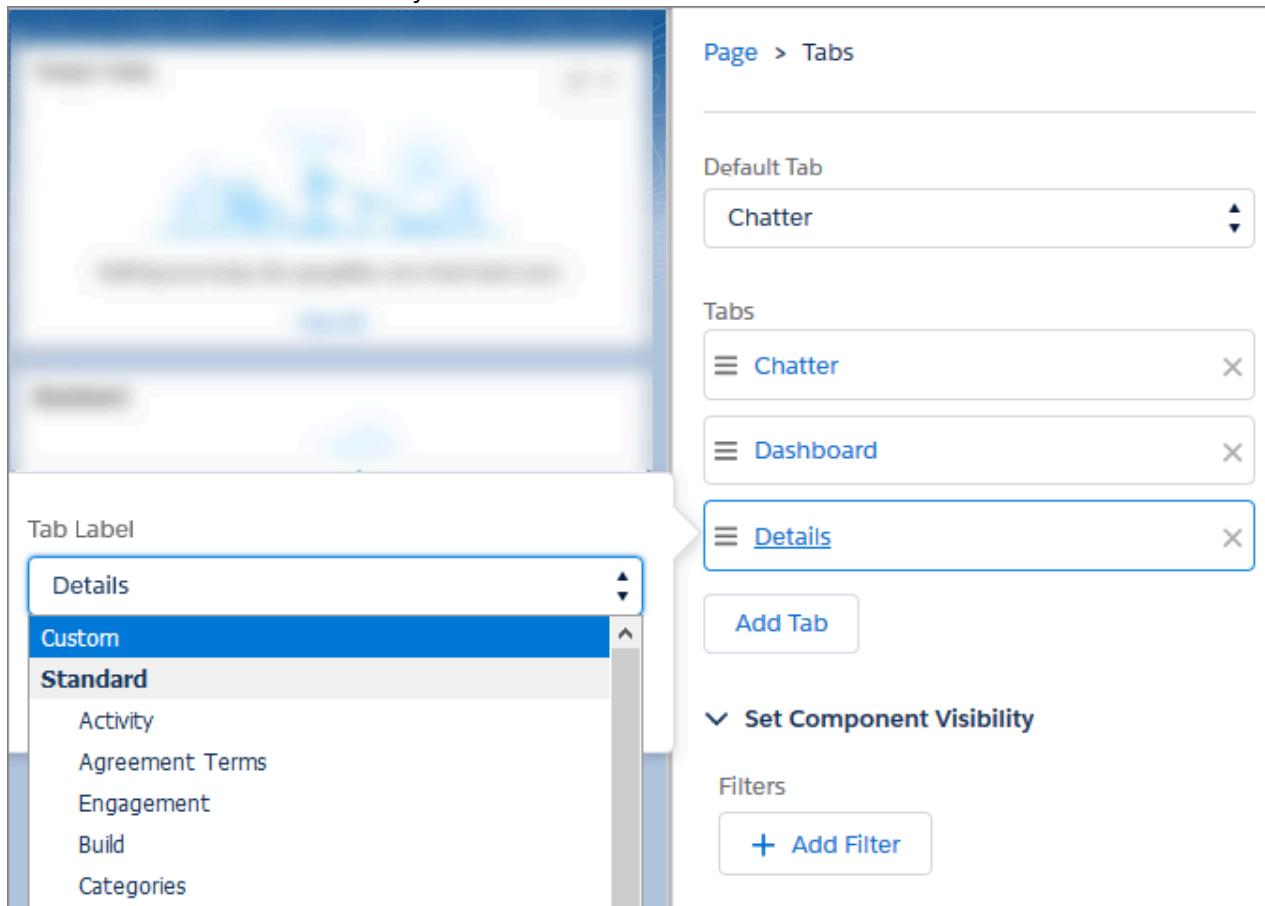
- Open a page from the Lightning App Builder list page. From Setup, in the Quick Find box, enter *App Builder*, and then select **Lightning App Builder**.

2. Add a Tabs component to the page.

A default set of tabs is added.

3. To add a tab, click **Add Tab** in the Tabs component properties.

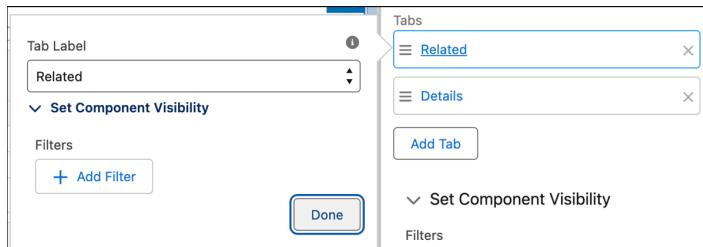
4. Customize a tab by clicking it in the properties pane. You can select a different standard label, or click **Custom** and enter the tab name you want.



Note Custom tab labels in the Tabs component—including those custom tab labels installed from packages—aren't translated. For example, if you create a custom Goals tab in English, then view the page as a user whose language is set to French, the tab still displays as Goals. However, you can use the `{ !$Label.customLabelName }` expression in a component label or attribute to represent a custom label that you create in Setup using the custom label feature. For more information, see [Build Localized Component Labels and Attribute Values on Lightning Pages with Custom Labels](#).

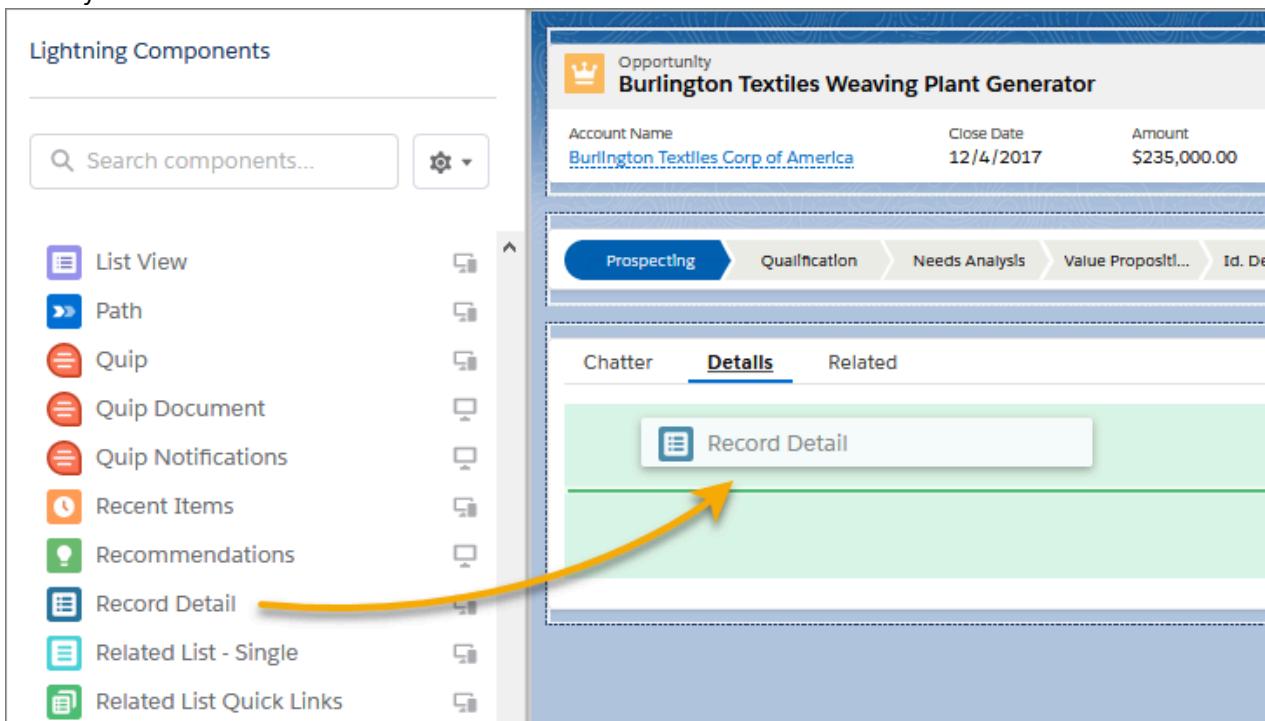
5. If needed, add a visibility rule to show or hide individual tabs.

- a. Click a tab in the Tabs properties pane. For example, click **Related**.
- b. Click **Add Filter** and specify filter criteria, then click **Done**.
- c. Click **Done** again to save your changes.



 **Note** If all components of an individual tab are hidden due to visibility rules, then Salesforce hides the tab, too. For example, if all components of the Related tab are hidden, then Salesforce hides the Related tab. If all individual tabs within a Tabs component are hidden due to visibility rules, then Salesforce hides the entire Tabs component on the record page.

6. To add your first component to a tab, select the tab on the canvas, and then drop the component directly below it.



7. Reorder tabs in the Tabs properties pane by dragging them into position. You can't drag and drop individual tabs on the canvas.

See Also

- [Create and Configure Lightning Experience Record Pages](#)
- [Lightning App Builder Considerations](#)
- [Personalize the Navigation Bar in Lightning Experience](#)

Break Up Your Record Details with Dynamic Forms

The more fields that there are on your page layout, the more that the Record Detail component becomes a monolithic block of fields that you can't customize. With Dynamic Forms, you can migrate the fields and sections from your page layout to individual components in Lightning App Builder. Then you can

configure them the same way as the other components on the page - spreading them out across multiple columns and tabs, and dynamically showing and hiding them based on record data, user details, and device type, so that your users get only the fields and sections that they want.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Dynamic Forms provides:

- An instant upgrade from page layouts: Place fields and sections wherever you want, and use blank spaces to align fields.
- Dynamic layouts: Use visibility rules to show and hide fields and sections.
- Simpler layout management:
 - Manage the fields and sections on your pages in the Lightning App Builder without touching the page layout editor.
 - Reduce the number of page layouts and record types you need with component visibility rules.
 - Get a single assignment model for the Lightning page instead of the dual model of assigning a Lightning page and a page layout.



Some pages have Dynamic Forms enabled by default. For those that don't, you can start using Dynamic Forms in two ways.

- Create a Lightning page from scratch, then drag Field and Field Section components onto it.
- Open an existing record page and [migrate its record details](#) using the migration wizard.

Dynamic Forms doesn't only affect how your users see fields on a record page. It also affects what they see when they click to edit, create, or clone a record. On Dynamic Forms-based pages, the fields that users see when creating, editing, or cloning come from the fields on the page, not from the page layout.

Dynamic Forms is supported for most, but not all standard LWC-enabled objects. See [LWC Migration for Record Home Pages](#) for a list of LWC-enabled objects. If you open a record page for an object in the Lightning App Builder and don't see a Fields tab in the component panel, then Dynamic Forms isn't supported for that object. As an example, the Note object doesn't support Dynamic Forms because it has a fixed layout. Dynamic Forms isn't supported on objects that aren't LWC-enabled. For example, Campaigns, Products, and Tasks, which aren't LWC-enabled, use information from page layout.

Section header labels on Dynamic Forms-enabled pages support using the `{ !$Label.customLabelName }` expression to help define the label's value. See [Build Localized Component Labels and Attribute Values on Lightning Pages with Custom Labels](#).

 **Note** On most LWC-enabled record pages, you can use Dynamic Forms in the record create modal

that displays via a lookup field. This feature is available via the Enable LWC Stacked Modals [release update](#) starting in Summer '24. Salesforce enforces this update in Spring '25.

Migrate a Record Page to Dynamic Forms

Some pages have Dynamic Forms enabled by default. For those that don't, you can migrate the fields and sections from your existing record pages as individual components in the Lightning App Builder. Then you can configure them the same way as the other components on the page, so that your users get only the fields and sections that they want. If you have save options on your account, case, or lead record pages, they are migrated.

Required and Read-Only Fields in Dynamic Forms

Universally required fields retain their status when you're working with them on a Dynamic Forms-based Lightning page. But you can make other fields required or read-only just for the page that you're working on.

Cross-Object Fields in Dynamic Forms

Drill into lookup relationship fields from the component palette on Dynamic Forms-enabled pages in the Lightning App Builder, and access fields from related objects. Drag the cross-object fields onto your record page to display relevant data from related objects.

Align Fields Horizontally in Dynamic Form Field Sections

Control field alignment across columns by using the Align fields horizontally property on Dynamic Forms Field Section components. This property prevents fields in multicolumn Field Sections from collapsing upward when there's a gap due to differences in field heights. Fields remain horizontally aligned with their neighbors in the same row.

Record Page Save Options in Lightning App Builder

With save options, you can define object-specific checkboxes that can be configured as part of a Dynamic Forms-enabled record save. Save options are available when editing an account or when creating, editing, or cloning a case or lead. Optionally, you can set the checkboxes to be selected by default.

Dynamic Forms and Mobile Using the Record Detail - Mobile Component

The Dynamic Forms Field Section component and the Field components that go inside were previously only available on desktop. To support mobile users, the Record Detail - Mobile component gives your mobile users a way to see the record details on their mobile devices on a Dynamic Forms-enabled page.

Enable Dynamic Forms on Mobile

With Dynamic Forms on Mobile, your mobile users can have the same customized experience that your desktop users have.

Dynamic Forms on Mobile Considerations

Keep these considerations in mind when working with Dynamic Forms on Mobile.

Dynamic Forms Tips and Considerations

Keep these tips and considerations in mind when working with Dynamic Forms.

Dynamic Forms Limitations

Keep these general, migration, and packaging limitations in mind when working with Dynamic Forms.

Dynamic Forms Known Issues

Keep these known issues in mind when working with Dynamic Forms.

Migrate a Record Page to Dynamic Forms

Some pages have Dynamic Forms enabled by default. For those that don't, you can migrate the fields and sections from your existing record pages as individual components in the Lightning App Builder. Then you can configure them the same way as the other components on the page, so that your users get only the fields and sections that they want. If you have save options on your account, case, or lead record pages, they are migrated.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

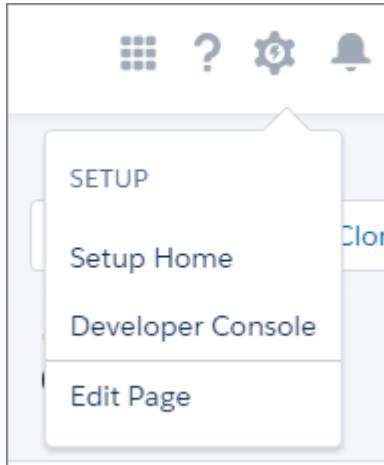
USER PERMISSIONS NEEDED

To create and save Lightning pages in the
Lightning App Builder:

Customize Application

 **Note** Dynamic Forms is supported for most, but not all standard LWC-enabled objects. See [LWC Migration for Record Home Pages](#) for a list of LWC-enabled objects. If you open a record page for an object in the Lightning App Builder and don't see a Fields tab in the component panel, then Dynamic Forms isn't supported for that object. As an example, the Note object doesn't support Dynamic Forms because it has a fixed layout. Dynamic Forms isn't supported on objects that are not LWC-enabled. For example, Campaigns, Products, and Tasks, which are not LWC-enabled, use information from page layout.

1. Open a Dynamic Forms-supported object record page for Lightning Experience in one of these ways.
 - a. From the Setup menu on a record page, select **Edit Page**.



When you select **Edit Page** for the first time, Salesforce makes a copy of the standard page. You edit

this copy in the Lightning App Builder. If a customized page exists and is active, selecting Edit Page opens that page to edit.

- b. From the Object Manager in Setup, open a Dynamic Forms-supported object, then select **Lightning Record Pages**.
 - c. Open a record page for a Dynamic Forms-supported object from the Lightning App Builder list page in Setup. To find it, enter *App Builder* in the Quick Find box, and then select **Lightning App Builder**.
2. Click the Record Detail component.
3. In the component detail pane, click **Upgrade Now** to start the Dynamic Forms migration wizard.



4. Walk through the wizard and select the page layout that has the fields that you want to migrate to the page.

Why choose a page layout when the Fields tab has all the fields that you need? The upgrade wizard takes the fields, sections, and save options from the page layout that you choose and automatically adds them to your page. As long as a field is inside a Field Section component, you can put it anywhere on the page, even inside tabs.

During migration, the Record Detail component is removed from the page and is replaced with fields and sections that you can configure and place anywhere you like.



If your record page supports the phone form factor, a Record Detail - Mobile component is added to your page as part of the migration. This component is unique to Dynamic Forms. It shows the standard record detail fields and sections to your users when they view the page on a mobile device.

If you have opted in to Dynamic Forms on Mobile, the Record Detail - Mobile component isn't added to pages you migrate to Dynamic Forms. The Dynamic Forms content is visible on mobile without the need of the Record Detail - Mobile component.

See Also

[Dynamic Forms and Mobile Using the Record Detail - Mobile Component](#)

[Break Up Your Record Details with Dynamic Forms](#)

Required and Read-Only Fields in Dynamic Forms

Universally required fields retain their status when you're working with them on a Dynamic Forms-based Lightning page. But you can make other fields required or read-only just for the page that you're working on.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

A field set to Required in the field definition in the Object Manager is a universally required field. Its Required value isn't editable in the field's properties in the Lightning App Builder property panel. Also, the required or read-only status of some standard fields—such as Created By, Last Modified By, Owner, and Record Type—can't be changed in the Lightning App Builder property panel.

Fields marked as Required or Read-Only on a page layout retain that state when migrated into a Dynamic Forms-based page.

You can find all universally required fields for your page in the Universally Required Fields section of the Fields palette. Fields that you make required at the page layout level or in the Lightning App Builder property panel don't appear in the Universally Required Fields section of the palette.

If you set a field on a Lightning page to Required or Read-Only in the Lightning App Builder property panel, the behavior applies only to the field on that page, not to all instances of the field.

If a field is set to Required in the Lightning App Builder, it's hidden by a visibility rule at run time, and users can save the record even if that field isn't populated.

Be sure to include universally required fields on your Lightning pages that are used for creating or editing records. They aren't added automatically. If required fields are missing from the page, and the missing required fields don't have values, users can't save a record after editing, creating, or cloning it.

Don't hide universally required fields with visibility filters. Users can't save a record if values are missing in any universally required field, even if they're hidden by a visibility rule.

See Also

[Break Up Your Record Details with Dynamic Forms](#)

[Require Field Input to Ensure Data Quality](#)

[Considerations for Universally Required Fields](#)

Cross-Object Fields in Dynamic Forms

Drill into lookup relationship fields from the component palette on Dynamic Forms-enabled pages in the Lightning App Builder, and access fields from related objects. Drag the cross-object fields onto your record page to display relevant data from related objects.

REQUIRED EDITIONS

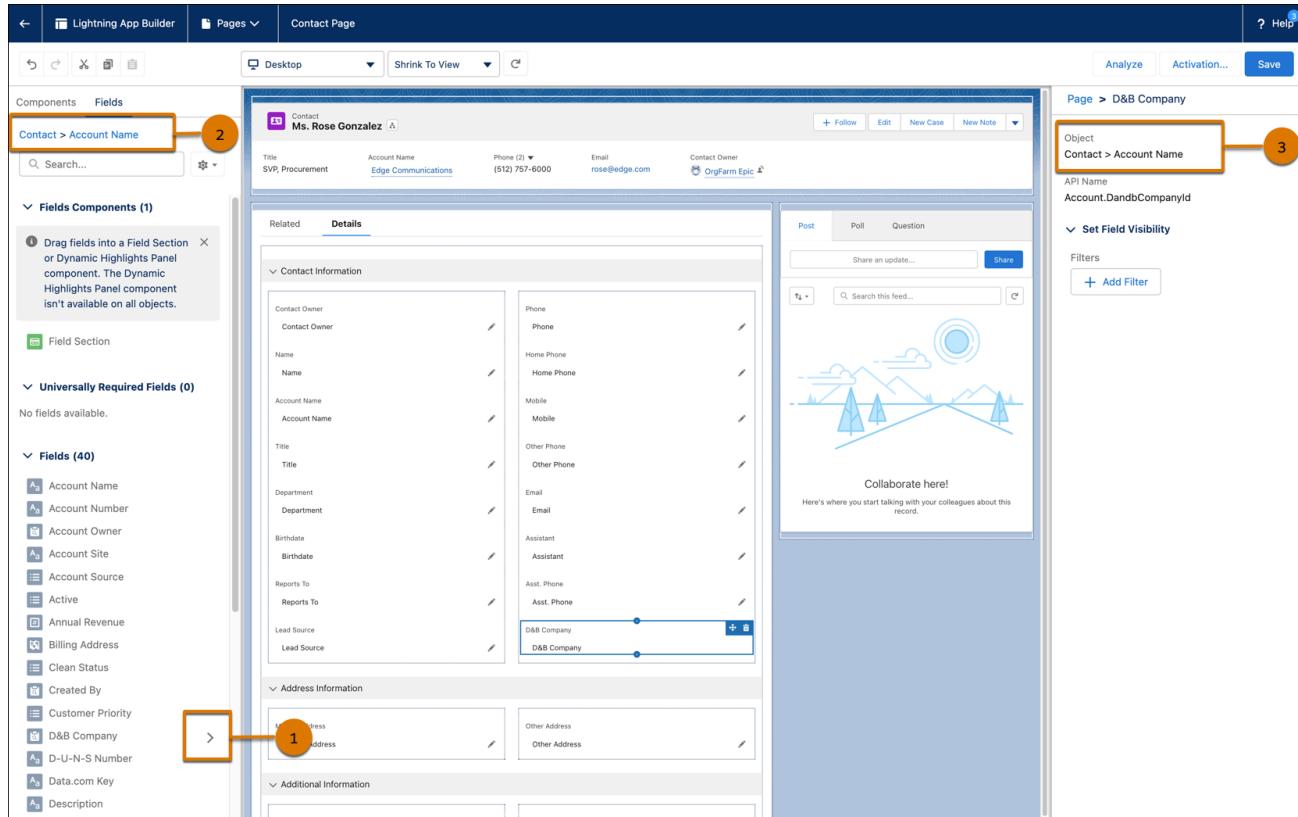
Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Cross-object fields are marked with an arrow icon (>) in the palette (1). Click the arrow icon to drill in.

Every time you drill into a cross-object field from the palette, the fields in the palette change to reflect the fields associated with the object that you clicked into. A breadcrumb at the top of the Fields tab tracks where you are (2).

When you select a cross-object field on the canvas, the Object property shows the relationship of the field to the base object associated with the Lightning page (3).



Dynamic Forms is supported for most but not all standard LWC-enabled objects. For a list of LWC-enabled objects, see [LWC Migration for Record Home Pages](#). However, with cross-object fields in the palette, you can drill into an object that's not LWC-enabled such as Price Book, see its fields on the palette, and then add them to your page. Sometimes fields from non-LWC-enabled objects on a Dynamic Forms-enabled page don't show the same way as they do on their respective object record pages. For example, they can lose their custom formatting. This behavior applies not only to the page in view mode, but also to the create, edit, and clone windows.

Cross-Object Fields and Person Accounts

In orgs with Person Accounts enabled, the behavior of account fields that are cross-object fields at runtime can vary based on the account relationship of the object record that's being viewed.

- On an object record with a person account relationship, general account cross-object fields are always shown, but business account cross-object fields are dropped.
- On an object record with a business account relationship, general account cross-object fields are always shown, but person account cross-object fields are dropped.

- On an object record without an account relationship, the default experience is the business account relationship with general and business account cross-object fields shown and person account cross-object fields dropped.

Limitations

Keep these limitations in mind when working with cross-object fields in Dynamic Forms.

- Polymorphic relationship fields aren't supported as cross-object fields in Dynamic Forms. A polymorphic relationship field is one where the related object could be one of several different types of objects. You can choose a record coming from different objects in a polymorphic lookup. For example, the Who relationship field of a task can be a contact or a lead. You can find polymorphic lookup fields in standard objects.
- User object fields aren't supported as cross-object fields.
- Cross-object fields on Dynamic Forms-enabled pages aren't editable in runtime. As a result, when users create, edit, or clone records that contain cross-object fields, the create, edit, or clone window scrolls to the first editable field, which sometimes aren't at the top of the window.
- You can't drill down more than two levels.

Align Fields Horizontally in Dynamic Form Field Sections

Control field alignment across columns by using the Align fields horizontally property on Dynamic Forms Field Section components. This property prevents fields in multicolumn Field Sections from collapsing upward when there's a gap due to differences in field heights. Fields remain horizontally aligned with their neighbors in the same row.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create and save Lightning pages in the Lightning App Builder: Customize Application

To enable this property, click a Field Section component on a Dynamic Forms-enabled page in Lightning App Builder.



Note Even with the property enabled, if a field is hidden because of visibility rules, the fields in its column still collapse upward to fill the empty space.

 **Example** In this example, two versions of the same field section contain account fields. The left image shows the behavior when the property isn't enabled. On the right, the horizontal alignment setting is enabled, causing the Website field to stay aligned horizontally with Account Name instead of collapsing upward.



See Also

[Break Up Your Record Details with Dynamic Forms](#)

Record Page Save Options in Lightning App Builder

With save options, you can define object-specific checkboxes that can be configured as part of a Dynamic Forms-enabled record save. Save options are available when editing an account or when creating, editing, or cloning a case or lead. Optionally, you can set the checkboxes to be selected by default.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

To use save options, some setup is required.

- Set up account save options like the **Evaluate this account against territory rules on save** option for layout-based pages.
- Set up case save options like the **Case Assignment** and **Email Notification** options for layout-based pages.
- Set up lead save options like the **Lead Assignment** option for layout-based pages.

When a field section is added to an account, case, or lead record page in the Lightning App Builder, Salesforce shows the save options checkboxes. Here is an example of how configuring Save Options for an account appears in Lightning App Builder.



Choose whether to show the assignment rule checkbox and, if so, whether the checkbox is selected by default. If you set up Case Save Options, selected by default, here is how it would appear to someone creating a case.



See Also

[Trailhead Territory Management Basics](#)

[Salesforce Help Set Up Assignment Rules](#)

Dynamic Forms and Mobile Using the Record Detail - Mobile Component

The Dynamic Forms Field Section component and the Field components that go inside were previously only available on desktop. To support mobile users, the Record Detail - Mobile component gives your mobile users a way to see the record details on their mobile devices on a Dynamic Forms-enabled page.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

 **Note** If you haven't yet enabled Dynamic Forms on Mobile, you can still use the Record Detail - Mobile component. But to give your mobile users the same experience that your desktop users have, from Salesforce Mobile App Setup, enable **Dynamic Forms and Dynamic Highlights on Mobile**. [Learn More](#)

When you migrate a record page that supports both desktop and phone to Dynamic Forms, a Record Detail - Mobile component is added to the page for you.

But when you create a fresh page using a template that supports both desktop and phone, after adding the fields and field sections, you must add the Record Detail - Mobile component to the page yourself.

The Record Detail - Mobile component wraps fields from the Record Detail component into a mobile-only container. So, on pages that support both desktop and phone, your desktop users see the Field Section components, and your mobile users see the Record Detail - Mobile component.

 **Example** Here's what a page that you create in desktop mode in the Lightning App Builder looks like when viewed on a phone. The Field Section components just below the Highlights Panel on desktop are dropped on mobile, and the Record Detail - Mobile component takes over to deliver the sections and fields.  The Record Detail - Mobile component displays with an “Unsupported form factor” message when viewed on a page in the Lightning App Builder in Desktop preview mode.

See Also

[Migrate a Record Page to Dynamic Forms](#)

[Break Up Your Record Details with Dynamic Forms](#)

Enable Dynamic Forms on Mobile

With Dynamic Forms on Mobile, your mobile users can have the same customized experience that your desktop users have.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create and save Lightning pages in the
Lightning App Builder:

Customize Application

-  **Note** Dynamic Forms (desktop and mobile) is supported for most but not all standard LWC-enabled objects. For a list of LWC-enabled objects, see [LWC Migration for Record Home Pages](#). If you open a record page for an object in the Lightning App Builder and don't see a Fields tab in the component panel, then Dynamic Forms isn't supported for that object. As an example, the Note object doesn't support Dynamic Forms because it has a fixed layout. Dynamic Forms isn't supported on objects that aren't LWC-enabled. For example, Campaigns, Products, Events, and Tasks, which aren't LWC-enabled, still use information from the page layout.

1. From Setup, in the Quick find box, enter *Mobile*, and then select **Salesforce Mobile App**.
2. Enable **Dynamic Forms and Dynamic Highlights on Mobile**.



But wait! There's more. Now that you've enabled this option, record pages that you create with Dynamic Forms fields after this point are fully mobile-friendly. However, to take advantage of the benefits of Dynamic Forms on mobile, update your existing Dynamic Forms-enabled record pages.

3. From Setup, in the Quick Find box, enter *App Builder*, and then select **Lightning App Builder**.
4. Open a Dynamic Forms-enabled record page.
5. Find the Record Detail - Mobile component on your page.
It's usually found at the bottom of the Details tab.



6. Hover over the Record Detail - Mobile component on the canvas and click  to remove it.
7. Save the page.
8. Optionally, if the page is active, click **Activation** and confirm that the page is activated for the phone form factor.

Dynamic Forms on Mobile works on iOS and Android phones and tablets on all currently supported devices that run the Salesforce mobile app. See [Requirements for the Salesforce Mobile App](#).

-  **Note** If you use an override action to handle default record values with an LWC custom component on a Dynamic Forms-enabled record page, and you have Dynamic Forms on Mobile enabled, to ensure the override works properly on mobile, you must [encode the default record values in your custom LWC component](#).

See Also

- [Dynamic Forms on Mobile Considerations](#)
[Break Up Your Record Details with Dynamic Forms](#)

Dynamic Forms on Mobile Considerations

Keep these considerations in mind when working with Dynamic Forms on Mobile.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

- Dynamic Forms is supported for most but not all standard LWC-enabled objects. For a list of LWC-enabled objects, see [LWC Migration for Record Home Pages](#). If you open a record page for an object in the Lightning App Builder and don't see a Fields tab in the component panel, then Dynamic Forms isn't supported for that object. As an example, the Note object doesn't support Dynamic Forms, because it has a fixed layout.
- Dynamic Forms isn't supported on objects that aren't LWC-enabled. For example, Campaigns, Products, and Tasks, which aren't LWC-enabled, use information from the object page layout.
- Enabling **Dynamic Forms and Dynamic Highlights Panel on Mobile** then disabling it can cause fields and field sections to be removed from your Dynamic Forms-enabled record pages when viewed on mobile. After disabling **Dynamic Forms and Dynamic Highlights Panel on Mobile**, add the Record Detail - Mobile component to your Dynamic Forms-enabled record pages so that mobile users can still see record fields.

If you no longer want to use Dynamic Forms, remove any field sections and fields from your record pages, then add the Record Detail component instead.

- Enabling **Dynamic Forms and Dynamic Highlights Panel on Mobile** changes the behavior of help text on mobile devices from a modal style to a [lightning-helptext](#) tooltip.
- Record-home offline priming isn't supported. If you're using AURA-based priming, you can't pre-cache your most recently used records.
- Offline draft edits to record pages aren't supported when Dynamic Forms on Mobile is enabled.
- Visualforce pages appear when viewing a Dynamic Forms-enabled record page on mobile, but not when editing it.

-  **Note** Keep working even when disconnected from the internet with Mobile Offline, available in [Salesforce Mobile App Plus](#).

See Also

[Enable Dynamic Forms on Mobile](#)[Break Up Your Record Details with Dynamic Forms](#)

Dynamic Forms Tips and Considerations

Keep these tips and considerations in mind when working with Dynamic Forms.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

General Considerations

- Lightning App Builder preview respects density settings for field sections with one exception. If you have the Compact density setting enabled, you don't see the field labels and field values aligned on the same row on the Lightning App Builder canvas.
- When you switch page templates on a page that uses Dynamic Forms, the list of available templates contains only the templates that are supported for your Dynamic Forms-enabled page.
- Some fields get special handling in Dynamic Forms. They're set to Read-Only or Required and that state isn't editable in the properties panel.
- If you add more than one instance of the same field to a Lightning page, all instances of that field on the page show the same data. However, two or more instances of the same field on a page with different visibility rules, UI behavior, or requiredness settings can cause problems with the page at run time, including issues with record edit behavior.
- Programmatic versions of the accordion components don't provide the same functionality as their App Builder counterparts. For example, `lightning-accordion` and `lightning:accordion` components don't currently support lazy loading.
- Expanding or collapsing a field section while designing a page has no effect on whether a section is expanded or collapsed for users during runtime.
- When you save a new custom object record with an empty object name, the API populates the custom object name field with the record ID. Make sure that the Name field is on the page and is marked as required.
- When Dynamic Forms-enabled pages display on a tablet, field sections inside tabs have more left padding than field sections outside of tabs.
- When adding new custom fields to Dynamic Forms-enabled Lightning pages as part of field creation, fields are added as the last field in the first column on the first field section on the page in top-down, left-to-right order. If the field section is in a container, such as a Tabs or Accordion component, then the first field section is chosen from the first tab or the first accordion section. The “active” tab doesn't matter. If the first field section on a page is the Dynamic Highlights Panel, then new fields are added to the second field section instead.
- When you view a Dynamic Forms-enabled record page on a mobile device, tapping the address field results in different behavior depending on whether Dynamic Forms on Mobile is enabled in your org.

When Dynamic Forms on Mobile isn't enabled, tapping an address field opens a prompt to select a map app. When Dynamic Forms on Mobile is enabled, tapping an address field takes you immediately to Google Maps™.

- When you create a record from a Dynamic Forms-enabled record page, the success confirmation toast message doesn't include a link like it does when you create a record from standard record pages.
- When you click a compound address field from a Record Detail component, it opens in Google Maps™ with just the address populated. When you click a compound address field from a Dynamic Forms-enabled page, it opens in Google Maps showing both the address and the latitude and longitude values. However, if the latitude and longitude values point to an area with no known address, Google Maps doesn't show an address.
- The User Interface setting Enable Collapsible Sections doesn't apply to Lightning record pages when Dynamic Forms is enabled on the page. This setting applies only to record pages in Salesforce Classic and to Lightning record pages that use the Record Detail component. The Dynamic Forms expand and collapse functionality takes precedence over the Enable Collapsible Sections setting.

General Tips

- We recommend not having both a Record Detail component and field sections on the same page. If you do, users can have issues with the page, including:
 - Multiple, overlapping save and cancel bars when both are on the page and both in inline edit.
 - Visibility rules on Field and Field Section components not working properly.
 - When users create, edit, or clone a record that uses Dynamic Forms, the fields they see come from the field sections on the page, not from the Record Detail component.
 - Field sections are top-down-left-right tab order only, while Record Detail is left-right-top-down tab order, which can cause confusion. As a result, fields inside a Field Section sometimes don't line up horizontally, while Record Detail fields do. If you want the fields to retain their horizontal alignment inside a field section, select the **Align fields horizontally** checkbox.
 - Dynamic actions are supported on custom objects. Actions on standard objects come from the record details on the page layout, not from the Dynamic Forms fields on the page.

Mobile Tips

 **Note** Dynamic Forms on Mobile gives your mobile users the same experience that your desktop users have. To avoid these mobile considerations, from Salesforce Mobile App Setup, enable **Dynamic Forms on Mobile**.

- Don't place Field Section components on mobile-only Lightning pages. Field Section components are desktop-only and don't appear when the page is viewed on a phone.
- You can use one Lightning page for both desktop and phone. Add the Record Detail - Mobile component to the same page with your Field Section components. Your desktop users see the Field Section component, and your mobile users see the Record Detail - Mobile component.

Visibility Rules and Dynamic Forms

- Field and Field Section components are supported on desktop only. Therefore, the visibility rules that you set on them are respected only in the desktop view, not on mobile. However, Dynamic Forms on Mobile gives your mobile users the same experience that your desktop users have. To avoid this issue, from Salesforce Mobile App Setup, enable **Dynamic Forms and Dynamic Highlights on Mobile**.
- If a field is hidden on a page due to a visibility rule, and a user edits the value of the field, its new or changed value isn't saved.
- Dependent picklist fields don't respect visibility rules. Even if they have visibility rules assigned to them, dependent picklist fields still appear in the View All Dependencies list.
- Component visibility rules on field sections behave differently than they do on fields. Visibility rules on fields are assessed dynamically. Changes a user makes while editing a record can make fields appear and disappear as visibility rules are evaluated. Visibility rules on field sections aren't dynamic and don't react to what a user does while editing. Field section visibility rules are evaluated only after the record is saved.
- If you hide a field with a visibility rule, users don't lose access to the field's data. Values in hidden fields remain intact and are visible outside of the Dynamic Forms-based page in places like reports, dashboards, and list views. Hidden fields are for user convenience, not for field-level security.
- When a tab is hidden by a visibility rule, any field sections within that tab are hidden in the Create and Edit windows. If all field sections on a page are contained in the hidden tab, then the Create and Edit windows are blank.

See Also

[Dynamic Forms Limitations](#)

[Dynamic Forms Known Issues](#)

[Break Up Your Record Details with Dynamic Forms](#)

Dynamic Forms Limitations

Keep these general, migration, and packaging limitations in mind when working with Dynamic Forms.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

General

- Dynamic Forms is supported for most but not all standard LWC-enabled objects. See [LWC Migration for Record Home Pages](#) for a list of LWC-enabled objects. If you open a record page for an object in the Lightning App Builder and don't see a Fields tab in the component panel, then Dynamic Forms isn't supported for that object. As an example, the Note object doesn't support Dynamic Forms because it has a fixed layout.

- Dynamic Forms isn't supported on objects that aren't LWC-enabled. For example, Campaigns, Products, and Tasks, which aren't LWC-enabled, use information from page layout.
- Dynamic Forms doesn't work in Internet Explorer 11. Users with IE 11 who try to view a page that uses Dynamic Forms get a page error.
- Dynamic Forms doesn't work on pages based on custom templates.
- You can add up to 100 fields per column in a Field Section component. A blank space component is considered a field and counts against the limit of 100 fields per region. It doesn't count against the component page limit.
- The columns inside a Field Section are considered components. A Field Section component with two columns counts as three components when determining the total components on a page.
- If you convert a section from two columns to one, the new single column can only hold 100 fields. If you have more than 100 fields total between the two columns, when you switch to a single column, the first 100 fields are moved into the single column, and the rest of the fields are dropped. If you change your mind after switching to single column, you can revert the change by immediately clicking the undo button ().
- Users can expand or collapse field sections only while in view or inline edit mode, not in the full edit, create, or clone windows. If you expand or collapse a field section and then refresh your browser, including pulling to refresh, the expanded or collapsed state doesn't persist.
- If you have Opportunity Products enabled and have the Prompt users to add products to opportunities setting enabled, when you create an opportunity from either an opportunity-related list or from opportunity home, the create window uses record details from the page layout.
- The Clone with Related action for opportunities uses record details from the page layout even if the opportunity page you launch the action from uses Dynamic Forms.
- When viewing an external or non-UI API-supported object record that has a related list for a child object that supports Dynamic Forms, if you create a record for that child object from the parent record related list, the create window uses record details from the child object page layout.
- When configuring key fields in a path, the fields in the Available Fields list come from the record details on the object's page layout, even if the page the path resides on uses Dynamic Forms.
- The fields available for inline editing in list views are determined by the fields on the page layout, not the ones configured with Dynamic Forms.

Migration

- Only fields, sections containing fields, and layout save options are included when migrating a page to Dynamic Forms. Other elements on the page layout, such as custom links and blank spaces, aren't included.
- You can add up to 100 fields per column in a Field Section component. And you can add up to 100 sections per region. If you use the migration tool to import fields and sections from a page layout that has more than those limits, then we migrate up to the first 100 sections in each region and the first 100 fields in each column. The rest of the fields are dropped, even if they're required fields. You can manually add the dropped items later. If you change your mind after you make the switch, you can click the undo button () to roll back the changes.
- The Lightning App Builder Record Detail to Dynamic Forms migration tool doesn't migrate all possible layout save options. For example, when saving a layout, if **Default** is selected but **Show on edit page** is deselected, then this option isn't migrated. Manually set this value after migration.

Packaging

- Users who don't have a license for a managed package don't see fields from that managed package in the Lightning App Builder palette.
- When a page that contains fields from a managed package is updated by an admin that doesn't have a license to that managed package, the fields from that managed package are removed from the page upon saving.

See Also

[Knowledge Article: Extended Support for Accessing Lightning Experience Using Microsoft Internet Explorer 11](#)

[Dynamic Forms Tips and Considerations](#)

[Dynamic Forms Known Issues](#)

[Break Up Your Record Details with Dynamic Forms](#)

Dynamic Forms Known Issues

Keep these known issues in mind when working with Dynamic Forms.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

- If you click **View All Dependencies** from the Path component, the dependencies that you see come from the record details on the page layout, not from the Dynamic Forms fields on the page.
- The CurrencyISOCode field doesn't appear as required (with the asterisk) at runtime, although it still behaves as required.
- When you hover over fields in the palette, the data types shown are inaccurate.
- Fields in two-column Field Section components don't horizontally align correctly. If you want the fields to retain their horizontal alignment inside a field section, select the **Align fields horizontally** checkbox.
- A record's printable view is based on the fields from its default page layout, not on the fields on the Dynamic Forms-based page.
- Dynamic Forms field components aren't supported inside the Macro Builder. When a Dynamic Forms-enabled record page is opened inside of the Macro Builder, the field components don't appear.
- Universally required fields that are only required for either person account or business account appear in the Universally Required Fields section of the Fields panel regardless of which record type you're building the page for. For example, if you're creating a page for the business account record type, you see both person account and business account required fields in the Universally Required Fields section. If the person account required field isn't present on the page when you save the business account page, you get a warning even though the person account field isn't needed for the business account page.
- The Einstein Account Tier field is missing from the Record Detail component on custom account

record pages in the Lightning App Builder and when the account record page renders for users. If you migrate a custom account record page to Dynamic Forms, the Einstein Account Tier field appears in the available fields list and on the record page in Lightning App Builder, but doesn't appear when the page renders for users.

- You can define a Lightning page that supports multiple accounts, like a business account and a person account. You can't designate different save option values for the business account or person account flow on that Lightning page.
- When you create an object from a related list in the context of a person account using Dynamic Forms, the **PersonContactId** field isn't prepopulated in the new record.
- If you edit a field inline and don't save, then use a quick action to edit and save, your inline edits aren't saved when your quick action edit is.
- If you move a record to a different status in Kanban view, and the status change triggers a field validation error, it opens the edit window but doesn't show the error.
- Some field section header labels that are created when migrating a Dynamic Forms page don't have the same label as the page layout section header. You can update the label in Lightning App Builder.

Packaging Known Issues with Dynamic Forms

- In a non-namespaced org, conflicts can happen if you install a managed package with a Dynamic Forms-enabled Lightning page that's based on an object that exists in the subscriber org and that contains a custom field with the same API name as a custom field on the same object in the subscriber org. The API name conflict between the managed package's custom field and the subscriber org's custom field causes the installed managed Lightning page to reflect the subscriber org's fields instead of the fields from the managed package.
This issue can also happen with prompt templates installed from a managed package. If the installed prompt template is referenced by a custom field on a Dynamic Forms-enabled Lightning page, and that prompt template itself references a custom field with a namespace that is different from the Lightning page's namespace, package deploy can fail.
Namespaces are used to differentiate custom object and field names from other orgs' names. For more information about namespaces in orgs, see [Understanding Namespaces](#).

Visibility Rule Known Issues with Dynamic Forms

- Visibility rules based on parent object fields don't work correctly.
- When you clone a record based on a Dynamic Forms-enabled page that has visibility rules, all the fields referenced in the visibility rules are added to the cloned record, regardless of whether those fields are on the page. The field value used in the rule evaluation and the field's saved value on the cloned record can vary, depending on the cloning user's access to the field used in the visibility rule.
 - If the user has read and write access to the field used in the visibility rule, and the field has a value, the source record's original value is used in the rule evaluation and saved to the cloned record.
 - If the user has read and write access to the field used in the visibility rule, and the field has no value, the source record's blank value is used in the rule evaluation, but the cloned record saves with the field's default value.
 - If the user has only read access to the field used in the visibility rule, the source record's original value is used in the rule evaluation, but the cloned record saves with the field's default value.

- If the user doesn't have read or write access to the field used in the visibility rule, the source value is treated as blank, and the cloned record saves with the field's default value.
- Values saved in the cloned record of a Dynamic Forms-enabled page depend on the cloning user's access to the field even when there are no visibility rules assigned. This behavior is different from [cloning record detail component values](#) on non Dynamic Forms-enabled pages.

Create from Lookup Known Issues with Dynamic Forms

- Creating a record from a lookup field opens a window with the Dynamic Forms fields only if the object is LWC-enabled. For example, Campaigns, Products, and Tasks still use information from the page layout. See [LWC Migration for Record Home Pages](#) for the list of LWC-enabled objects.
- When working with actions, using Dynamic Forms via a Create from Lookup field is supported only for standard actions. Other types of actions, such as default actions and custom quick actions, continue to use information from the page layout.
- On a Dynamic Forms-based page, you get a page error when there are multiple instances of the same lookup field that look up to an object that's not supported in the UI API. See the [User Interface API Developer Guide](#) for the list of supported objects.
- For a non-UI API supported object that has a Forms-enabled object as a related list, if you click **New** from the related list, the parent lookup field doesn't appear in the New dialog. But you can save the new record.

See Also

[Dynamic Forms Tips and Considerations](#)

[Dynamic Forms Limitations](#)

[Break Up Your Record Details with Dynamic Forms](#)

Dynamic Highlights Panel

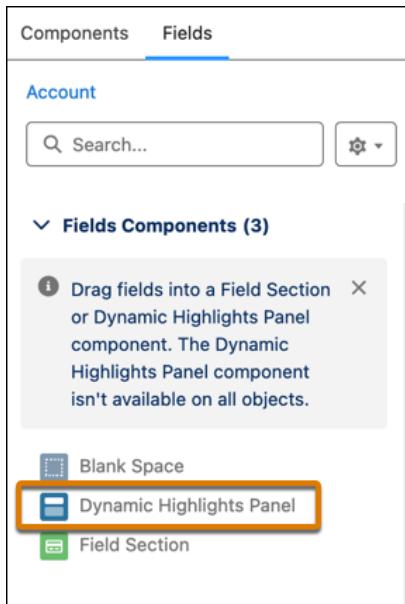
The Dynamic Highlights Panel component displays up to 12 key record fields, which are configured in Lightning App Builder. The Dynamic Highlights Panel is responsive. When you change the size of your browser window, your information stays visible and wraps instead of being truncated.

REQUIRED EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** editions

The Dynamic Highlights Panel is a container for fields, just like a Field Section. You can find it on the Fields tab in the Lightning App Builder.



When you add the Dynamic Highlights Panel to the canvas, a Primary Field value is preset to the object's Name field. You can change the Primary Field value in the property pane. If no secondary fields are added to the panel, the panel displays collapsed in runtime.

You can use visibility rules to show and hide fields in the panel. You can also apply conditional formatting to fields in the panel to make them stand out and help users quickly identify the most relevant information on the page.

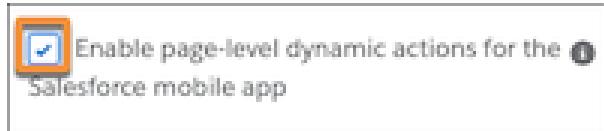
You can also customize the actions that appear in the panel. The Dynamic Highlights Panel is supported for all LWC-enabled objects. For a list of LWC-enabled objects, see [LWC Migration for Record Home Pages](#).

The Dynamic Highlights Panel must be enabled for mobile before your mobile users can see it on record pages. You can enable the panel on the Salesforce Mobile App page in Setup.

When you enable **Dynamic Actions on Mobile**, some visibility rules may not work on standard objects as expected. Enable the **Dynamic Forms and Dynamic Highlights Panel on Mobile** setting at the same time. Enabling both settings ensures record pages on mobile use the latest LWC-based Lightning page technology, which provides improved support for visibility rules.

The Dynamic Highlights Panel uses dynamic actions by default. Actions on pages that use the panel can come from different places based on how and where actions are configured and whether Dynamic Actions on Mobile is enabled.

- If **Dynamic Actions on Mobile** isn't enabled for your org, desktop and mobile users viewing a page with the Dynamic Highlights Panel see the actions configured inside the Dynamic Highlights Panel. If the Dynamic Highlights Panel has no actions configured, desktop and mobile users see actions from the page layout. You can enable Dynamic Actions on Mobile from the Salesforce Mobile App page in Setup.
- If Dynamic Actions on Mobile is enabled for your org, a checkbox appears in the page properties on record pages in Lightning App Builder.



If you don't select the checkbox, mobile and desktop users see the actions from the Dynamic Highlights Panel, if configured, and from the page layout if not configured.

When this option is selected, the page-level actions you configure in the page properties pane take precedence on mobile over all other actions on the page, including those configured in the Dynamic Highlights Panel, the Highlights Panel, or from the page layout.

See Also

[Dynamic Highlights Panel Considerations and Limitations](#)

[Get Help for Lightning App Builder](#)

[Conditional Field Formatting in Lightning App Builder](#)

[Visibility Rules on Lightning Pages](#)

Dynamic Highlights Panel Considerations and Limitations

Keep these considerations and limitations in mind when using the Dynamic Highlights Panel.

REQUIRED EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** editions

Considerations

Keep these considerations in mind when using the Dynamic Highlights Panel.

- If the Primary Field is hidden for a user due to field-level security settings, the object's Name field is set as the Primary Field value for that user at runtime.
- When you disable **Dynamic Forms and Dynamic Highlights Panel on Mobile**, the Dynamic Highlights Panel component is removed from record pages when they're viewed on mobile. To make sure that mobile users can still see record highlights, add a regular Highlights Panel component to any page that contains a Dynamic Highlights Panel. Then create a visibility rule for the Highlights Panel that sets it to appear only on mobile devices. Desktop users then see the Dynamic Highlights Panel, and mobile users see the Highlights Panel.

If you no longer want to use the Dynamic Highlights Panel component, remove it from your record pages, and add a regular Highlights Panel component instead.

- Fields in the Dynamic Highlights Panel are read only. Inline edit isn't supported for Dynamic Highlights Panel fields, nor are they included in the available fields when you edit a record.
- Dynamic Highlights Panel fields are fixed width and don't dynamically adjust based on the field values.
- When adding new custom fields to Dynamic Forms-enabled Lightning pages as part of field creation,

those fields are added as the last field in the first column on the first field section on the page in top-down, left-to-right order. If the first field section on a page is the Dynamic Highlights Panel, then new fields are added to the second field section instead.

- When you enable **Dynamic Actions on Mobile**, some visibility rules may not work on standard objects as expected. Enable the **Dynamic Forms and Dynamic Highlights Panel on Mobile** setting at the same time. Enabling both settings ensures record pages on mobile use the latest LWC-based Lightning page technology, which provides improved support for visibility rules.

Limitations

Keep these limitations in mind when using the Dynamic Highlights Panel.

- Blank spaces aren't supported.
- The Dynamic Highlights Panel doesn't support the merging of multiple address or phone fields. It also doesn't support a dropdown-menu version of those fields.
- On pages that use the Dynamic Highlights Panel, the panel scrolls with the page. The panel's primary field and actions don't stick at the top when you scroll.
- Visibility rules aren't supported for the Primary Field, but you can use them to show and hide the other fields in the panel.
- You can't select a cross-object field as the Primary Field for the Dynamic Highlights Panel in the Lightning App Builder.

Create Dynamic Actions in Lightning App Builder

With dynamic actions, you can add flexibility and control to actions on your record pages. Assign dynamic actions in the Lightning App Builder instead of in the page layout editor, and then apply filters to control when and where actions are visible to users.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** editions

USER PERMISSIONS NEEDED

To create actions:	Customize Application
--------------------	-----------------------

To create and save Lightning pages in the Lightning App Builder:	Customize Application
---	-----------------------

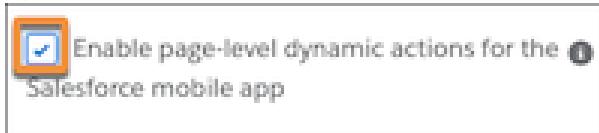
You can create dynamic actions for standard and custom objects on mobile and desktop. Dynamic actions are supported by default on custom objects on mobile. To enable dynamic actions on standard objects, from Setup, in the Quick find box, enter *Mobile*, and then select **Salesforce Mobile App**. Enable **Dynamic Actions on Mobile**.

Check out this video to see how dynamic actions work on mobile devices.

Watch the video: <https://play.vidyard.com/nX6qa9qy2rNCS8yWtSvbqA>

-  **Note** The Group object doesn't respect dynamic actions assigned in the Lightning App Builder. You can add or update actions on groups from the Group Layouts page in Setup. Custom Lightning page templates with dynamic actions aren't supported in Salesforce Mobile.

1. From Setup, in the Quick Find box, enter *App Builder*, and then select **Lightning App Builder**.
2. Edit an existing record page, or click **New** to create one.
3. Add or select the Highlights Panel component on the object's record page.
4. To enable Dynamic Actions, in the Highlights Panel properties pane, click **Upgrade Now**, and either step through the migration assistant to migrate existing actions or start with new actions.
When you enable Dynamic Actions, it applies only to the record page that you're currently working on in Lightning App Builder. Users can then see the dynamic actions configured for that page in Lightning App Builder. Users don't see actions configured in the object's page layout.
5. If you want to use a different set of actions for mobile than for desktop, or if you're adding dynamic actions to a mobile-only Lightning page, go to the page Properties pane, and select **Enable page-level dynamic actions for the Salesforce mobile app**.



6. In the Properties pane, click **Add Action**, and then choose an action from the Actions window.
-  **Note** If you migrated actions from a page layout, you can modify them in the Actions window.
7. To add visibility rules based on the record field, device type, or other filters, click **Add Filter**.
An eye icon  next to an action's name indicates that visibility rules are applied. Visibility filters support a single value at a time, not comma-separated values. To add more than one value, add a new filter.
 8. To save your changes, click **Done**.

By default, if you configure Dynamic Actions for a record page in a Highlights Panel or Dynamic Highlights Panel component, mobile and desktop users see the same actions. If your desktop page has multiple Highlights Panel components and you add one or more dynamic actions to each of them, the dynamic actions are consolidated into one action bar for that record page on mobile.

-  **Note** For visibility rules on dynamic actions in the Highlights Panel or Dynamic Highlights Panel, if a user doesn't have access to a field used as part of the action's visibility filter criteria, the criteria don't apply, and therefore evaluate to true.

When **Enable page-level dynamic actions for the Salesforce mobile app** is selected, the page-level actions that you configure in the page properties pane take precedence on mobile over all other actions on the page, including those configured in the Dynamic Highlights Panel, the Highlights Panel, or from the page layout.

Create Dynamic Related Lists in Lightning App Builder

With dynamic related lists, you can add flexibility and control to related lists on your record pages. Add, customize, and filter related lists in the Lightning App Builder instead of in the page layout editor. To help your users see the most relevant records, set up two or more dynamic related lists with different filters on the same object.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

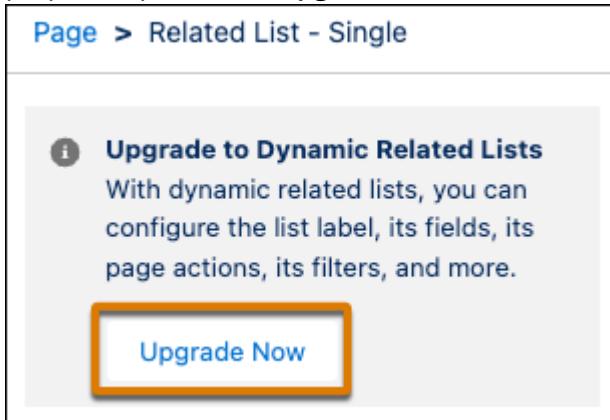
USER PERMISSIONS NEEDED

To create dynamic related lists: Customize Application

To create and save Lightning pages in the Lightning App Builder: Customize Application

You can create dynamic related lists for custom objects and for Salesforce record home objects that are enabled for LWC. Dynamic related lists are supported by default on desktop. To use them on mobile, enable Dynamic Related Lists for Mobile (Beta) in Salesforce Mobile App Setup.

1. From Setup, in the Quick Find box, enter *App Builder*, and then select **Lightning App Builder**.
2. Edit an existing record page, or click **New** to create one.
3. To add a new dynamic related list to your record page, drag the Dynamic Related List–Single component onto the page.
4. To upgrade an existing list, select a Related List–Single component on the record page. Then, in the properties pane, click **Upgrade Now** to convert it to a Dynamic Related List–Single component.



When you upgrade a list, Salesforce uses your assigned page layout to populate the fields and actions in the properties pane. If a related list isn't defined on the object's page layout, fields populate the properties pane in the default order defined by Salesforce, and actions don't populate automatically.

If you don't see the option to upgrade, Dynamic Related List-Single isn't supported for the object associated with the record page or for the related list selected in the properties pane.

5. In the Dynamic Related List-Single properties pane, customize the list.
 - Select the related list to add to the page and which record the list comes from.
 - Give the list a descriptive name.
 - Select a list type to define how the list appears on the page, and choose the number of records to show in the list.
 - Show or hide actions on the list, and choose the actions that are available.
 - Select and order the fields to show as columns in the list, and define the record sort order.
 - Filter the records in the list. Users can't remove the filters that you apply.
6. To add visibility rules based on the record field, device type, or other filters, click **Add Filter**.
The  on the component indicates that visibility rules are applied.
7. Save your changes to the record page, and then activate the page to share it with your users.

See Also

[Standard Lightning Page Components](#)

Conditional Field Formatting in Lightning App Builder

Make a field stand out by adding special formatting based on rules and criteria you provide, so that end users can quickly identify the most relevant information on a record page. You can apply conditional formatting to almost any field on a Dynamic Forms-enabled page, including the fields in the Dynamic Highlights Panel component.

REQUIRED EDITIONS

Lightning App Builder available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Lightning pages and Dynamic Forms available in: Lightning Experience and the Salesforce mobile app

Available in: **Group Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Conditional formatting uses rulesets, which are collections of rules. Each rule is made up of conditions that determine how and when formatting appears on a field. The conditions can be based on the field's value or on the values of other fields on the page.

In the Lightning App Builder, items that have conditional formatting applied are indicated with an icon ().

To see, edit, or delete any rulesets that have been created for an object, use the Conditional Field Formatting node when viewing the object in Object Manager.

Conditional formatting is designed to amplify the information on the field that is already there. For best results and to ensure that the configuration is accessible, make sure the icon, icon color, and field value

convey the same information, such as a green happy face on a field with a value of `Positive`.

To make conditional formatting accessible for all of your users, fields that use conditional formatting should always have a field value visible.

Because it relies on Dynamic Forms, conditional field formatting can appear on mobile only when Dynamic Forms is enabled for mobile. To make sure that your mobile users can see the conditional formatting you've applied to record fields, turn on **Dynamic Forms and Dynamic Highlights Panel on Mobile** from **Setup | Salesforce Mobile App**.

Supported Fields, Field Types, and Operators

On record pages, conditional formatting rules rely on the data captured in fields associated with the page's object. Not all fields, field types, and operators are supported.

These field types are supported:

- String type fields: Autonumber, Currency, Email, Number, Percent, Phone, Text, Text Area, URL
- ID
- Checkbox (boolean)
- Geolocation
- Picklist
- Formula fields that resolve to one of these preceding types
- Roll-up summary fields that resolve to one of these preceding types

These operators are supported.

- CONTAINS
- = and == (Equal)
- <> or != (Not Equal)
- > (Greater Than)
- >= (Greater Than or Equal)
- < (Less Than)
- <= (Less Than or Equal)

 **Example** For example, you have a Customer Sentiment field on a page with available values of Positive, Neutral, and Negative. You can add conditional formatting to the field with rules configured to show a green happy-face icon when the field value is `Positive`, a grey neutral face when the value is `Neutral`, and a red sad-face icon when the value is `Negative`. That way, viewers can instantly see the sentiment based on the style and color of the icon.

Customer Sentiment ruleset

Rules are executed in the order listed. Rule 1 is evaluated, then Rule 2, and so on. When a rule evaluates to True, the remaining rules are ignored. A ruleset can have up to 10 rules. [Learn More](#)

Format

Icon

Rule 1) Icon (Green Happy Face) IF Customer_Sentiment__c Equal Positive [Edit](#)

Rule 2) Icon (Gray Neutral Face) IF Customer_Sentiment__c Equal Neutral [Edit](#)

Rule 3) Icon (Red Sad Face) IF Customer_Sentiment__c Equal Negative [Edit](#)

[+ Add Rule](#)

[Cancel](#) [Save Ruleset](#)

See Also

[Break Up Your Record Details with Dynamic Forms](#)

Add Conditional Formatting to Record Fields in Lightning App Builder

Give fields on Dynamic Forms-enabled pages custom icons and colors that can appear, disappear, and change color based on the rules you create.

REQUIRED EDITIONS

Lightning App Builder available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Lightning pages and Dynamic Forms available in: Lightning Experience and the Salesforce mobile app

Available in: **Group Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create and save Lightning pages in the
Lightning App Builder:

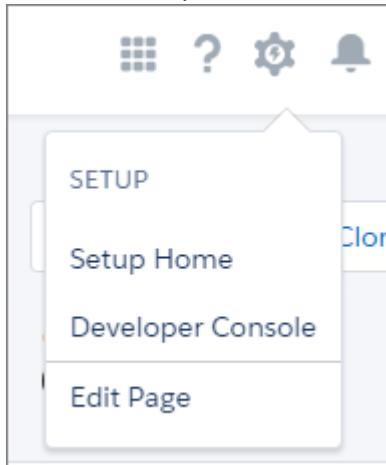
Customize Application

Conditional formatting is made up of rulesets, which are collections of rules. Each rule is made up of sets

of conditions. For example, if you have a picklist field on a record with five available values, you could create a ruleset that applies a star icon to the field. Then you can configure rules and conditions to apply a different color of the star icon for each of the five field values.

1. Open a Dynamic Forms-enabled object record page in one of these ways.

- From the Setup menu on a record page, select **Edit Page**.



When you select **Edit Page** for the first time, Salesforce makes a copy of the standard page. You edit this copy in the Lightning App Builder. If a customized page exists and is active, selecting Edit Page opens that page to edit.

- From the Object Manager in Setup, open an object, then select **Lightning Record Pages**.
- Open a record page from the Lightning App Builder list page in Setup. To find it, in the Quick Find box, enter *App Builder*, and then select **Lightning App Builder**.

2. Click a record field on the Lightning App Builder canvas.

3. In the field's property pane, click in the Conditional Formatting field and either select an existing ruleset or, to create a new one, select **Create Ruleset**.

If you apply an existing ruleset, you can stop here. If you're creating a ruleset, continue with the steps.

4. Give your ruleset a name, then click **Next**.

5. To create a rule, select the icon to show on the field, set its color, and configure the conditions for the icon to appear.

You can add up to 10 conditions to a rule. If you don't specify any conditions, the rule always applies.

Conditional formatting is designed to amplify the information on the field that is already there. For best results and to ensure that the configuration is accessible, make sure the icon, icon color, and field value convey the same information, such as a green happy face on a field with a value of **Positive**.

6. Click **Next**.

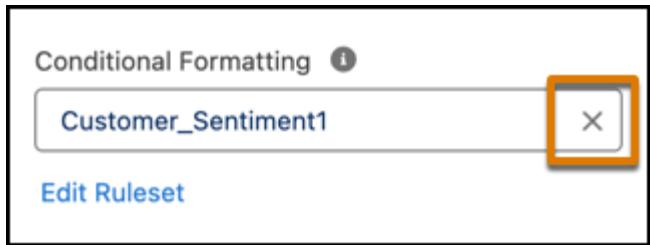
7. Review the ruleset and the rules it contains.

8. Optionally, edit existing rules or add new ones to the ruleset.

9. When the ruleset contains the rules you want, click **Save Ruleset**.

10. Save the page.

To remove formatting from a field, click the X in the Conditional Formatting property to remove the ruleset assignment.



To see, edit, or delete rulesets that have been created for an object, use the Conditional Field Formatting node when viewing the object in Object Manager.

See Also

[Conditional Field Formatting in Lightning App Builder](#)

[Considerations for Conditional Field Formatting in Lightning App Builder](#)

Considerations for Conditional Field Formatting in Lightning App Builder

Keep these considerations in mind when configuring conditional formatting for fields on Dynamic Forms-enabled record pages in the Lightning App Builder.

REQUIRED EDITIONS

Lightning App Builder available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Lightning pages and Dynamic Forms available in: Lightning Experience and the Salesforce mobile app

Available in: **Group Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Considerations

- You can apply conditional formatting only to fields on objects that support Dynamic Forms. Dynamic Forms is supported for most but not all standard LWC-enabled objects. See [LWC Migration for Record Home Pages](#) for a list of LWC-enabled objects. If you open a record page for an object in the Lightning App Builder and don't see a Fields tab in the component panel, then Dynamic Forms and conditional formatting aren't supported for that object. For example, the Note object doesn't support Dynamic Forms because it has a fixed layout. In addition, the Campaigns, Products, and Tasks objects, which aren't LWC-enabled, use information from page layouts and therefore don't support Dynamic Forms or conditional formatting.
- The icons available for conditional formatting come from the [Lightning Design System](#). If you don't see a specific icon in the icon picker, then it's not supported for conditional field formatting.
- If you apply conditional formatting to a field, but it doesn't appear when the page displays at run time, then conditional formatting isn't supported for that field.
- Conditional formatting rules can evaluate differently at run time, depending on the permissions of the user viewing the record page.

For example, let's say that you have fields A, B, C, and D, and a user only has permission to see fields A and B. The user goes to a page that contains field A, which has conditional formatting configured for it with two rules. The first rule has conditions based on fields C and D, which the user doesn't have permission to view. The second rule has conditions based on field B, which the user does have permission to view.

Because the user doesn't have permissions for fields C and D, the first rule could evaluate to `true` or `false`, but the user can't know the outcome because they can't see those fields. However, because they can see field B, they can see the outcome of the conditions of the second rule. In this case, Salesforce ignores the outcome of the first rule and delivers conditional formatting to the user based only on the outcome of the second rule, because the conditions for that rule are based on a field they have permission to view field B.

Therefore, it's possible that the user in this example sees a different result for conditional formatting than their colleagues who have access to fields C and D.

- Conditional formatting rules evaluate based on the raw value of a number, not its rounded up display value.
- Conditional field formatting rules treat blank (null) values on numerical fields as zero values.
- Conditional formatting rulesets aren't automatically included when you add a Lightning page to a change set. If you create an outbound change set with a Lightning page that contains conditional formatting, make sure to add all of the page's conditional formatting rulesets to the change set. Conditional formatting rulesets are listed as "UI Format Specification Set" in the change set Component Type dropdown.

Limitations

- A Lightning page can have up to 15 unique rulesets.
- A ruleset can have up to 10 rules.
- A rule can have up to 10 conditions. If you don't specify any conditions, the rule always applies.
- You can't add conditional formatting to [cross-object fields](#), nor can you select cross-object fields when creating rules.

See Also

[Conditional Field Formatting in Lightning App Builder](#)

[Add Conditional Formatting to Record Fields in Lightning App Builder](#)

Dynamic Interactions in the Lightning App Builder

Dynamic Interactions is part of our continuing drive to make Lightning pages more dynamic and interactive. With Dynamic Interactions, an event occurring in one component on a Lightning page, such as the user clicking an item in a list view, can update other components on the page. Dynamic Interactions lets admins create applications with components that communicate and transform based on user interactions, all in the Lightning App Builder UI. It unlocks capabilities that previously were reserved only for developers.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

To get the most out of Dynamic Interactions, admins and developers work together.

Developers write custom components that power the Dynamic Interactions. The developer defines the events that the component supports and then exposes the events in the Lightning App Builder UI. Then, in the Lightning App Builder, an admin configures the event by setting up the interactions between the components.

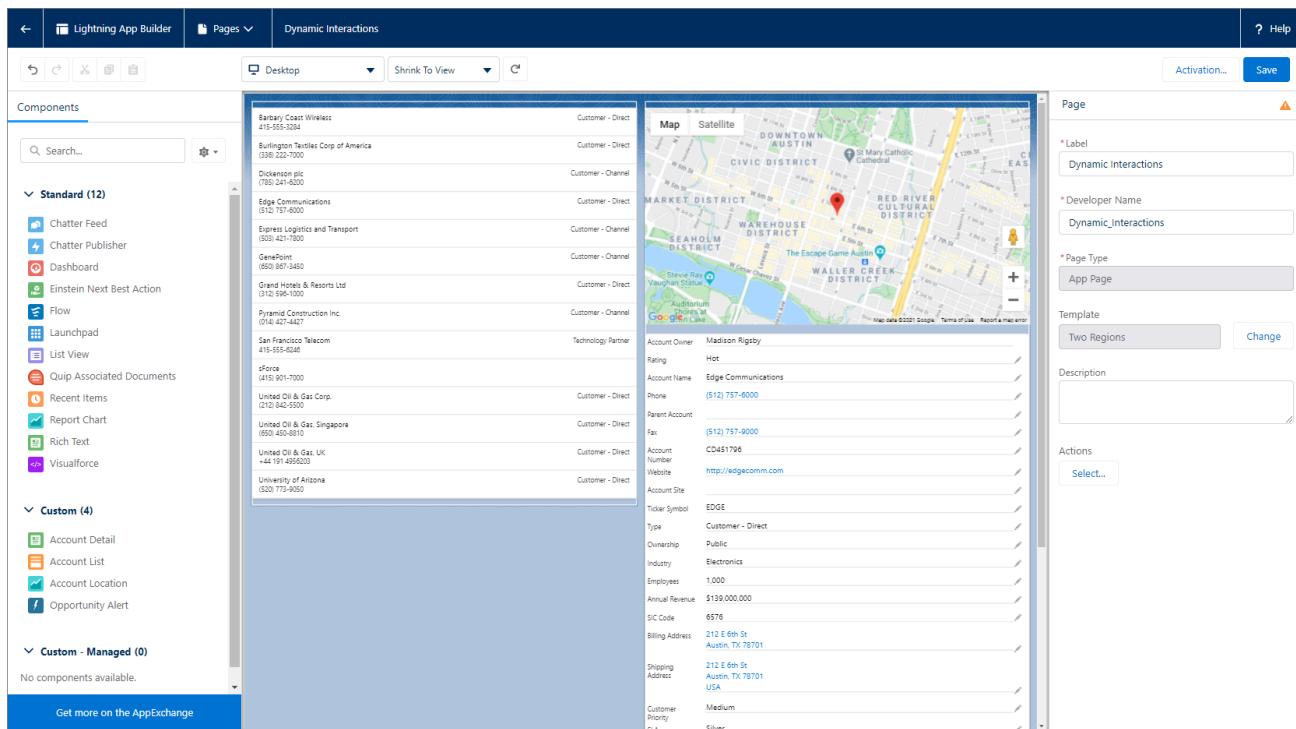
Dynamic Interactions has four major building blocks.

- Event—Anything that can trigger an interaction, such as a mouse click, a button press, or a change in a field’s value.
- Interaction—An activity that happens between the source and the target.
- Source—The item triggering the event. Currently, only custom Lightning web components and the [Dynamic Actions Bar component \(Pilot\)](#) are supported as sources.
- Target—The item that’s the target of the interaction. Any component on a Lightning page can be a target.

An event occurring in one component (the source) can trigger changes in one or more other components on the page (the targets). A single source can have multiple targets.

 **Note** Dynamic Interactions is supported only on app pages, and interactions are limited to activity between components.

 **Example** Kai (they/them) and Rina (she/her) are an admin and developer team. Kai wants to give their on-the-go users an easy way to check the location of various accounts so that they can plan efficient site visits using a simple app page. Kai enlists Rina’s help to make this happen. As a developer, Rina knows that she can wire up events between components using code. But she wants to give Kai the power to configure the event interactions in the way that they need to without having to come back to her every time a change is necessary. Rina creates a custom source component for Kai, plus two other components to act as event targets. After installing the new components in their org, Kai has an app page with three custom components: Account List, Account Detail, and Account Location (a map).



The source component is the Account List on the left. It has an Item Selected event enabled, which Rina exposed to the Lightning App Builder UI using new Dynamic Interactions code. After Kai finishes configuring the event interactions, when a user clicks a list item in the Account List component, the event fires. The Account Detail component updates to show that account's details. At the same time, the Account Location component pinpoints that account's location on a map. Every new click or tap on an account in the list results in updating the content in the other two components.

See Also

- [Lightning Web Components Dev Guide: Configure a Component for Dynamic Interactions in the Lightning App Builder](#)
- [Lightning Web Components Dev Guide: XML Configuration File Elements](#)

Configure Interactions in the Lightning App Builder

After you have a custom source component with exposed events in your org, you can assign event targets and configure the event interactions in the Lightning App Builder.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create and save Lightning pages in the Lightning App Builder:

Customize Application

1. From Setup, in the Quick Find box, enter *App Builder*, then select **Lightning App Builder**.
2. Edit an existing app page, or click **New** to create one.
3. In the Lightning App Builder, add a custom source component to your page.
If you don't have target components on your page, add those too.
4. Click the source component on the canvas.
5. In the properties pane, click the **Interactions** tab.
6. Under the desired event, click **Add Interaction**.
The properties pane changes to show you the interaction details.
7. Select an interaction.
Currently, the only interaction available is **Update Properties**.
8. Select the target component for the interaction.
Components are listed with their region location to help you select the correct component when you work on pages with components spread across multiple regions.



9. Configure the target component properties.

When you select a target component, its properties appear in the Interaction Details pane. There you define the value that you want each target property to have when the interaction happens. If you leave a target property blank, its value remains unchanged when the event triggers.

To use an expression to define a target property value dynamically, click .

Checkbox properties have a **No change** option. If you've changed the value of a checkbox property, use this option to reset it back to its original value, and to indicate that you want its value to remain unchanged when the event triggers.

10. Save the page.

See Also

- [Dynamic Interactions in the Lightning App Builder](#)
- [Dynamic Interactions Considerations](#)

Expressions in Dynamic Interactions Target Properties

An expression is a small chunk of code that can be evaluated to a value. It can be as literal and simple as `1+1` or a more complex combination of variables, operators, and functions. One expression is supported for Dynamic Interactions in the Lightning App Builder: `{ !Event.eventPropertyName }`. You can use this expression when setting target component property values in the Lightning App Builder UI and in Metadata API. The value of the `eventPropertyName` part of the expression varies based on which properties the developer makes available for the event in the source component.

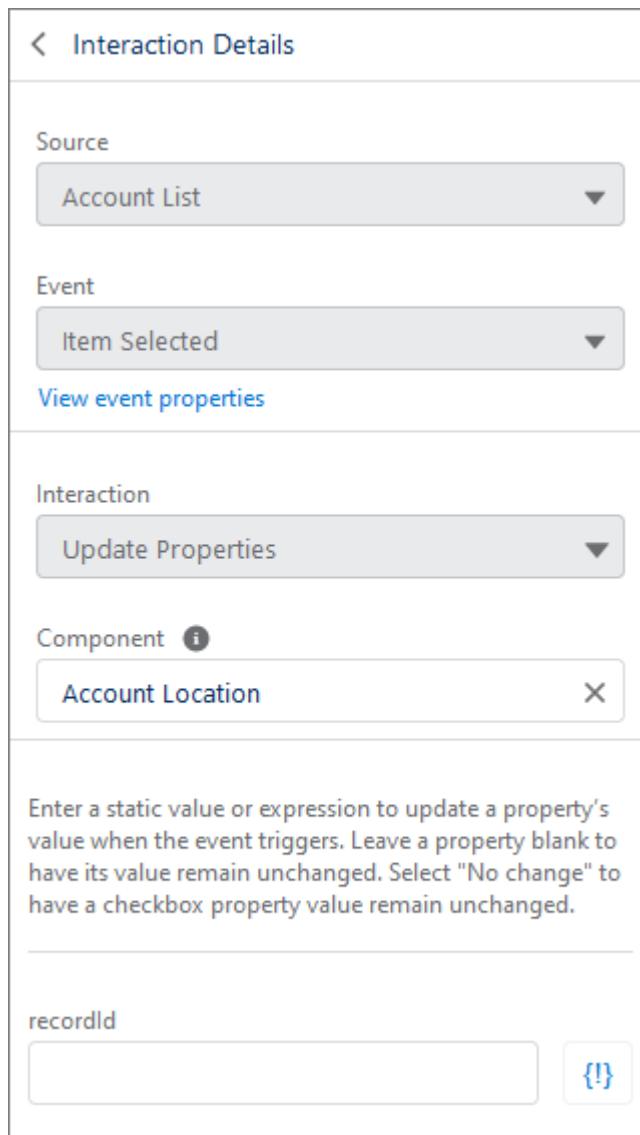
REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

 **Note** If you see the  button on a target property, you can set its value with an expression.

With an expression, you can pass a value, such as a record ID or an API name, to the target item. Then when the event fires, the target item can use that value. Using an expression to define a property value makes the event interaction dynamic. We can illustrate this with an example of an app page with custom Account List and Account Location components. Here, an admin is configuring an interaction between the Account List source component and the Account Location target component.



The screenshot shows the 'Interaction Details' configuration screen. It includes sections for Source (Account List), Event (Item Selected), Interaction (Update Properties), Component (Account Location), and a text area for entering an expression. The expression field contains 'recordId' and has an expression icon ({}).

Interaction Details

Source: Account List

Event: Item Selected

View event properties

Interaction: Update Properties

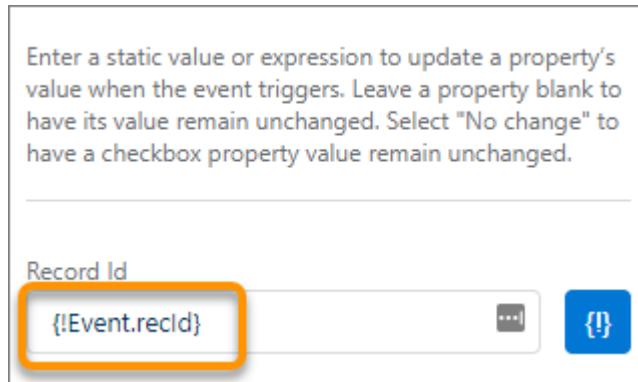
Component: Account Location

Enter a static value or expression to update a property's value when the event triggers. Leave a property blank to have its value remain unchanged. Select "No change" to have a checkbox property value remain unchanged.

recordId 

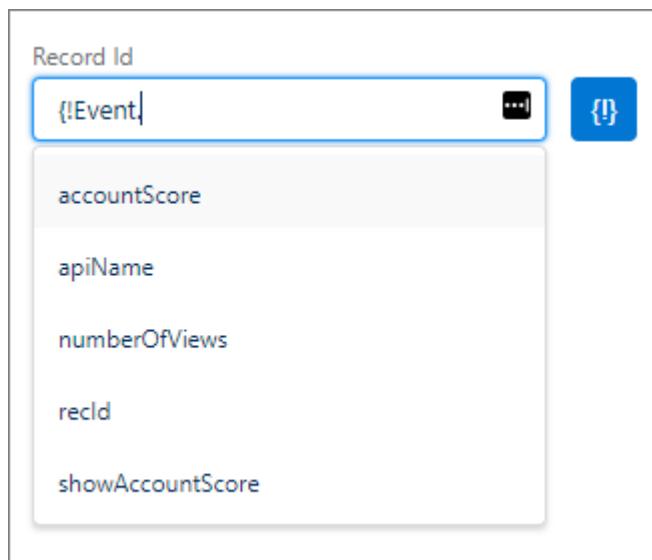
The goal for the interaction in this example is that when a user clicks an account in the list, the Account Location component updates to show the selected account's location as a point on a map. But the Account Location component needs to know the record ID of the selected account to do that. Passing the record ID into the Account Location component tells the component which record to look at to pull the address information. In this case, the developer made the `recId` property available as part of the

event, so the target value can be set to the active list item's record ID using the `{!Event.recId}` expression.



Why not enter a 15- or 18-digit record ID into the field? If you do, any item clicked in the Account List always resolves to that one record ID, which isn't dynamic and is incorrect.

If the developer makes more than one target property available, you can set the property to be passed in by starting to type the expression in the field. After the period, you can use autocomplete to select which property to use in the expression. But don't forget to add the ending bracket `}` after you make your selection.



 **Note** Expression autocomplete doesn't work when:

- Using expressions in the rich text editor
- Typing an expression in a text field without first clicking the expression button ()

See Also

[Configure Interactions in the Lightning App Builder](#)

[Dynamic Interactions Limits and Limitations](#)

[Dynamic Interactions in the Lightning App Builder](#)

Dynamic Interactions Considerations

Keep these considerations in mind when working with Dynamic Interactions.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

- Informational components within target components are ignored when the target component's properties are shown in the Event property editor.
- If you switch page templates for a page that contains Dynamic Interactions, the available template list shows only templates that support Dynamic Interactions.
- Aura components re-render when an interaction targeting them is triggered.
- Autocomplete doesn't work when entering an expression in the rich text editor.
- After a component's event metadata is used on a Lightning page or released as part of a managed package, certain breaking changes aren't allowed, such as removing the event, renaming a property, or changing a property type.

Dynamic Interactions Limits and Limitations

Keep these considerations in mind when working with Dynamic Interactions in the Lightning App Builder.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

- Dynamic Interactions is supported only on app pages.
- Only LWC custom components can be source components, but any component present on the page (Aura or LWC) can be a target.
- Dynamic Interactions isn't supported on pages based on custom page templates.
- Only String and Rich Text type properties can use expressions to define their values.
- The "required" property restriction isn't respected when defining a new value for a target property in Dynamic Interactions, regardless of its type or whether it's defined with an expression.
- `Event` is the only context supported for expressions in interactions.
- You can use expressions only for properties of type String, Integer, and Boolean.
- You can't set a target property value as an array or list of values, such as a multi-select picklist.
- You can set a target property value of a String attribute to `empty` using Metadata API but not in the Lightning App Builder UI.

- Dynamic Interactions doesn't work in the Mobile Only app in the Salesforce mobile app or in the legacy tablet mobile experience.
- When a dependent property is autopopulated with a value based on a selection you made or a value you entered in another property, the autopopulated value isn't saved unless you "touch" the dependent property field by clicking into it or tabbing to it.

Guidance for App Builder

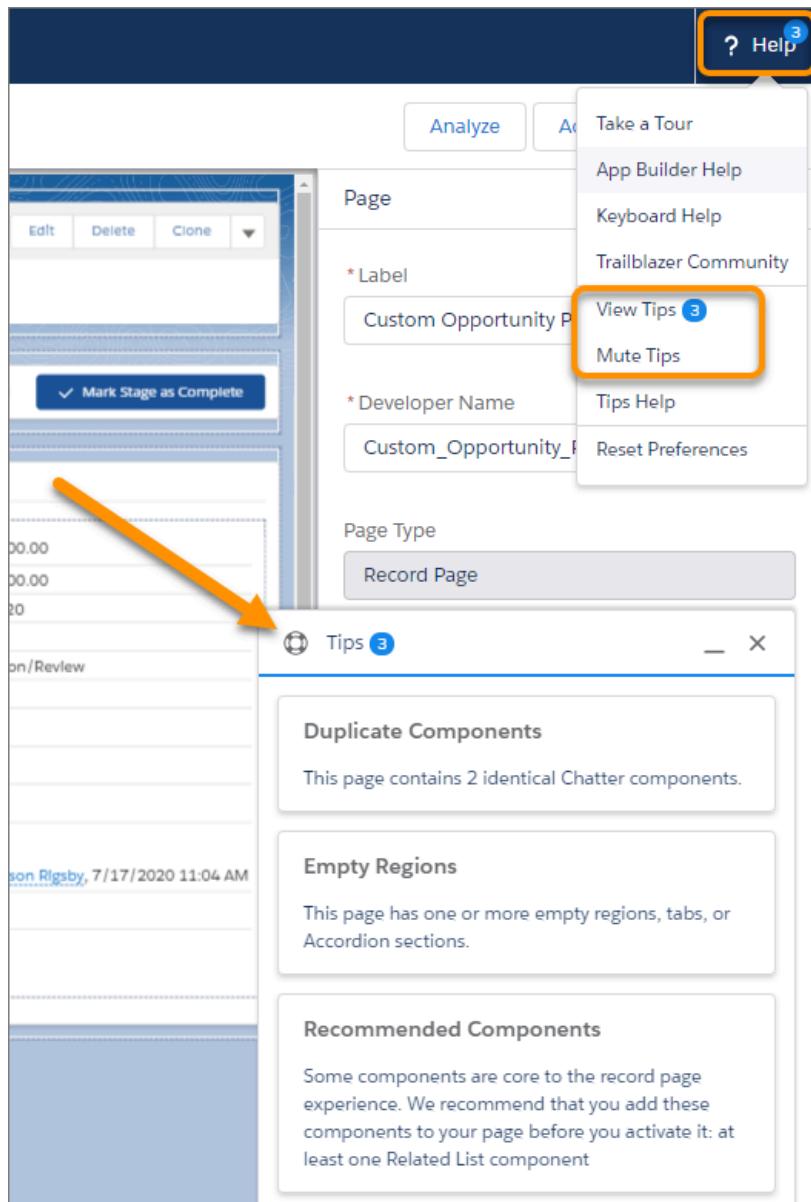
Get suggestions for improving your Lightning pages just when you need them. Guidance for App Builder gives you feedback for enhancing your Lightning pages right in the design phase.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Tips are available for categories such as performance, usability, and structural issues. Some issues that the tips cover are also captured in the Salesforce Optimizer report. For example, if two components on a page are identical, you see a duplicate components tip. When you fix the issue, the tip disappears.



If tips are available for your page, you see an indicator icon on the Help menu.

To hide tips while designing your page, go to the Help menu and select **Mute Tips**. You can reopen the tips prompt by selecting **View Tips**.

See Also

[Lightning App Builder Considerations](#)

[Salesforce Help: Improve Your Implementation with Salesforce Optimizer](#)

Lightning Page Performance

Several factors can negatively affect your Lightning page's performance at runtime. For example, your users can run into page performance issues if they aren't using the latest version of a Salesforce recommended browser to view the page. Some factors you can control, and others you can't. Here are

some ways you can improve performance while configuring your page in the Lightning App Builder.

REQUIRED EDITIONS

Lightning App Builder available in: both Salesforce Classic and Lightning Experience

Lightning Home and utility bar pages available in: Lightning Experience

Lightning app and record pages available in: both the Salesforce mobile app and Lightning Experience

Email application pane pages available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Consider the number, type, and placement of components on the page. These design-related issues can cause a page to load slowly:

- Too many components visible on page load
- Too many heavy-impact related lists visible, especially those lists with multiple-object relationships
- Too many fields in your Record Detail component
- Having the News or Twitter component visible

If your page falls into one or more of these categories, we recommend that you move some components into a non-default tab or Accordion component section, unless you're designing the page for blind or low-vision users. For the heavy-impact related lists, we recommend that you move the related lists lower on their respective page layouts so that they aren't part of the initial page load.

If your Record Detail component has many fields, we recommend that you reduce the fields on your page layout to fewer than 60. Alternatively, move the Record Detail component into a non-default tab or Accordion section so that it's not part of the initial page load.

For pages that are viewed on a phone, we recommend a maximum of eight visible components per page. If a page has more than eight, put some in a tab or Accordion section or hide them for mobile with a visibility filter.

You can see how your pages are performing in the Lightning Usage App from the App Launcher. It shows you the most used pages in your org and their page load time by browser or by page.

Performance Analysis for App Builder automatically runs when you build a page. If your page performance is poor or moderate, recommendations to improve performance appear. You can manually check a Lightning record page's runtime performance at any time by clicking **Analyze** from the Lightning App Builder toolbar. It assesses page performance based on all visible standard and custom components on the page. Components in nondefault tabs and Accordion sections aren't analyzed.

The User Metrics card provides 90 days of your org's user data such as browser speeds, network latency, and number of cores. Use this information to help you decide which Performance Analysis for App Builder recommendations to take.

-  **Note** Salesforce measures performance in Experienced Page Time (EPT). The page load time mentioned here and in the Performance Analysis for App Builder tool is EPT.

Considerations

- If a page has more than 5 related lists or more than 25 fields in the record detail, users can encounter performance issues when viewing the page in the Salesforce mobile app.
- In the Performance Analysis for App Builder tool:
 - Components that are restricted to the desktop form factor via a visibility rule aren't included in the phone form factor performance assessment.
 - Analysis of page performance on a phone is available only on pages whose template supports the phone form factor.
 - Components aren't the only factors influencing page load time, so the numbers in the component impact chart don't add up to 100% of the predicted page load time.
 - Analysis of pages for the desktop form factor is measured in seconds. For the phone form factor, the page is scored based on the components that are visible when the page loads on a phone.
 - User metrics are org-specific or page-specific data. Previously activated pages display metrics from visits to that page. New pages display metrics from visits to all org pages. Metrics displayed in a sandbox can differ from metrics in production.

See Also

- [Technical Requirements and Performance Best Practices](#)
- [Measure Performance for Your Salesforce Org](#)
- [Gain Insight About Page Performance with the Lightning Usage App](#)
- [Get Lightning Experience Adoption Insights with the Lightning Usage App](#)
- [Slowest Desktop Record Pages](#)

Gain Insight About Page Performance with the Lightning Usage App

See how your Lightning pages are performing and understand the data in the Lightning Usage App. The Lightning Usage App shows you the most used pages in your org and their page load time by browser or by page.

REQUIRED EDITIONS

To open the Lightning Usage App, from the App Launcher, select **Lightning Usage App**.

Lightning App Builder available in: both Salesforce Classic and Lightning Experience

Lightning Home and utility bar pages available in: Lightning Experience

Lightning app and record pages available in: both the Salesforce mobile app and Lightning Experience

Email application pane pages available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Page Names in the Lightning Usage App

In the Lightning Usage App, under Page, the tables list Lightning pages by page name. Use this glossary to understand some of the page names for page types in your org.

-  **Note** Salesforce groups similar page types for the same object together in the Lightning Usage App. For example, the Opportunity object has two Lightning record pages. Results for both pages are grouped as Opportunity Record in the Lightning Usage tables.

Name	Example Page Name	Page Type
Record	Account Record	An object's record home page.
Record List	Account Record List	An object list view in a standard app or a Lightning Console App. Record List can also refer to a Task list on the Task home page.
Related List	Account Related List	An expanded related list page for an object. To view the expanded related list, users click View All from the related list on a record page.
Search	Account Search	Search results for a specific object.
one:recordActionWrapper	Account one:recordActionWrapper	The window that appears when users select an action for an object.

Page Performance for Lightning Console Apps

Salesforce measures page performance using Experienced Page Time (EPT). EPT is a metric that calculates how long it takes for a page to load into a state that a user can meaningfully interact with. In the Lightning Usage App, performance for Lightning pages in Salesforce Console apps can differ from Lightning pages in standard apps.

Salesforce Console apps let you open multiple records at a time, with related records opened in subtabs under the original record. If a user opens record tabs in a Salesforce Console app, the page reloads more quickly each time the user navigates back to an open tab. As a result, the average EPT for that page

appears to be lower in the Lightning Usage App. If a page loads in less than 150 milliseconds, it's filtered out of the page performance results in the Lightning Usage App.

Considerations for Lightning Usage App Tables

Keep these considerations in mind when viewing tables in the Lightning Usage App.

- Monthly tables show results for the last complete month rather than the last 30 days. For example, if you view a monthly report on May 20, you see results from April 1 through April 30. On June 1, you can see results from May.
- Daily results in the Lightning Usage App are sometimes outdated by several days. Each day, the Lightning Usage App runs jobs on usage data from the previous day. Results from that day appear in the Lightning Usage Apps tables only after the jobs complete.

Slowest Desktop Record Pages

Use the Slowest Desktop Record Pages table in the Lightning Usage App to see which desktop record pages could use some tweaking to run faster. It lists the desktop record pages with a median loading time, called Experienced Page Time (EPT), of 4 seconds or longer.

REQUIRED EDITIONS

Lightning App Builder available in: both Salesforce Classic and Lightning Experience

Lightning Home and utility bar pages available in: Lightning Experience

Lightning app and record pages available in: both the Salesforce mobile app and Lightning Experience

Email application pane pages available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

The table lists the page name (1), developer name (2), and number of views (3). Click **View** (4) to open the record page in Lightning App Builder and view the Record Page Performance Analysis results.



See Also

[Technical Requirements and Performance Best Practices](#)

[Measure Performance for Your Salesforce Org](#)

[Lightning Page Performance](#)

[Gain Insight About Page Performance with the Lightning Usage App](#)

Lightning App Builder Considerations

Keep these considerations in mind when working with Lightning pages and Lightning apps in the Lightning App Builder.

REQUIRED EDITIONS

Lightning App Builder available in: both Salesforce Classic and Lightning Experience

Lightning Home and utility bar pages available in: Lightning Experience

Lightning app and record pages available in: both the Salesforce mobile app and Lightning Experience

Email application pane pages available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Tips

- To reduce user confusion, give each tab and Accordion section a unique label.
- If you're curious about a component on the canvas, click it to see more information in the properties pane.
- Put the Activities component in a place where no other components are beneath it.
- When editing navigation items in a Lightning app, consider how many items you include. Users can't remove the items that you include in the navigation bar, and they can't personalize the navigation bar when it contains more than 50 items. For example, if you include 32 items in an app's navigation bar, users can add 18 more personal items. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add.
- We don't recommend putting record fields into a narrow region on the right-hand side of a page. On pages that support Agentforce, the Einstein panel covers up content on the right-hand side of the page. If record fields are behind the Einstein panel, users who use Einstein for field generation can't see the updated field information.

General

- The Lightning App Builder supports the same browsers as Lightning Experience and isn't supported on mobile browsers. The minimum recommended resolution for the Lightning App Builder is 1280x1024.
- When viewing a page in the Lightning App Builder:
 - The palette shows only the components that are available for the devices supported by the page template. For example, if you're working on a page whose template supports only desktop, the component palette contains components that support desktop, whether they support desktop only or both desktop and phone.
 - The palette shows only the components that are available for the object tied to that page. For person account pages, some of the components that you see apply only to business accounts.

- The form factor switcher shows only the devices that the page template supports. If the canvas view is switched to a form factor that a component on the page doesn't support, that component displays as a placeholder with a warning.
- User interface controls that appear inside components on the canvas, such as buttons or links, are there to help you design and configure the component, but they aren't functional. For example, the quick action buttons inside a Highlights Panel component aren't clickable. Screen readers and assistive technologies announce the roles of these interface elements, but they don't function at design time.
- When a component is viewed on a device that the component doesn't support, it's dropped from the page.
- You can create and edit a record page even if you don't have permission to access the object that the page is associated with. You can add, remove, delete, and reconfigure components on the page, but you don't see any of the content for the components that are based on that object.
- In Salesforce Classic, Lightning page tabs don't display on the All Tabs page when you click . Lightning page tabs also don't appear in the Available Tabs list when you customize the tabs for your apps.
- On Lightning Experience record pages, a Record Detail component that contains more than four external lookup fields breaks the page at runtime.
- When you click a compound address field from a Record Detail component, it opens in Google Maps™ with just the address populated. When you click a compound address field from a Dynamic Forms-enabled page, it opens in Google Maps showing not only the address, but also the latitude and longitude values.
- When you enter a custom label for a tab, that label text is shown in title case at runtime. The only exception is for Location object pages. A custom tab label that appears on a Location object page isn't changed to title case. It appears to users exactly as it was entered.
- Changes you make to a Lightning page can take up to 15 minutes to be visible, especially when using Salesforce in multiple browser tabs. For example, let's say you have an Account record page open in one tab and Setup open in a second tab. If you open and make changes to the same record page in the second tab, such as adding a field by using the Add Field wizard or changing components in the Lightning App Builder, those changes don't appear right away when you return to the first browser tab. There are a few ways to resolve this issue.
 - Log out and log back in to Salesforce.
 - Wait for 15 minutes until the page cache expires, and then refresh the page.
 - Click **Edit Page** from the Setup menu on the record page to open it in Lightning App Builder, and then click **Save** without making any changes.
- When you select the Insert Component button on the canvas () , focus shifts to the component panel. At the same time, the "Select a component" element is activated. After you select a component to insert, the "Select a component" element disappears, and focus shifts to the inserted component. This is the intended behavior.

The "Select a component" element is a placeholder element, not a button. It's not focusable, and can't be tabbed to or selected.

Page Templates

- The Three Regions template and the pinned region templates are designed with Lightning console apps in mind. They feature a main region and two sidebars with fixed proportional widths. The main

region is 50%, and the side region widths are each 25%. Three-region templates require more screen width to display correctly. Three-region templates can display incorrectly on certain devices or monitors with low resolutions.

- When switching page templates, if a region in the template that you're switching to has more than 25 components mapped to it, all components after the 25th are dropped from the page. You can proceed with the template swap, but you must manually add the dropped components from the palette and reconfigure their properties.
- The console pinned region templates let users view and work with records while navigating subtabs in console apps. These templates also work in apps with standard navigation. However, we recommend that you use a non-pinned region template in standard apps, because those apps don't benefit from a pinned region. When working with pinned region templates, keep these considerations in mind.
 - The templates are available only for record pages.
 - Pinned regions don't support theming. For example, if you use custom theming to brand your app with the color green, the pinned region doesn't apply the green color.

Page Activation and Assignment

- If a device isn't on the options list when you try to assign a Lightning page as the org default, the page template doesn't support it.
- You can see which record pages are activated for which Lightning app and which form factor in the Lightning Record Pages related list in Object Manager.
- If you no longer want a page to be an app or org default, redo the activation process for the page, and select the option to remove it as the default.
- If you activate a page and then return to make changes, you don't have to activate it again. Clicking **Save** after you make your edits pushes the changes to your users.
- If you no longer want a page assigned to a particular form factor, redo the activation process, and select the option to remove it.
- If you activate a page for only one form factor, you can add support for another form factor later as long as the page's template supports it.
- Changes that you make to Lightning record page assignments aren't immediately reflected in the Salesforce mobile app. To see a newly assigned record page, close and restart the Salesforce mobile app.
- If there are no custom app-level page assignments set for the Service Console or Lightning Sales Console apps, users viewing those apps in the Salesforce mobile app see the org default record page assigned to the object instead of the system app default record page.
- If you assign a custom record page as the org-wide default for an object, it becomes the default page for that object across all your Lightning apps. It also appears as the default record page for that object in Classic apps when you open them in Lightning Experience.
- Accounts and person accounts have different standard default pages to begin with. However, when you create a Lightning page for accounts and assign it as the default for an org or an app, that page becomes the org or app default for business accounts and person accounts. To display a custom record page for person accounts, create a custom account record page, then assign it to the person account record type.
- An activated Lightning record page takes precedence over the Lightning Experience Override setting for the View action on the object.

Pages and Packages

- If you package a Lightning page that references protected labels, then if a subscriber org clones that page, the protected labels cause the components that use them on the cloned page to be invalid.
- A Lightning page installed from a managed package appears in the Lightning pages list with a Clone option rather than Edit or Delete. The Edit and Delete buttons are also replaced with Clone on the installed page's detail page.
- You must have the Manage Prompt Templates permission to successfully package Lightning pages that reference prompt templates. Without this permission, package creation succeeds, but the prompt template isn't included in the package.

See Also

[Create and Configure Lightning Experience Record Pages](#)

[Lightning App Builder Limits and Limitations](#)

Considerations and Limitations for Flows in Lightning Pages

Here are some things to keep in mind when you add a flow component to a Lightning page.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Lightning pages always use Lightning runtime, so also review [Limitations of Lightning Runtime for Flows](#).

- Running Flows from a Lightning Page—When a user opens a Lightning page that has a flow component, the flow runs when the page loads. Make sure that the flow doesn't perform any actions, such as create or delete records, before the first screen.
- Input Variable Limitations—These variables aren't supported.
 - Collection variables
 - Record variables
 - Record collection variables
- The component supports only manually entered values for input variables.
- Text input variables accept a maximum length of 4,000 characters.
- Deployment Considerations—Change sets and the Metadata API deploy all flows as inactive, which users can't run. If you deploy a Lightning page (known as FlexiPage in the API) that contains a flow component, make sure to activate the flow.

Lightning App Builder Limits and Limitations

Keep these limits and limitations in mind when working with Lightning pages and Lightning apps in the Lightning App Builder.

REQUIRED EDITIONS

Lightning App Builder available in: both Salesforce Classic and Lightning Experience

Lightning Home and utility bar pages available in: Lightning Experience

Lightning app and record pages available in: both the Salesforce mobile app and Lightning Experience

Email application pane pages available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

- In Salesforce Classic, the Default On and Default Off options for Lightning page tabs don't work the same way as for other custom tabs. The Lightning page menu item appears for the selected profiles in Salesforce for Android, Salesforce for iOS, and Salesforce mobile web whether you choose Default On or Default Off. To hide the Lightning page for the selected profiles, select the **Tab Hidden** option.
- You can place up to 100 tabs in a Tabs component.
- A Lightning page region can contain up to 100 components.
- The columns inside a Field Section are considered components. A Field Section component with two columns counts as three components when determining the total components on a page.
- Each tab in a Tabs component and each section inside an Accordion component counts against the limit for total components on a page. For example, a Tabs component containing three tabs counts as four components.
- Lightning component global actions aren't supported for app pages.
- Page overrides by profile aren't supported in packaging. They're supported in change sets, but must be added manually.
- Dropdown menus in the Lightning App Builder can display up to 200 items. Enter a few characters, and all available matches are displayed as you type.
- Although you can add the Potential Duplicates component to person account pages, the component doesn't display duplicate person accounts.
- Custom tab labels in the Tabs component—including those labels installed from packages—aren't translated. For example, if you create a custom Goals tab in English, then view the page as a user whose language is set to French, the tab still displays as Goals. However, you can use the `{!$Label.customLabelName}` expression in a component label or attribute to represent a custom label that you create in Setup by using the custom label feature. For more information, see “[Build Localized Component Labels and Attribute Values on Lightning Pages with Custom Labels](#).”
- Right-to-left (RTL) language layout isn't supported in the Lightning App Builder.

See Also

[Lightning App Builder Considerations](#)

Keyboard Navigation and Shortcuts in the Lightning App Builder

The Lightning App Builder supports some unique keyboard combinations to help take actions inside the tool and navigate from element to element.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

These shortcuts and navigation tips are also available in the Lightning App Builder. Open the Help menu and select **Keyboard Help**.

Global Shortcuts

Action	Shortcut
Keyboard help	? (Shift + /)
Switch panel focus	F6
Switch focus to the docked prompt	Ctrl+F6 (Windows) or Cmd+F6 (macOS)
New page	n
Open page	o
Save page	Ctrl+s (Windows) or Cmd+s (macOS)
Undo	Ctrl+z (Windows) or Cmd+z (macOS)
Redo	Ctrl+Shift+z (Windows) or Cmd+Shift+z (macOS)

Canvas Navigation

Action	Navigation
Highlight component	Down Arrow or Up Arrow
Select the highlighted component	<p>Press Enter.</p> <p>To move focus to the properties pane, press Tab several times.</p>
Insert a component from the component pane before or after another component	<ol style="list-style-type: none"> 1. Highlight a component on the canvas. 2. Press Tab several times to highlight an Add Component icon, and then press Enter. 3. Press F6 until the focus is on the Components tab in the component pane.

Action	Navigation
	<ol style="list-style-type: none"> 4. Press Tab until the focus is on the component pane. 5. Use arrow keys to navigate to the component that you want to add, and then press Enter.
Add a component to empty region	<ol style="list-style-type: none"> 1. Tab to move focus to the empty region. 2. Press Enter to activate the insertion point. 3. Press F6 until the focus is on the Components tab in the component pane. 4. Press Tab until the focus is on the component pane. 5. Use arrow keys to navigate to the component that you want to add, and then press Enter.
Move a component	<ol style="list-style-type: none"> 1. Select the component that you want to move. 2. Cut the selected component by pressing Ctrl+x (Windows) or Cmd+x (macOS). 3. Press Tab to highlight another component on the canvas, and then press Enter to select it. 4. To insert the cut component before the selected one, press Tab several times to highlight the Add Before icon, press Enter to activate the icon, and then press Ctrl+v (Windows) or Cmd+v (macOS) to paste. 5. To insert the cut component after the selected one, press Tab several times to highlight the Add After icon, press Enter to activate the icon, and then press Ctrl+v (Windows) or Cmd+v (macOS) to paste.
Remove a component	<ol style="list-style-type: none"> 1. Select the component that you want to delete. 2. Press Delete. 3. Alternatively, press Tab to highlight the Delete icon () on the selected component, and then press Enter.

Action	Navigation
Access Tab and Accordion component contents	<ol style="list-style-type: none"> 1. Highlight the Tabs or Accordion component. 2. Press Tab several times until the focus is on the first tab or section. 3. Press Up Arrow or Down Arrow to cycle through the tabs or sections. 4. Press Tab to highlight the first component inside the selected tab or section.

 **Note** After using the arrow keys to navigate into Accordion, Tabs, or other container component content, you must press Tab (or Shift+Tab if moving up the canvas) before you can continue to use the arrow keys to navigate between components.

See Also

[Keyboard Shortcuts](#)

Manage Your Notifications with Notification Builder

Keep your users in the know with timely notifications, whether they're at their desks or on the go. Create custom notifications to give your users new information and reminders. Choose whether Salesforce notifications appear on desktop, mobile, Slack, or at all.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To view notification types:	View Setup and Configuration
-----------------------------	------------------------------

To create and edit notification types:	Customize Application
--	-----------------------

[Create and Send Custom Desktop or Mobile Notifications](#)

Create custom notifications that appear on Salesforce desktop or mobile when important events occur. Use a process in Process Builder, Apex code, or the invocable action API to send the notification.

[Custom Slack Notifications](#)

Reach your users on Slack with important notifications.

[Manage Notification Delivery Settings](#)

Modify where your notifications appear and which apps deliver your notifications.

Considerations for Notifications

Learn about the considerations for sending and viewing notifications, such as notification limits.

See Also

[Salesforce Mobile App Notification Types](#)

Create and Send Custom Desktop or Mobile Notifications

Create custom notifications that appear on Salesforce desktop or mobile when important events occur. Use a process in Process Builder, Apex code, or the invocable action API to send the notification.

Create a Desktop or Mobile Notification

Define the custom notification's details.

Add a Mobile Action Group for Actionable Notifications (Beta)

You can now add actions to custom notifications. These notifications appear on a user's screen, depending on their settings, and can be used for a variety of actions, such as approving requests, following up on deals that might be at risk, or closing a case.

Send a Desktop or Mobile Notification From a Process

Add the Send Custom Notification action to a process, then add recipients and content.

Other Ways to Send a Custom Notification

We've made it easy to build notifications using clicks, not code. But if your business needs are more complex, we also support flows and programmatic solutions.

Create a Desktop or Mobile Notification

Define the custom notification's details.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To view notification types: View Setup and Configuration

To create and edit notification types: Customize Application

1. From Setup, in the Quick Find box, enter *Notification Builder*, and then select **Custom Notifications**.
2. Click **New**, and then select **Salesforce Desktop or Mobile**.
3. Add your Custom Notification Name and API Name, and notification channels.

Channel	Description
Desktop	Sends a notification to the desktop notification tray.
Mobile	Sends an in-app and push notification to enabled supported apps.

**Note**

- Mobile in-app notifications require the [Enable in-app notifications setting](#). Mobile push notifications depend on a user's device-level and, if available, app-level push notification settings. Push notifications for the Salesforce mobile app require the [Enable push notifications setting](#)
- If you created a custom notification type with a mobile delivery channel before Winter '20, your notification is automatically delivered to the Salesforce mobile app. If you create a custom notification type with a mobile delivery channel in Winter '20 or later, you must manually enable the Salesforce mobile app and any other supported apps in Notification Delivery Settings.

4. Add a mobile action group if you want to create notifications that include embedded actions. For example, you can create a notification that allows users to approve requests directly from their lock screen.
5. Save your notification type.
6. If you enable the mobile channel, you must enable the supported apps for your notification type.
 - a. From Setup, in the Quick Find box, enter *Notification Builder*, and then select **Notification Delivery Settings**.
 - b. Select your custom notification type, and select **Edit** from the dropdown menu.
 - c. Select the supported applications for your notification type, and save.

Add a Mobile Action Group for Actionable Notifications (Beta)

You can now add actions to custom notifications. These notifications appear on a user's screen, depending on their settings, and can be used for a variety of actions, such as approving requests, following up on deals that might be at risk, or closing a case.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create and edit mobile action groups: Customize Application

To view Apex classes: View All Data



Note Actionable notifications is a pilot or beta service that is subject to the Beta Services Terms at [Agreements - Salesforce.com](#) or a written Unified Pilot Agreement if executed by Customer, and applicable terms in the [Product Terms Directory](#). Use of this pilot or beta service is at the Customer's sole discretion.

1. From Setup, in the Quick Find box, enter *Notification Builder*, and then select **Custom Notifications**.
2. Click on a custom notification.
3. Click on the **Mobile Action Groups** tab and create a new mobile action group, giving it a unique API name.
4. Click **New Action** to add up to three actions per mobile action group. For each action, add a name, API name, and select an action type.



Note

- You can create a mobile action group only if mobile is selected as a supported channel.
- You can have a total of two mobile action groups per custom notification, and each group can have a maximum of three actions.



Send a Desktop or Mobile Notification From a Process

Add the Send Custom Notification action to a process, then add recipients and content.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To view notification types: View Setup and Configuration

To create and edit notification types: Customize Application

To create, edit, or view processes:
Manage Flow

AND

View All Data

- Before you begin, make sure that the custom notification type that you want to call from your process exists. If not, [create the notification type](#).

- Before you can add an action to your process, you must define the process properties, configure the process trigger, and add process criteria. If you're new to creating processes, learn more in [Process Builder help](#).

After you create an action and select **Send Custom Notification** for the type, fill in the relevant fields to add the action to your process.

- Enter an easily recognizable name for this action.

The name appears on the canvas and helps you differentiate this action from others in your process.

The name truncates to fit on the canvas.

- Select a notification type.

- Select a recipient category, and designate or find a recipient ID.

- Current User – The user who initiated the record change, platform event, or process that triggered the process. This option is useful for confirmation notifications, such as a successful submission of a form.
- Find User – The user who receives the notification each time this action is executed.
- User Field from a Record – A user referenced via UserId on the record that initiated the process or on a related record.
- Find Group – All users in the group that receives the notification each time this action is executed.
- Find Queue – All users in the queue that receives the notification each time this action is executed.
- Account Field from a Record – All users on the account team for an account referenced via AccountId on the record that initiated the process or on a related record. This option is available if you enabled account teams.
- Opportunity Field from a Record – All users on the opportunity team for an opportunity referenced via OpportunityId on the record that initiated the process or a related record. This option is available if you enabled team selling.
- Owner Field from a Record – An owner or queue referenced via OwnerId on the record that initiated the process or a related record. With this option, you can send a notification to all record owners, regardless of whether the owner is an individual owner or a queue.

- Write a helpful notification title and body using text and merge fields.



Note The content of custom push notifications depends on the [Display full content push notifications setting](#). If full content push notifications aren't enabled, only the notification title is sent.

- Save the action.

Other Ways to Send a Custom Notification

We've made it easy to build notifications using clicks, not code. But if your business needs are more complex, we also support flows and programmatic solutions.



Tip To see how to specify the target using JSON, see [pageReference](#).

Flow Builder

[Flow Core Action: Send Custom Notification](#): The Send Custom Notification action is available in Flow Builder. Add it to your flow, then add recipients and content.

Some tips specific to custom notifications:

- To query for the Notification Type ID directly from a flow, add the Get Record element to your flow and filter by API name. If you've installed a notification type via a managed package, filter by the namespace prefix as well as the API name.
- To add recipients, define Recipient ID as a resource. Then add values to your Recipient ID collection by adding the Assignment element to your flow.

If you're new to creating flows, learn even more in [Flow Reference](#).

Apex

[CustomNotification](#) : Use the **CustomNotification** Apex class to create, configure, and send custom notifications from Apex code, such as a trigger.

When possible, create and send custom notifications from Apex rather than via API call. It's less code, and more efficient.

API

CustomNotificationType: Use the **CustomNotificationType** object to create notification types and query for your Custom Notification Type ID. It's available for the following APIs:

- [Metadata](#)
- [Tooling](#)
- [Lightning Platform REST and SOAP](#)

[customNotificationAction Invocable Action](#): Use the **customNotificationAction** action to send custom notifications to desktop and delivery channels.

Custom Slack Notifications

Reach your users on Slack with important notifications.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Create and Send Custom Slack Notifications

Create custom notifications that appear on Slack when important events occur. Use a flow in Flow Builder or the invocable action API to send the notification.

Merge Fields in Slack Notifications

To show record-specific information in your Slack notification, include merge fields in the notification content.

Slack Apps Supported by Notification Builder

Learn the Slack apps that can distribute your Slack notification to recipients.

Create and Send Custom Slack Notifications

Create custom notifications that appear on Slack when important events occur. Use a flow in Flow Builder or the invocable action API to send the notification.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To view notification types: View Setup and Configuration

To create and edit notification types: Customize Application

Create a Slack Notification

Define the custom notification's details, select which Slack app distributes the notification, and then configure the Slack message.

Clone a Standard Notification Into a Custom Slack Notification

When you clone a Slack-enabled standard notification, you can use it as a template and then modify the content into a custom notification that meets your specific needs.

Ways to Send a Slack Notification

Use a flow in Flow Builder to configure how your Slack notification is sent using clicks, not code. But if your business needs are more complex, you can use the invocable action API to send your notification.

Create a Slack Notification

Define the custom notification's details, select which Slack app distributes the notification, and then configure the Slack message.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To view notification types: View Setup and Configuration

To create and edit notification types: Customize Application

Before you create a custom Slack notification, you must enable Salesforce for Slack Integrations and configure the [Slack app that distributes your notification](#).

1. From Setup, in the Quick Find box, enter *Notification Builder*, and then select **Custom Notifications**.
2. Select **New**, and then select **Slack**.
3. Add your custom notification name and API name. Then select the object associated with the notification type.
-  **Note** You can't create custom notifications for the Task object.
4. Select the Slack app that distributes the Slack message to recipients.
5. Enter the title and body text of the Slack message. You can include [merge fields](#) in the title and body text.
6. Optionally, add buttons to include in the Slack message. For each button, select the button's action, and enter its label. You can include merge fields in each button label.
 **Note** The button's position in the list corresponds to where the button appears in the Slack message. For example, the button in position 1 is placed in the left-most position in the Slack message. To reorder a button, drag the button into a new position on the list.
7. Save your notification.

Clone a Standard Notification Into a Custom Slack Notification

When you clone a Slack-enabled standard notification, you can use it as a template and then modify the content into a custom notification that meets your specific needs.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Essentials, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To view notification types: View Setup and Configuration

To create and edit notification types: Customize Application

Before you create a custom Slack notification, you must enable Salesforce for Slack Integrations and configure the [Slack app that distributes your notification](#).

 **Note** As of Summer '22, the clone option is available only for certain standard notification types that are enabled for Slack.

1. From Setup, in the Quick Find box, enter *Notification Builder*, and then select **Notification Delivery Settings**.
2. Select the Slack-enabled standard notification type that you want to clone, and then select **Clone** in the dropdown menu.
3. Add your notification name and API name.
4. Select the Slack app that distributes the Slack message to recipients.
5. Modify the Slack message title, body, and buttons as needed.
6. Save your notification.

Ways to Send a Slack Notification

Use a flow in Flow Builder to configure how your Slack notification is sent using clicks, not code. But if your business needs are more complex, you can use the invocable action API to send your notification.

Flow Builder

Flow Core Action: Send Notification: After you create your custom Slack notification, your notification is available in Flow Builder as a Send Notification action. For example, if you created a custom Slack notification named My Opportunity Notification, look for the My Opportunity Notification action in the Notifications category.

Select your custom notification as an Action in the flow, then configure the recipients and record associated with your notification.

If you're new to creating flows, learn more in [Flow help](#).

API

- [Metadata](#)
- [Tooling](#)
- [Lightning Platform REST and SOAP](#)

sendNotification Invocable Action: Use the **sendNotification** invocable action to send your custom Slack notification. Each Send Notification action corresponds to a supported notification. For example, if you created a custom Slack notification named My Opportunity Notification, you can send this notification using the **sendNotification/my_opportunity_notification** action.

Merge Fields in Slack Notifications

To show record-specific information in your Slack notification, include merge fields in the notification content.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Merge fields automatically include values from a record in your notification. The syntax for a merge field is `{!Record.Field}`. The fields that you can reference depend on the object that you select for the notification.

For example, you want to include the stage name of an Opportunity record in the title of your Slack notification. First, make sure to select **Opportunity** as the object for this notification. Then, in the notification title, include the merge field `{!Record.StageName}`.

Merge fields are supported in the notification title, body text, and button labels. You can use merge fields to reference custom objects and custom fields.

 **Note** Custom Slack notifications support only one nested level in a merge field. For example, `{!Record.Field}` is supported, but `{!Record.ChildRecord.Field}` isn't supported.

 **Example** This example notification alerts Slack users about a change in the stage of an Opportunity record. Title: `Stage update to {!Record.Name}` Body Text: `The stage of {!Record.Name} is now {!Record.StageName}. The estimated amount of this opportunity is {!Record.Amount}`. Button Action: `View Record` Button Label: `View {!Record.Name}`

Slack Apps Supported by Notification Builder

Learn the Slack apps that can distribute your Slack notification to recipients.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

To send notifications to Slack, you must enable [Salesforce for Slack Integrations](#) and then configure the Slack app that distributes your notification. Notification Builder supports these Slack apps.

- [Care Coordination for Slack](#)
- [Sales Cloud for Slack](#)
- [Salesforce for Slack](#)
- [Service Cloud for Slack](#)

Manage Notification Delivery Settings

Modify where your notifications appear and which apps deliver your notifications.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To view notification delivery settings:	View Setup and Configuration
To edit notification delivery settings:	Customize Application

For desktop and mobile notifications, choose the delivery channels or mobile delivery apps. For Slack notifications, enable or disable Slack as a delivery channel. For custom notifications, you can edit the delivery channels for supported Salesforce-provided apps and apps installed from a managed package. For standard notifications, you can edit the delivery channels for supported Salesforce-provided apps only.

1. From Setup, in the Quick Find box, enter *Notification Builder*, and then select **Notification Delivery Settings**.
2. Select the notification type, and select **Edit** from the dropdown menu.
Notification Delivery Settings shows you only the notification types available for your org.
3. Select the channels or applications for your notification type, and then save.

For desktop or mobile notifications, if either desktop or mobile delivery is listed but unavailable, you can enable it from Custom Notifications in Setup.

To modify the Slack app for a Slack notification, edit the notification from Custom Notifications in Setup.

 **Note** Mobile in-app notifications require the [Enable in-app notifications setting](#). Mobile push notifications depend on a user's device-level and, if available, app-level push notification settings. Push notifications for the Salesforce mobile app require the [Enable push notifications setting](#).

See Also

[Connect REST API Developer Guide: Notification Settings Resources](#)
[Metadata API Developer Guide: NotificationTypeConfig](#)

Considerations for Notifications

Learn about the considerations for sending and viewing notifications, such as notification limits.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To view notification types: View Setup and Configuration

To create and edit notification types: Customize Application

Sending Notifications

- You can create up to 500 custom notification types.
- Each notification can have up to 10,000 users as recipients after expanding any groups, queues, or teams. To have more recipients, you can add an action to the same process within Process Builder or to the same flow in Flow Builder.
- An org can execute up to 10,000 notification actions per hour. When you exceed this limit, no more notifications are sent in that hour, and all unsent notifications are lost. Notification actions resume in the next hour.

For example, if your notification action processes are triggered 10,250 times between 4:00 and 4:59, Salesforce executes the first 10,000 of those actions. The remaining 250 notifications aren't sent and are lost. Salesforce begins executing notification actions again at 5:00.

- When you send a custom notification from a process, the Target ID for the notification is the record that started the process. However, target records that don't have their own detail page (for example, a case comment, which appears only in a Case Comment related list) don't support direct navigation. Use Flow Builder to send the notification from a flow and specify either a different Target ID or Target Page Reference.



Tip For examples showing how to specify the target using JSON, see [pageReference](#).

- The title and body fields of custom desktop and mobile notifications support plain text only.
- Desktop notification titles have a maximum of 120 characters, and notification bodies have a maximum of 320 characters. Longer notification bodies are truncated with an ellipsis (...).
- The content of custom mobile push notifications depends on the [Display full content push notifications setting](#). If full content push notifications aren't enabled, only the notification title is sent.
- When you disable a delivery channel for a standard or custom notification type, you pause the delivery of a notification. However, a notification is still created and stored whenever an existing notification type is triggered. If you enable a delivery channel for an existing notification type, the stored notifications become visible in the notification tray for that delivery channel.
- Mobile in-app notifications require the [Enable in-app notifications setting](#). Mobile push notifications depend on a user's device-level and, if available, app-level push notification settings. Push notifications for the Salesforce mobile app require the [Enable push notifications setting](#).
- If you created a custom notification type with a mobile delivery channel before Winter '20, your notification is automatically delivered to the Salesforce mobile app. If you create a custom notification type with a mobile delivery channel in Winter '20 or later, you must manually enable the Salesforce

mobile app and any other supported apps in Notification Delivery Settings.

Viewing Notifications

- Recipients can view notifications received within the last 30 days. Older notification records are automatically deleted.
 - On desktop, a user can view notifications from the last 30 days in their notification tray. The tray shows 50 notifications at a time, ordered by the most recent notifications. The user can scroll to view older notifications within the last 30 days.
 - In the Salesforce mobile app, a user can view notifications from the last 30 days in the app's bell icon. The bell icon shows 20 notifications at a time, ordered by the most recent notifications. The user can swipe to view older notifications within the last 30 days.
- Each user's notification tray holds a maximum of 10,000 notifications. If the notification tray exceeds 10,000 notifications, the user can't receive new notifications. A user can't manually delete notifications, so when the notification tray exceeds 7,500 notifications, a purge job truncates the tray to 5,000 notifications. This purge job runs up to one time per day. To avoid dropped notifications, we recommend that a user receives an average of fewer than 5,000 notifications per day.
- In the Salesforce mobile app:
 - A push notification for an individual notification type depends on a user's app-level Push Notification Settings. If the mobile channel is enabled for a notification type, but a user disables push notifications for that type, the user doesn't receive a push notification.
 - If you enable the mobile delivery channel for a notification type after you've disabled it, users' Push Notification Settings for that type default to enabled.
- A desktop notification can be delivered in real time to up to 1,000 concurrently logged-in recipients. Additional concurrently logged-in recipients must refresh their Salesforce page to see their latest notifications. Recipients who aren't logged in see their notifications as expected upon login.
- Your org saves your most recent 1 million custom notifications for view in notification trays. Your org can save up to 1.2 million custom notifications, but it trims the amount to the most recent 1 million notifications when you reach the 1.2 million limit.
- A desktop notification's timestamp matches the time zone of the recipient's web browser. The timestamp doesn't match the time zone setting of the recipient's Salesforce user profile.

See Also

[Considerations for Salesforce Mobile App Notifications](#)

Custom Domains

Provide a branded experience for users who access your external-facing Salesforce content by serving your Digital Experiences or Salesforce Sites on a domain that you own, such as <https://www.example.com>.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, and Unlimited** Editions.

Applies to: Salesforce Sites and LWR, Aura, and Visualforce sites



<p>Learn About Custom Domains</p> <p>Custom Domains in Salesforce</p> <p>Options to Serve a Custom Domain</p>	<p>Get Ready</p> <p>Determine How to Serve Your Custom Domain</p> <p>Custom Domain Prerequisites</p> <p>Point Your Custom Domain to Your Salesforce Org</p>	<p>Complete Option-Specific Prerequisites</p> <p>Prerequisites for a Custom Domain That Uses Your HTTPS Certificate</p> <p>Prerequisites for the Salesforce CDN</p> <p>Prerequisites for a Custom Domain That Uses a Third-Party Service or CDN</p>
<p>Set Up Your Domain</p> <p>Test Your Custom Domains in a Sandbox</p> <p>Set up a custom domain that uses:</p> <ul style="list-style-type: none"> • Your HTTPS certificate • The Salesforce CDN • A third-party service or CDN • A temporary non-HTTPS domain 	<p>Serve Your Sites on Your Domain</p> <p>Add a Custom URL</p> <p>Redirect System-Managed Site URLs to Your Custom Domain</p> <p>Custom Domain Build Example</p>	<p>Maintain Your Custom Domain</p> <p>Update an Expiring Certificate for Your Custom Domain</p> <p>Custom Domain Management</p> <p>Troubleshoot Common Custom Domain Issues</p>

See Also

[Experience Cloud](#)

Custom Domains in Salesforce

When you serve your Digital Experiences or Salesforce Sites on a domain that you own, your brand can shine. Learn about domains and custom URLs in Salesforce, and understand the relationship between them.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, and Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

 **Note** Custom domains that serve your sites are unrelated to the Use Custom Domain option on the Salesforce login pages. For information about logging in with that option, see [Troubleshoot Login Issues](#). For information about including your company's brand in a subdomain of the URLs that Salesforce hosts for your org, for example, `https://mycompany.my.salesforce.com`, see [My Domain](#).

Unfamiliar with terms like DNS, CDN, and CNAME? Want to review the difference between a DNS resolver and a certificate? See [Custom Domain Terminology](#).

Why Custom Domains

The primary reason to set up a custom domain is to provide a branded experience to your users. With a custom domain, your users access your external-facing Experience Cloud site or Salesforce Site and its functionality through your branded URL. Experience Cloud sites, include Digital Experiences built with [Experience Cloud](#), [Commerce](#), and [Industries](#) licenses.

Also, with the many-to-many relationship between sites and custom domains, you can serve your external-facing content to meet your branding needs. For example, you can serve content from multiple Experience Cloud sites on one parent website, or serve multiple sites on one domain through custom paths.

Custom domains are especially useful if you own multiple brands and want them to share content. For example, let's say you have a parent company with two distinct brands. Each brand has its own registered domain, and you want each of those domains to serve the same parent website. You can use custom domains to point both brand domains to a single parent site with content from Salesforce.

Admins also benefit from custom domains. Because you can serve multiple sites on one domain, custom domains simplify the management of your domains and your corresponding network allowlists. And even if your system-hosted site URL changes due to a My Domain change, your custom domain remains constant, reducing the number of updates required.

Domains in Salesforce

Salesforce serves multiple domains for your org, including a login domain, such as `MyDomainName.my.salesforce.com`, and site domains that can end in `*.force.com`, `*.my.site.com`, or `*.my.salesforce-sites.com`. You can't edit those Salesforce-managed domains, but you can add and manage custom domains from the Domains Setup page.

When you set up a domain in Salesforce, you specify the domain that you own, such as `https://www.example.com`, and select an HTTPS option to serve that domain. For example, you can route your domain through your own HTTPS certificate, through the Salesforce content delivery network (CDN) partner, or through a third-party CDN. Then, you point your domain to your site.

Custom URLs

Whether you use Experience Cloud sites, Salesforce Sites, or both solutions, your domains and sites can have a many-to-many relationship through custom URLs. Each domain can serve up to 200 sites, and each site can be associated with up to 500 domains. An Experience Cloud site counts as two sites, so if you use only Experience Cloud sites, each domain can serve 100 sites.

After you set up a custom domain for your org, you use Custom URLs to map your domain and its paths to specific sites. For each Custom URL, you specify the domain record, the site, and the path. For example, if you have one site, you create a custom URL to point your custom domain to that site for the root path (/).

With Custom URLs, you can also serve different sites via the same domain. For example, you own `https://www.example.com`, and you have Experience Cloud sites for your online store, customer service and job postings. You can set up three custom URLs, each with a different path:
`https://www.example.com/shop`, `https://www.example.com/help`, and
`https://www.example.com/apply`. When users access the domain using one of those URLs, the custom path determines which site they see.

For an example of the many-to-many relationship between sites, see [Custom Domain Build Example](#).

See Also

[Custom Domains](#)

[Custom Domain Prerequisites](#)

[Options to Serve a Custom Domain](#)

[Add a Custom URL](#)

Determine How to Serve Your Custom Domain

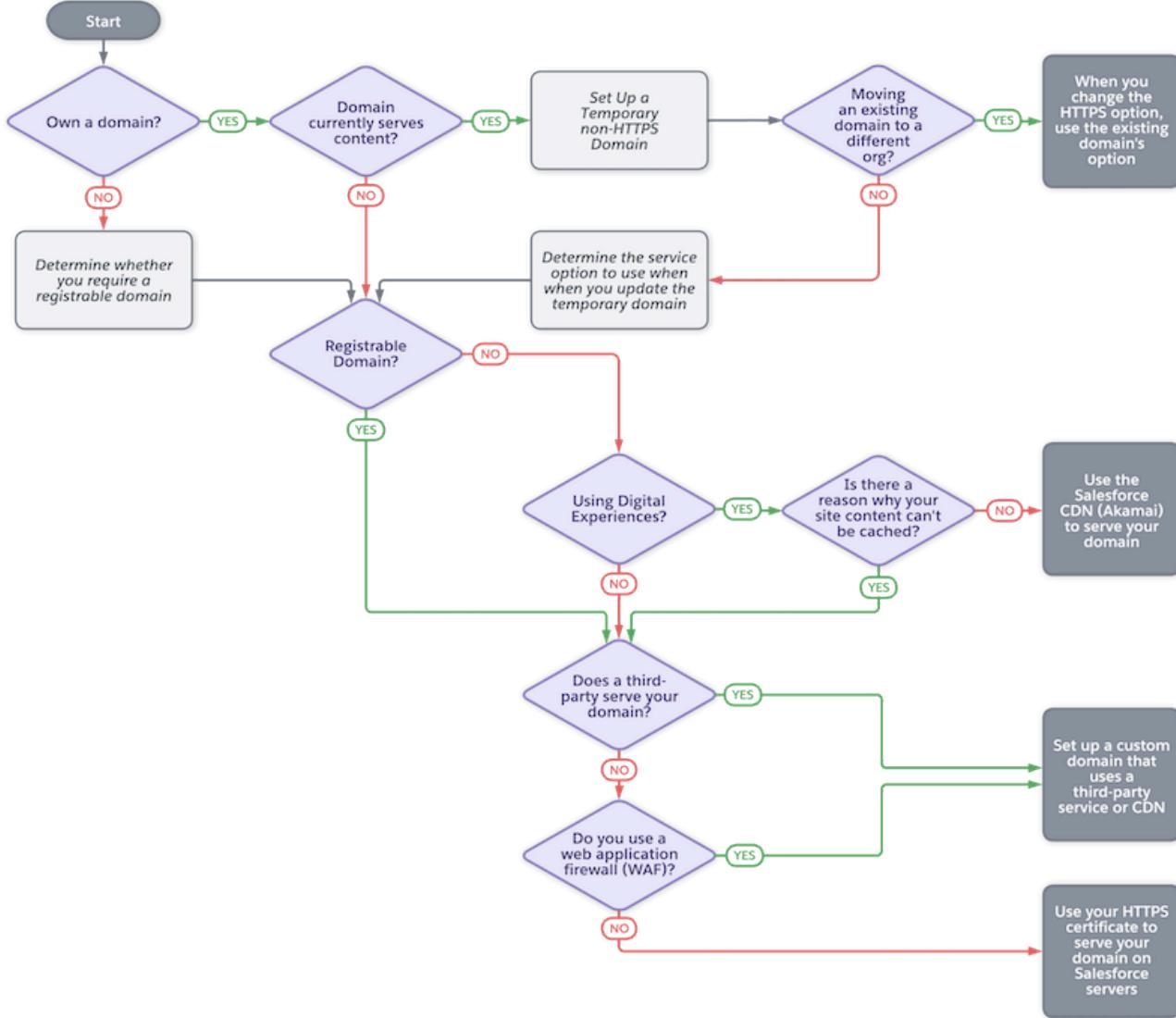
To determine the recommended domain configuration option to serve your custom domain, answer a few questions about your domain and your Salesforce configuration.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, and Unlimited** Editions.

Applies to: Salesforce Sites and LWR, Aura, and Visualforce sites



- If you don't own a domain yet, determine the domain to use to serve your site content, including whether you require a registrable domain. For more information, see [Custom Domain Prerequisites](#).
- If your domain currently serves content, to minimize the downtime for your domain, plan to set up a [temporary non-HTTPS domain](#).
 - To move a domain to another org, set up a temporary non-HTTP domain as the first step. After the domain is configured in your new org, use the same domain configuration option as the existing domain. For more information, see [Move a Domain to Another Production Org](#).
 - If your domain serves content outside of Salesforce, to determine the final option to use for your domain, go to the next step.

3. If you plan to use a registrable domain, skip to the question about whether a non-Salesforce third-party service or CDN serves your domain.
4. If you have digital experiences, including the ones built with [Experience Cloud, Commerce, and Industries](#) licenses, we recommend that you use the Salesforce content delivery network (CDN).
 - a. If your company's policies prohibit the caching of your site content, go to the question about whether a non-Salesforce third-party service or CDN serves your domain.
 - b. Otherwise, when you add your domain in Salesforce, select the domain configuration option **Serve the domain with the Salesforce Content Delivery Network (CDN)**. For instructions, see [Serve Your Experience Cloud Site with the Salesforce Content Delivery Network \(CDN\)](#).
- To learn how CDNs optimize page load times and site performance by caching your site content, see [Content Delivery Networks \(CDNs\) and Salesforce](#).
5. If a non-Salesforce third-party service or CDN serves your domain, when you add your domain in Salesforce, select the domain configuration option **Use a third-party service or CDN to serve the domain**. For instructions, see [Use a Third-Party Service or CDN to Serve Your Custom Domain](#).
6. If your domain uses a web application firewall (WAF), when you add your domain in Salesforce, select the domain configuration option **Use a third-party service or CDN to serve the domain**. For instructions, see [Use a Third-Party Service or CDN to Serve Your Custom Domain](#).
7. If none of the other scenarios apply, when you add your domain in Salesforce, select the domain configuration option **Serve the domain with your HTTPS certificate on Salesforce servers**. For instructions, see [Serve a Custom Domain with Your HTTPS Certificate on Salesforce Servers](#).

For more information about each of these options and the corresponding traffic flows, see Options to Serve a Custom Domain.

See Also

- [Custom Domains](#)
- [Options to Serve a Custom Domain](#)
- [Custom Domain Prerequisites](#)

Options to Serve a Custom Domain

Salesforce supports three HTTPS options to serve your domain. Whether you serve content for your domain via an HTTPS certificate that you own, through the Salesforce content delivery network (CDN), or through an external service, we require that you use HTTPS. When initial configuration requires that your domain is available before you enable HTTPS, Salesforce also supports a temporary non-HTTPS option.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, and Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

 **Tip** This topic provides details on the four options to serve your custom domain. To determine which is the correct option for you, see [Determine How to Serve Your Custom Domain](#).

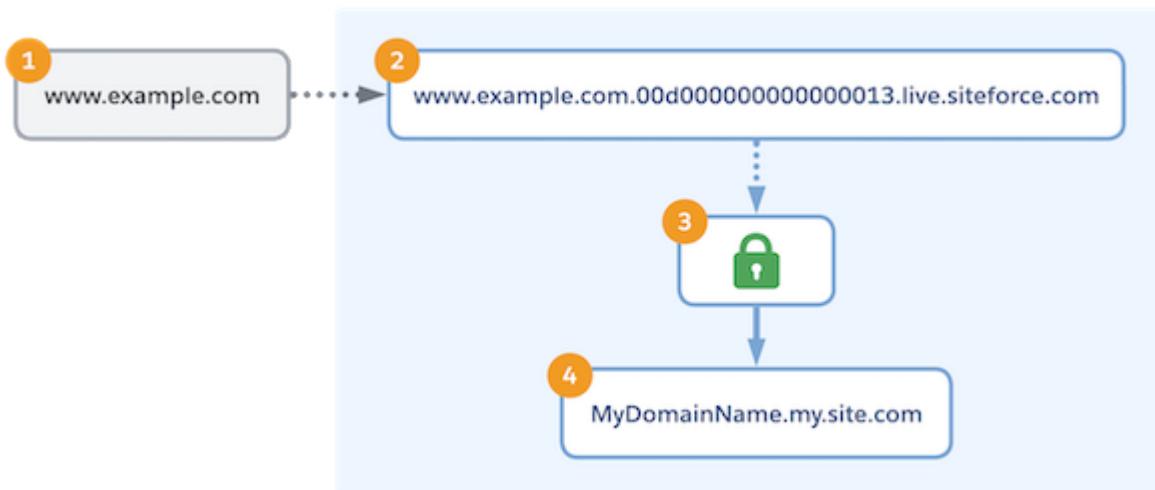
Unfamiliar with terms like DNS, CDN, and CNAME? Want to review the difference between a DNS resolver and a certificate? See [Custom Domain Terminology](#).

Serve the Domain with Your HTTPS Certificate on Salesforce Servers

With this option, you upload your HTTPS certificate to Salesforce and configure your domain to use that certificate. This option requires a certificate authority (CA)-signed certificate and that the DNS record for your domain point directly to your Salesforce org.

If the DNS record for your domain points to an external service, you can't use this option. Common examples of an external service include a web application firewall (WAF), a third-party host, or a third-party content delivery network (CDN). To set up a domain that points to an external service, choose the option to use a third-party service or CDN to serve your domain.

This diagram shows the routing of traffic when Salesforce uses your HTTPS certificate to serve your Experience Cloud site content on your custom domain. Dotted lines (·····→) represent DNS configurations, and the solid line (→) represents user traffic flow through HTTPS. The gray line represents traffic that originates outside Salesforce, and the blue lines represent traffic that originates in Salesforce. In this example, the domain name is www.example.com and the 18-digit org ID is 00d0000000000000013.



With your DNS provider, you point your custom domain (1) to the Salesforce internal canonical name (CNAME) record for your org (2), which includes your org ID. In Salesforce, your certificate is stored on a secure server (3). Salesforce uses that certificate to serve the content from your Experience Cloud site (4).

For more information on this option, including prerequisites, see [Serve a Custom Domain with Your HTTPS Certificate on Salesforce Servers](#).

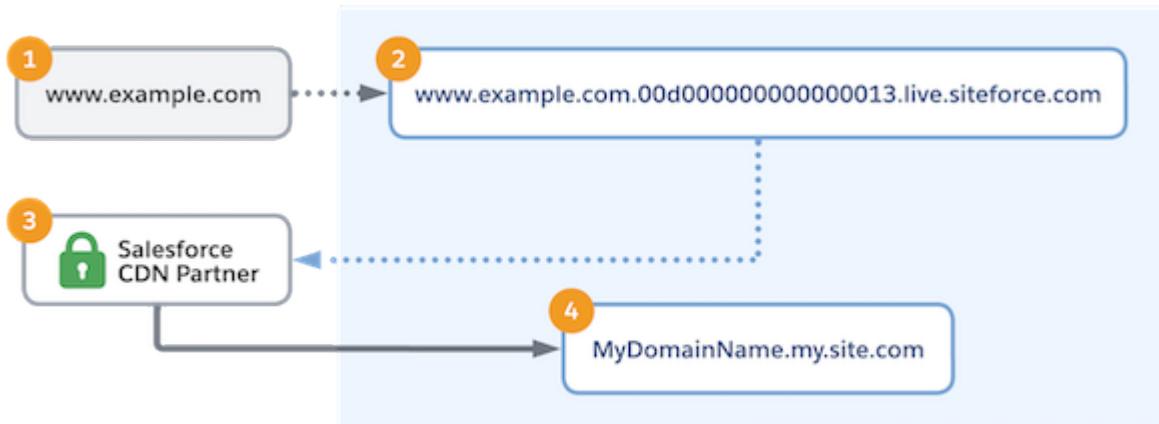
Serve the Domain with the Salesforce Content Delivery Network (CDN)

With this option, you optimize page load times and site performance for your Experience Cloud site. Salesforce partners with a CDN provider to efficiently deliver publicly cacheable content to users on your Experience Cloud sites.

The Salesforce CDN is the recommended option for custom domains that serve Digital Experiences, including Experiences built with [Experience Cloud](#), [Commerce](#), and [Industries](#) licenses.

If you use Marketing Cloud Account Engagement (Pardot) in a Professional Edition org, the Salesforce CDN is the only HTTPS option available for your custom domains. The Salesforce CDN isn't available for Salesforce Sites or in Professional Edition orgs without Marketing Cloud Account Engagement.

This diagram shows the routing of traffic when Salesforce serves your custom domain with the Salesforce CDN. Dotted lines (.....➡) represent DNS configurations, and the solid line (➡) represents user traffic flow through HTTPS. The gray lines represent traffic that originates outside Salesforce, and the blue line represents traffic that originates in Salesforce. In this example, the domain name is www.example.com and the 18-digit org ID is 00d000000000000013.



With your DNS provider, you point your custom domain (1) to the Salesforce internal CNAME (2), which includes your org ID. Within Salesforce, user traffic is routed to the Salesforce CDN partner (3), which acts as an intermediary for your Salesforce content (4).

-  **Note** The Salesforce CDN for Digital Experiences serves only subdomains, such as www.example.com or parts.example.com. Salesforce is unable to serve a registrable domain, such as example.com, when using the CDN for Digital Experiences. If your site needs a registrable domain served via a CDN, host it on a CDN outside of Salesforce Experience Cloud.

To review the benefits, limitations, and instructions for this option, see [Serve Your Experience Cloud Site with the Salesforce Content Delivery Network \(CDN\)](#).

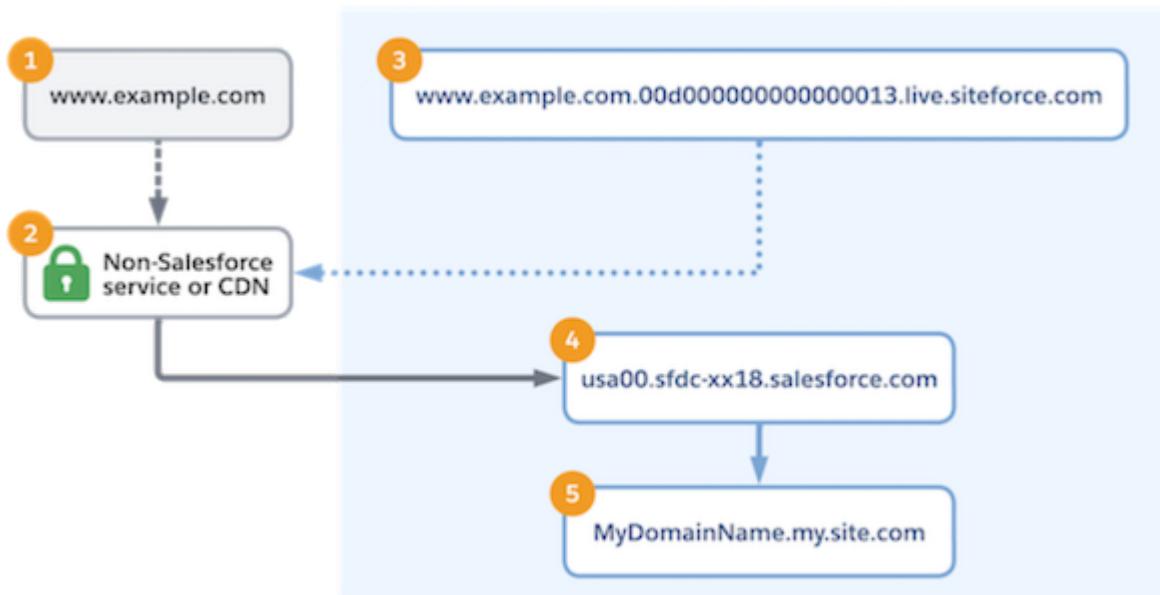
Use a Third-Party Service or CDN to Serve the Domain

If a non-Salesforce service or host serves your custom domain, you can still serve your site content via that domain. Common examples include a third-party CDN, a web application firewall (WAF), or a third-party hosting service that either provides or uses your HTTPS certificate.

With this configuration option, your third-party service or CDN serves as an intermediary. In other words, traffic flows through the third party. With this configuration option, your domain points to the third-party service.

This diagram shows the routing of traffic when a third-party service or CDN serves content from your Experience Cloud site on your custom domain. In this example, the domain name is www.example.com, the 18-digit org ID is 00d000000000000013, and the org's target host name is usa00.sfcd-xx18.salesforce.com.

The dashed line () represents the configuration that points your domain to your third-party service or CDN. The dotted line () represents routing through DNS, and the solid lines () represent user traffic flow through HTTPS. The gray lines represent traffic that originates outside Salesforce, and the blue lines represent traffic that originates in Salesforce.



Your custom domain (1) points to the third-party service or CDN (2). For example, you point your custom domain to the third party in DNS. Or you set up a web application filter (WAF) as a proxy.

In Salesforce, you specify the external host name for your domain. The Salesforce CNAME (3) uses that external host name to point to your non-Salesforce service or CDN (2). In the third-party service or CDN, the configuration points to your org's target host name (4). To identify the domain and serve content from your site (5), Salesforce uses the value passed in the Host HTTP Header of the request from the third-party service or CDN.

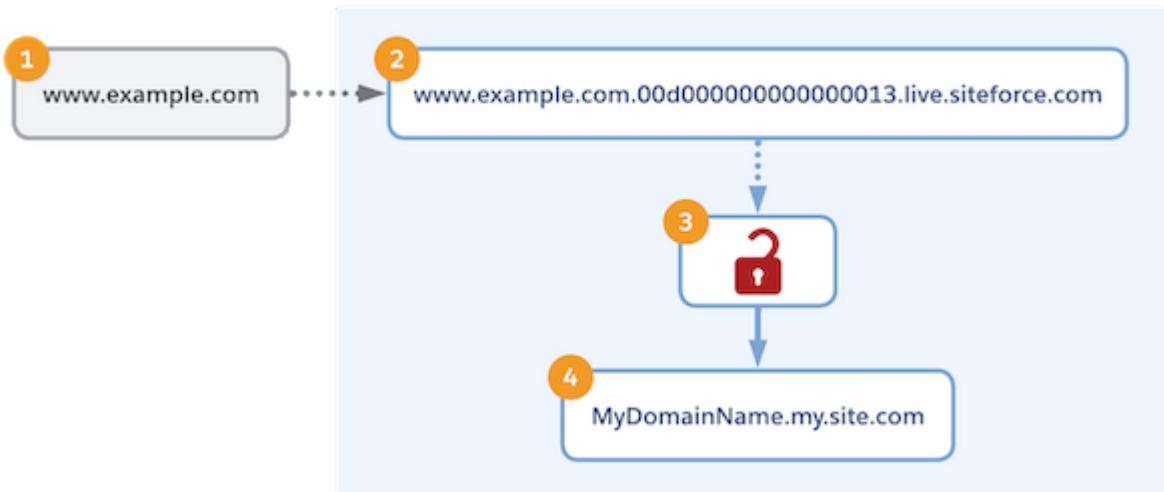
For more information on this option, including considerations and prerequisites, see [Use a Third-Party Service or CDN to Serve Your Custom Domain](#).

Use a Temporary Non-HTTPS Domain

Salesforce requires that your custom domain is served via HTTPS. However, some configuration steps can require a temporary non-HTTPS configuration. When your domain serves other content, or to move a custom domain to a new production org, you can use a temporary domain to minimize disruption to your domain. Or when your HTTPS certificate isn't ready to be uploaded to Salesforce, you can use a temporary domain to start configuring your custom URLs.

-  **Note** This option is intended as a temporary configuration only. When your custom domain is served via HTTP, users who attempt to access your custom domain via HTTPS can see a certificate mismatch error and experience a connection timeout.

This diagram shows the routing of traffic when Salesforce serves your domain as a temporary non-HTTPS domain. Dotted lines (···→) represent DNS configurations, and the solid line (→) represents user traffic flow through HTTP. The gray line represents traffic that originates outside Salesforce, and the blue lines represent traffic that originates in Salesforce. In this example, the domain name is www.example.com and the 18-digit org ID is 00d000000000000013.



To confirm ownership of your custom domain (1), with your DNS provider, you point the domain to the Salesforce internal CNAME (2), which includes your org ID, via a CNAME or TXT record. If a CNAME record routes traffic to Salesforce, Salesforce uses an HTTP-only endpoint that's served on a secure server (3) to serve the content from your Experience Cloud site (4). However, the hosted certificate (3) supports only HTTP on the custom domain instead of HTTPS. Also, the returned certificate creates a hostname-mismatch error because that certificate doesn't support the custom domain name.

For more information, see [Use a Temporary Non-HTTPS Domain to Serve Your Custom Domain](#).

See Also

Custom Domains

Custom Domain Prerequisites

Before you serve your site content with a custom domain, register a domain. To get that domain ready to serve your site content, update the Domain Name System (DNS) record to point to your org. Review your site URLs for potential conflicts, and complete the prerequisites for your chosen domain configuration option.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, and Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

Determine Your Desired Domain Configuration Option

The domain configuration option for your domain specifies how the domain serves your site content via HTTPS. If you have Experience Cloud licenses, we recommend that you use the Salesforce content delivery network (CDN). For both Experience Cloud sites and Salesforce Sites, you can serve your domain with your HTTPS certificate on Salesforce servers. Or you can use a non-Salesforce service or CDN to host your domain.

If you don't have a domain yet, your chosen domain configuration option can affect the features that you require from your domain name registrar. For example, if you chose to serve your site via a third-party CDN, you can select a company that provides domain name registration and the CDN service. For more information, see [Determine How to Serve Your Custom Domain](#).

Register a Domain

If you own a domain such as `www.example.com`, you can use it to serve your site content.

If you don't own a domain, choose a domain name and register it. When selecting a registrar, consider these factors.

- If you require services other than domain registration, selecting a provider that offers multiple services can reduce your cost.
- The reputation of the registrar or service provider. If you plan to bundle services, their reputation is especially important. Ensure that your selected provider offers sufficient privacy protections.
- Salesforce doesn't recommend registrable domains, such as `example.com` instead of `www.example.com`. If you choose to serve your site with a registrable domain, see the important considerations in the section on Registrable Domains.
- The contract's duration. A long-term contract can provide lower overall cost. With a shorter-term

contract, if you become dissatisfied with the service, you can transfer your domain name to another company.

Registrable Domains

A registrable domain—sometimes called a root domain or naked domain—is the domain's public suffix, such as `.com` or `.org`, plus the label to the left of that suffix. An example is `example.com`, without the `www` subdomain.

You can't use the [Salesforce CDN](#) with a registrable domain. Because the Salesforce CDN is the recommended configuration for custom domains that serve Experience Cloud sites, Salesforce doesn't recommend registrable domains.

If you choose to serve your sites on a registrable domain, ensure that your provider supports alias records or CNAME flattening.

One of the prerequisites for a custom domain is to configure a canonical name (CNAME) record on your domain that points to your Salesforce org. However, you can't set a CNAME record on a registrable domain because of a DNS limitation. To bypass this limitation, some DNS vendors implement a CNAME behavior for registrable domains that mimics the standard CNAME behavior. To serve your site with a registrable domain, ensure that your DNS provider supports this workaround, which is often referred to as an alias record or CNAME flattening.

If your DNS provider doesn't support this workaround, you can't use your registrable domain to serve your Experience Cloud sites or Salesforce Sites.

If you choose to serve your sites on a registrable domain, we recommend that you configure [enable HSTS preloading](#) on the domain.

Update Your Domain DNS Record to Point to Your Org

Before you add a domain in Salesforce, update the DNS record for your fully qualified domain name (FQDN) to include your 18-digit Salesforce org ID. For more information, see [Point Your Custom Domain to Your Salesforce Org](#).

Select the Org to Serve Your Site

Except for testing in a sandbox, a single fully qualified domain name (FQDN) is meant to exist in one org only. If you have multiple production orgs, determine the org in which to add and maintain the domain.

-  **Note** Although it's technically feasible for domains with the same FQDN to exist in multiple non-sandbox orgs, that configuration isn't supported.

Review Your Site URLs for Potential Conflicts

If you host multiple sites on the same domain, be sure to review your site URLs for conflicts because it's possible to configure the same URL for pages on two different sites.

Let's say that you host Site A and Site B on the same domain, `https://www.example.com`. Site A's URL uses the custom URL path prefix `/products`. Site B serves pages from the root path and has a page with the page path `/products`. As a result, both Site A's URL and Site B's page URL are `https://www.example.com/products`.

In this scenario, a site visitor can access the Site B page only through a navigation menu on Site B. If a site visitor navigates to `https://example.com/products` any other way, they're directed to Site A.

If you identify a potential conflict, either rename the site path or choose a different path for serving the content on your custom domain.

Complete Additional Prerequisites Based on Your Chosen Domain Configuration Option

Review the prerequisites for your selected domain configuration option. See [Prerequisites for a Custom Domain That Uses Your HTTPS Certificate](#), [Prerequisites for the Salesforce CDN](#), and [Prerequisites for a Custom Domain That Uses Your HTTPS Certificate](#).

See Also

[Custom Domains](#)

Point Your Custom Domain to Your Salesforce Org

Before you add a domain in Salesforce, point your domain to your Salesforce org in Domain Name Service (DNS). If your custom domain uses your HTTPS Certificate or the Salesforce content delivery network (CDN) partner, add a canonical domain name (CNAME) record for your fully qualified domain (FQDN) to DNS. The CNAME record references your Salesforce org ID and your FQDN. Then, if you plan to use the Salesforce CDN to serve your Experience Cloud site on your custom domain, add a second CNAME record that our CDN partner requires.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

USER PERMISSIONS NEEDED

To view a domain:

Manage Custom Domains

OR

View Setup and Configuration

To add a domain:

Manage Custom Domains

OR

View Setup and Configuration plus either a Site.com Publisher license or Create and Set Up Experiences

To edit or delete a domain:

Manage Custom Domains



Tip Unfamiliar with terms like DNS, CDN, and CNAME? Want to review the difference between a DNS resolver and a certificate? See [Custom Domain Terminology](#).

When you add a custom domain, Salesforce checks the domain's DNS configuration to verify that you own the domain. To meet this requirement, configure a CNAME record for your domain in DNS that point to your org. If you use the Salesforce content delivery network (CDN) or if Salesforce serves your domain with your HTTPS certificate, the CNAME record is required for your permanent custom domain configuration. If a third-party service or CDN serves your domain, the CNAME record is required for initial setup only.



Note If your domain serves other content via an existing A, AAAA, or CNAME record, you can't add a CNAME record. If you update the DNS record to use the CNAME for your Salesforce org instead, your site is disrupted. To minimize that disruption, [set up a temporary non-HTTPS domain](#) that uses a temporary TXT record in DNS to validate your ownership and configure your custom domain. Then, after you finish that configuration, update that temporary domain to your final HTTPS configuration option.

1. Get the internal Salesforce CNAME.

- a. To test a custom domain in a sandbox, you create the domain in production and select the sandbox as the associated org. In DNS, use the internal Salesforce CNAME for your production org.

A canonical name (CNAME) record is an entry in the DNS record of a domain that points to a domain name instead of an IP address. When you add a domain in Salesforce, we verify that the domain points to your org via a CNAME record. That CNAME record has a target of your internal Salesforce CNAME, which includes your FQDN and your 18-digit Salesforce org ID.

You can find your internal Salesforce CNAME and 18-character org ID at the top of the Domain Setup page.

To serve your Experience Cloud sites or Salesforce Sites on a domain that you own, add and activate the domain. Then add at least one [custom URL](#).

Before you add or rename a domain, point your domain to **[domain].00d0000000000maq.live.siteforce.com** in DNS. For example, if your domain name is www.example.com, add a CNAME record in DNS to www.example.00d0000000000maq.live.siteforce.com. To learn about how to verify ownership of the domain by pointing your domain to your org, see [Salesforce Help](#).

Note: The unique API identifier for this org is **00d000000000000maq** in lowercase characters.

Domain Edit	
	<input type="button" value="Save"/> <input type="button" value="Save & New"/> <input type="button" value="Cancel"/>
Domain Name	<input type="text"/>
Domain Configuration Option	<input checked="" type="radio"/> Serve the domain with your HTTPS certificate on Salesforce servers <input type="radio"/> Serve the domain with the Salesforce Content Delivery Network (CDN) i <input type="radio"/> Use a third-party service or CDN to serve the domain i <input type="text" value="Enter your third-party hostname..."/> <input type="radio"/> Use a temporary non-HTTPS domain i
Allow HSTS preloading registration	<input checked="" type="checkbox"/> Allow HSTS preloading registration
<p>i After saving, you will need to activate the domain for your changes to take effect.</p>	
	<input type="button" value="Save"/> <input type="button" value="Save & New"/> <input type="button" value="Cancel"/>

To get to this page, from Setup, in the Quick Find box, enter *Domains*, then select **Domains**, and then select **Add a Domain**. The option to add a domain is available only if at least one site exists in the org.

This internal Salesforce CNAME is in the format

[YourFQDN] . [Your18CharOrgId].live.siteforce.com .

Your domain's fully qualified domain name (FQDN) is all the parts of the domain required to look up this authority by name unambiguously using the internet's DNS system. For example, www.example.com.

For example, to add www.example.com as a domain in a production org, if your 18-character org ID is 00d000000000000maq, your domain's internal Salesforce CNAME is www.example.com.00d0000000000maq.live.siteforce.com.

2. If you plan to use a registrable domain to serve your site, verify that your DNS provider supports alias records or CNAME flattening. Then use the DNS vendor's configuration system to point the domain to your internal Salesforce CNAME in DNS.

A registrable domain is a top-level domain, such as example.com without the www subdomain. For more information on registrable domains, also known as root domains or naked domains, see [Custom Domain Prerequisites](#).

Also note these limitations for registrable domains.

- If your DNS provider doesn't support alias records or CNAME flattening, you can't use a registrable

domain to serve your sites.

- Salesforce is unable to serve a registrable domain via our content delivery network (CDN) partner. The Salesforce CDN only serves subdomains such as `www.example.com` or `parts.example.com`. If your site needs a registrable domain served from a CDN, host it on a CDN outside of Salesforce.

- If you don't plan to use a registrable domain to serve your site, work with your DNS provider to update DNS. Add a CNAME record for your fully qualified domain name that points to your internal Salesforce CNAME.

Here's an example of a CNAME record for `www.example.com` that points to an org with ID `00d00000000000maq`.

NAME	TTL	CLASS	TYPE	VALUE
<hr/>				
<code>www.example.com.</code>	3600	IN	CNAME	<code>www.example.com.00d00000000000maq.live.siteforce.com.</code>

- If you plan to use the Salesforce CDN to serve your Experience Cloud site on your custom domain, get the target value for your domain's `_acme-challenge` CNAME record.
 - In Setup, enter `Domains` in the Quick Find box, and then select **Domains**, and then add or edit a domain.
 - Select **Serve the domain with the Salesforce Content Delivery Network (CDN)** as the Domain Configuration Option.

The details for that option include the format of the `_acme-challenge` CNAME record.

The screenshot shows the 'Domain Edit' page in the Salesforce setup interface. At the top, there's a note about adding or renaming a domain. Below that, the 'Domain Configuration Option' section is highlighted with an orange border. It contains three radio button options: 'Serve the domain with your HTTPS certificate on Salesforce servers' (disabled), 'Serve the domain with the Salesforce Content Delivery Network (CDN)' (selected), and 'Use the Salesforce CDN for a site that's hosted on Experience Delivery' (disabled). A callout box points to the second option with the text: 'To use this option, ensure that: • The public DNS record of the domain name points directly to [domain].00dsg00000pvxa2a4.live.test1.pc-rnd.siteforce.com. • The public DNS record of the domain name includes a CNAME entry with hostname _acme-challenge.[domain] and target _acme-challenge. [domain].00dsg00000pvxa2a4.live.test1.pc-rnd.siteforce.com'. Below this, there's a note about third-party services and temporary domains, followed by a checkbox for HSTS preloading registration which is checked. A yellow info bar at the bottom says: 'After saving, you will need to activate the domain for your changes to take effect.' At the bottom right are 'Save', 'Save & New', and 'Cancel' buttons.

To get the values for your domain, replace [domain] with your FQDN. The full format for the _acme-challenge name is: _acme-challenge. [YourFQDN], and the format for the target value is _acme-challenge. [YourFQDN] . [Your18charOrgId] . live . siteforce . com.

5. If you plan to use the Salesforce CDN to serve your Experience Cloud site on your custom domain, work with your DNS provider to add your domain's _acme-challenge CNAME record.

This CNAME record, referred to as the _acme-challenge, is for your domain's certificate. The first CNAME record that you added is for your domain, pointing to your internal Salesforce CNAME. Both CNAME records are required for the Salesforce CDN.

! **Important** If other TXT records exist in DNS for your domain's _acme-challenge subdomain, remove them before you provision your domain.

Here's an example _acme-challenge CNAME record for `www.example.com` that points to an org with ID `00d00000000000maq`.

NAME	TTL	CLASS	TYPE	VALUE
<hr/>				
<code>_acme-challenge.www.example.com.</code>	3600	IN	CNAME	<code>_acme-challenge.www.example.com.00d00000000000maq.live.siteforce.com.</code>

6. Verify the CNAME record values.

- On Windows, open Command Prompt, then enter `nslookup -type=recordtype domain`, where `domain` is the domain name you intend to use as your custom domain, such as `www.example.com`.
- On MacOS or Linux, open Terminal, and then enter `dig -t recordtype domain`, where `domain` is the domain name you intend to use as your custom domain, such as `www.example.com`. If the returned CNAME values are incorrect or missing, work with your hosting provider to update the CNAME records.

! **Note** Some hosting providers' configuration options can modify the CNAME value and prevent Salesforce from verifying ownership of that domain. For example, with some hosting providers, you can't use the proxied option for DNS CNAME configuration.

See Also

- [Custom Domains](#)
- [Custom Domain Prerequisites](#)
- [Use a Temporary Non-HTTPS Domain to Serve Your Custom Domain](#)

Test Your Custom Domains in a Sandbox

Develop and test your custom domains that serve content from your Experience Cloud sites and Salesforce Sites in a sandbox before you activate the domain in production.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

[Considerations for Custom Domains in Sandboxes](#)

Determine whether you can set up a custom domain in your sandbox, and review considerations for using a content delivery network (CDN) with a custom domain in a sandbox. Learn how to prevent

errors when you refresh, clone, or delete a sandbox associated with a custom domain.

[Set Up a Custom Domain for Testing in a Sandbox](#)

Test a new domain before you activate it in production, or test changes to your custom domain or to your sites. Create a custom domain in production that points to your sandbox. That domain serves the Experience Cloud sites or Salesforce Sites in your sandbox with the custom URLs in the sandbox.

[Activate a Sandbox Custom Domain in Production](#)

After you test a custom domain that serves your Experience Cloud sites or Salesforce Sites in a sandbox, move any updates to your sites and the custom URLs for your domain to production. Then activate the custom domain in production.

See Also

[Custom Domains](#)

Considerations for Custom Domains in Sandboxes

Determine whether you can set up a custom domain in your sandbox, and review considerations for using a content delivery network (CDN) with a custom domain in a sandbox. Learn how to prevent errors when you refresh, clone, or delete a sandbox associated with a custom domain.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, and Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

Feature Exclusions

In each production environment, you can define multiple sandbox custom domains, but only one custom domain associated with a sandbox org can use the Salesforce content delivery network (CDN).

Third-party CDNs are supported only if the CDN configuration meets the [prerequisites](#).

Sandbox Refreshes from Production

When you refresh a sandbox from its owning production Salesforce org, the custom domain remains associated with the sandbox in a provisioned but inactive state. After you activate the refreshed sandbox, log in and validate the sandbox Salesforce Sites and Experience Cloud sites domains from the Custom URLs Setup page.

Cloning a Sandbox

You can create a sandbox by cloning an existing sandbox rather than using your production org as the

source. When you activate a cloned sandbox, any custom domains associated with the sandbox remain associated with the inactive source sandbox. Until the custom domain is associated with the cloned sandbox again, calls to the custom domain return an error.

To associate your custom domain with the cloned sandbox, after the cloned sandbox is activated, log in to your production org. From the Domains Setup page, edit the custom domain, and change the associated org to an org other than your sandbox. Save your changes, and then activate the domain. Then edit the custom domain again, and update the associated org to your sandbox. Save and activate your domain.

After you reactivate the custom domain for your sandbox, log in to the sandbox and validate the sandbox Salesforce Sites and Experience Cloud sites domains from the Custom URLs Setup page.

Custom Domains That Point to a Sandbox Experience Builder Site

After you refresh or clone a sandbox, republish your sandbox Experience Builder sites. Calls to a custom domain that points to an Experience Builder site return an error until you publish the site.

Deleting a Sandbox

Before you delete a sandbox, log in to production and associate any custom domains for that sandbox with a different org. If you don't update the domain in production, calls to the custom domain URL return an error.

See Also

[Custom Domains](#)

[Test Your Custom Domains in a Sandbox](#)

[Create, Clone, or Refresh a Sandbox](#)

[Use a Third-Party Service or CDN to Serve Your Custom Domain](#)

Set Up a Custom Domain for Testing in a Sandbox

Test a new domain before you activate it in production, or test changes to your custom domain or to your sites. Create a custom domain in production that points to your sandbox. That domain serves the Experience Cloud sites or Salesforce Sites in your sandbox with the custom URLs in the sandbox.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, and Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

USER PERMISSIONS NEEDED

To view a domain:

Manage Custom Domains

OR

View Setup and Configuration

To add a domain:

Manage Custom Domains

OR

View Setup and Configuration plus either a Site.com Publisher license or Create and Set Up Experiences

To edit or delete a domain:

Manage Custom Domains

To add, edit, and delete custom URLs:

Manage Custom Domains

OR

View Setup and Configuration AND either Create and Set Up Experiences OR a Site.com Publisher license



Note To test a custom domain in a sandbox, create the domain in production and select the sandbox as the associated org. You can view the domain in your sandbox, but you manage sandbox custom domains in production.

Before you start, review the [considerations](#) for this feature. Then, in your sandbox, create the Experience Cloud sites or Salesforce Sites that you want to serve with the custom domain.

1. Determine the domain configuration option for serving your custom domain. See [Determine How to Serve Your Custom Domain](#).
2. Complete the [prerequisites](#) for a custom domain, including any prerequisites for your chosen domain configuration option.
To test a custom domain that serves content from sites in your production org, use the 18-character org ID for your production org when you update DNS. For more information, see [Point Your Custom Domain to Your Salesforce Org](#).
3. In production, create the custom domain that serves site content from your sandbox.
 - a. If you have a certificate authority (CA)-signed certificate using Certificate and Key Management for your domain, see [Set Up a Custom Domain That Uses Your HTTPS Certificate](#).
 - b. To use the Salesforce CDN partner to host an Experience Cloud site on your custom domain, see [Set Up a Custom Domain That Uses the Salesforce CDN](#).

In each production environment, only one sandbox custom domain at a time can use the Salesforce content delivery network (CDN). To use the Salesforce CDN in a different sandbox custom domain, update the existing sandbox custom domain that serves an Experience Cloud site via the CDN partner. Specifically, change the domain configuration option or associated org for that domain so that it no longer uses the Salesforce CDN partner in a sandbox.

If you use Marketing Cloud Account Engagement (Pardot) in a Professional Edition org, you must use the Salesforce CDN partner to serve your custom domain.

- c. If your domain is served by a web application firewall (WAF), an external service, or a non-Salesforce service CDN, see [Set Up a Custom Domain That Uses a Third-Party Service or CDN](#).
 - d. If your domain already serves content, you can use a temporary non-HTTPS domain to minimize disruption. You can also use a temporary non-HTTPS domain to configure a domain for which your HTTPS certificate isn't ready. After you configure your custom domain, switch to one of the other supported domain configuration options. See [Use a Temporary Non-HTTPS Domain to Serve Your Custom Domain](#).
 - e. For Associated Org, select your sandbox.
The custom domain serves the Experience Cloud sites or Salesforce Sites in the associated org.
4. Save your domain.
- After you save your new domain, Salesforce provisions the domain or gets it ready for use. The provisioning process can take 4–14 hours. When that process is complete, the domain's status on the Domains Setup page changes to Awaiting Activation and you receive an email.
5. To activate your domain, on the Domains Setup page in production, click **Activate** next to your custom domain.
- Your site can be unavailable for 5–10 minutes, so activate your custom domain when your site traffic is low. When your site is active, the status changes from Awaiting Activation to Completed and the Current Domain Configuration Option changes to reflect the option that you selected.
6. To serve your sites via your activated domain, log in to the sandbox and [add a custom URL](#).

Your custom domain is ready for testing. After you complete your testing, you can [activate the custom domain in production](#).

See Also

[Custom Domains](#)

[Test Your Custom Domains in a Sandbox](#)

[Experience Cloud](#)

[Salesforce Sites](#)

Activate a Sandbox Custom Domain in Production

After you test a custom domain that serves your Experience Cloud sites or Salesforce Sites in a sandbox, move any updates to your sites and the custom URLs for your domain to production. Then activate the custom domain in production.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, and Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

USER PERMISSIONS NEEDED

To view a domain:

[Manage Custom Domains](#)

OR

[View Setup and Configuration](#)

To edit a domain:

[Manage Custom Domains](#)

Before you activate your custom domain in production, [set up a custom domain for testing in sandbox](#), and complete testing. Then move the Experience Cloud site and custom URLs that the sandbox custom domain serves to production. For example, move your Experience Cloud sites and the custom URLs associated with the domain via a change set.

1. If your domain serves content from a site in your sandbox, validate that the content that the domain serves exists in production.
Check for the Experience Cloud site or Salesforce Site, the relevant site pages and content, and the custom URLs for the domain.
2. Validate that your DNS record points to your production org. See [Point Your Custom Domain to Your Salesforce Org](#).
If your custom domain points to your sandbox or another production org in DNS, point the custom domain to your production org, and test in your sandbox again.
3. Update the custom domain in Salesforce.
 - a. From Setup in production, in the Quick Find box, enter *Domains*, and then select **Domains**.
 - b. For your existing custom domain, select **Edit**.
 - c. Update the value in the Associated Org field from your sandbox to **Production**, and save your changes.

After you save your changes, Salesforce provisions the domain or gets it ready for use. When the provisioning process is complete, you receive an email and the domain status changes to Awaiting Activation.

4. To activate your domain, on the Domains Setup page, click **Activate** next to your custom domain. Your site can be unavailable for 5–10 minutes, so activate your custom domain when your site traffic is low. When your site is active, the status changes from Awaiting Activation to Completed and the Current Domain Configuration Option changes to reflect your selected domain configuration option.

See Also

[Custom Domains](#)

[Test Your Custom Domains in a Sandbox](#)

[Set Up and Configure Your Org for Experience Cloud Sites](#)

Serve a Custom Domain with Your HTTPS Certificate on Salesforce Servers

Use an HTTPS certificate that you own to serve your custom domain. After you set up your domain in Salesforce, that domain can serve your Experience Cloud sites or Salesforce Sites.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

-  **Note** With this option, Salesforce hosts your domain using your HTTPS certificate. If the DNS record for your domain points to an external service, you can't use this option. Common examples of an external service include a web application firewall (WAF), a third-party host, or a third-party content delivery network (CDN). To set up a domain that points to an external service, see [Use a Third-Party Service or CDN to Serve Your Custom Domain](#).

Prerequisites for a Custom Domain That Uses Your HTTPS Certificate

Before you serve your domain with an HTTPS certificate that you own, confirm that this domain configuration option applies to your configuration. Then update your DNS configuration, and upload your HTTPS certificate.

Set Up a Custom Domain That Uses Your HTTPS Certificate

Upload an HTTPS certificate that you own to Salesforce servers, then serve your custom domain with that certificate. After you set up your domain in Salesforce, that domain can serve your Experience Cloud sites or Salesforce Sites.

See Also

[Custom Domains](#)

[Options to Serve a Custom Domain](#)

Prerequisites for a Custom Domain That Uses Your HTTPS Certificate

Before you serve your domain with an HTTPS certificate that you own, confirm that this domain configuration option applies to your configuration. Then update your DNS configuration, and upload your HTTPS certificate.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, and Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

 **Tip** Unfamiliar with terms like DNS, CA, and CNAME? See [Custom Domain Terminology](#).

1. Confirm that your HTTPS certificate meets these requirements.
 - a. A certificate authority (CA) signed the certificate, or your CA is prepared to sign your certificate.
 - b. The expiration date is a future date.
 - c. The certificate matches the domain name. Optionally, use a wildcard or Subject Alternative Name (SAN) certificate.

For example, to add `www.example.com` as a domain name in Salesforce, you can use an HTTPS certificate for `www.example.com`. Or, if your certificate authority supports SAN certificates, you use a SAN certificate that includes `www.example.com`. To have the certificate cover `www.example.com`, `shop.example.com`, and `parts.example.com`, you can use an HTTPS certificate for `*.example.com`.

2. To point your domain directly to the internal canonical name (CNAME) for your org, update DNS. For instructions, see [Point Your Custom Domain to Your Salesforce Org](#).

If the DNS record for your domain points to a web application firewall (WAF), you can't use this option. If a third-party service or content delivery network (CDN) serves your domain, you can't use this option, even if the third party hosts your domain with your HTTPS certificate. In those cases, see [Use a Third-Party Service or CDN to Serve Your Custom Domain](#).

3. Upload your certificate.
 - a. If your CA uses intermediate certificates, see the instructions in the knowledge article, [Merge a complete certificate chain for custom HTTPS domains](#).
 - b. To upload your certificate to Salesforce, see [Generate a Certificate Signed by a Certificate Authority](#). You can get the required CA signature as a part of that process.
 - c. To import an existing certificate that is already signed, see the knowledge article, [Use HTTPS certificate that exists within your Community domain](#).

See Also

[Custom Domains](#)

[Set Up a Custom Domain That Uses Your HTTPS Certificate](#)

Set Up a Custom Domain That Uses Your HTTPS Certificate

Upload an HTTPS certificate that you own to Salesforce servers, then serve your custom domain with that certificate. After you set up your domain in Salesforce, that domain can serve your Experience Cloud sites or Salesforce Sites.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, and Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

USER PERMISSIONS NEEDED

To view a domain:

Manage Custom Domains

OR

View Setup and Configuration

To add a domain:

Manage Custom Domains

OR

View Setup and Configuration plus either a Site.com Publisher license or Create and Set Up Experiences

To edit or delete a domain:

Manage Custom Domains

To add, edit, and delete custom URLs:

Manage Custom Domains

OR

View Setup and Configuration AND either Create and Set Up Experiences OR a Site.com Publisher license

Unfamiliar with terms like DNS, CDN, and CNAME? Want to review the difference between a DNS resolver and a certificate? See [Custom Domain Terminology](#).

-  **Note** To create and activate a custom domain in a sandbox, add the domain in the production org that owns the sandbox.

1. Complete the [prerequisites](#) for this option.

With this option, Salesforce hosts your domain using your HTTPS certificate. If a third party hosts your domain, if you use a web application firewall (WAF), or if you use a third-party content delivery network (CDN), see [Use a Third-Party Service or CDN to Serve Your Custom Domain](#) instead.

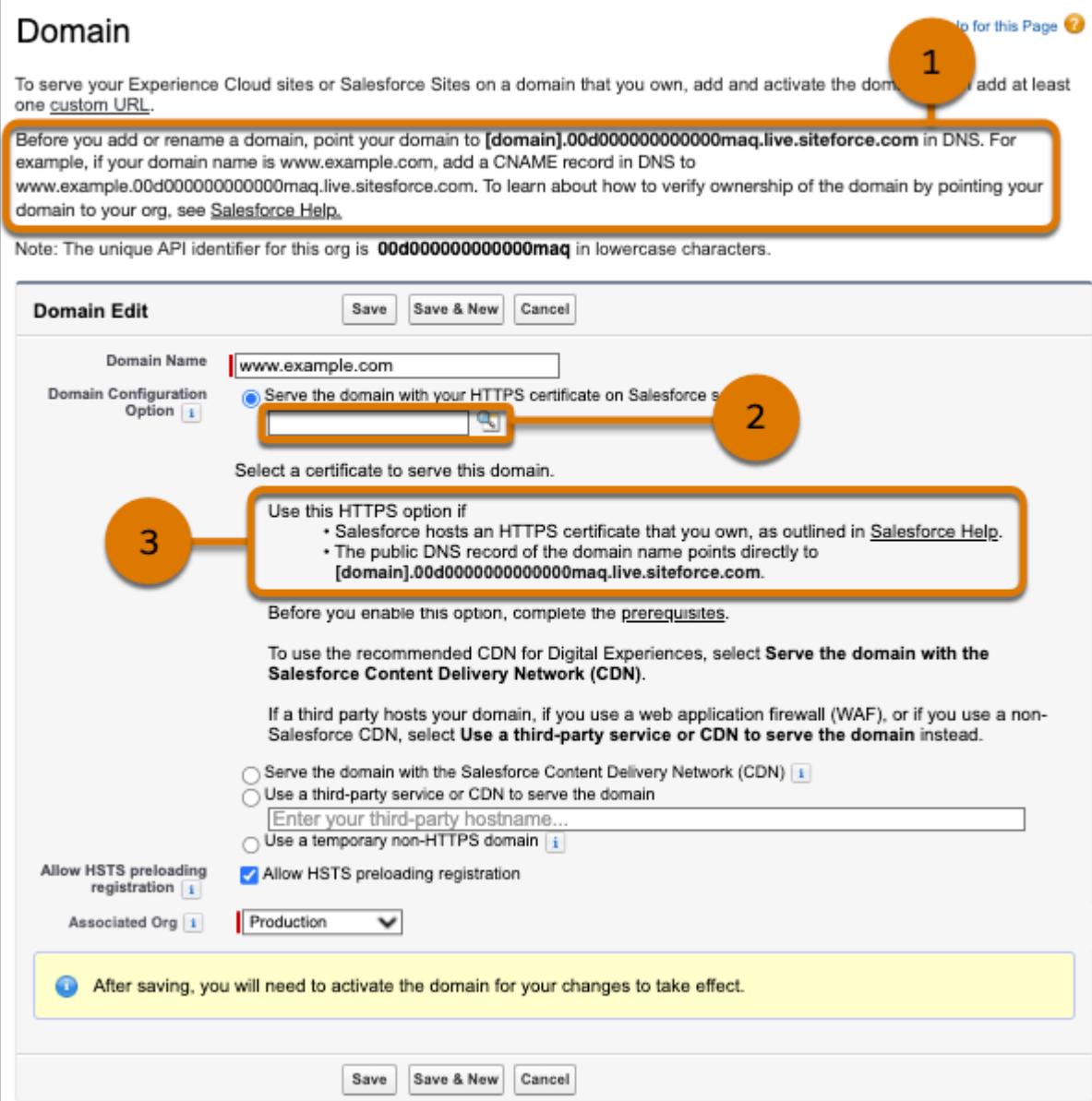
2. From Setup, in the Quick Find box, enter *Domains*, and then select **Domains**.
3. Click **Add a Domain**.
4. Enter the domain name.

Salesforce validates ownership based on the fully qualified domain name (FQDN) that you enter when you add a domain to your org. If you get an error message, [point your custom domain to your org](#), and then wait for the changes to propagate. After you update your domain's DNS record, it can take up to 20 hours for that change to take effect worldwide.

5. For Domain Configuration Option, select **Serve the domain with your HTTPS certificate on Salesforce servers**.

6. Click the lookup icon () , and then select your certificate.

Here's the Domain page when you select the domain configuration option to serve the domain with your HTTPS certificate.



The screenshot shows the 'Domain' page with the following annotations:

- Annotation 1:** A callout points to the top right of the page, containing the text "To serve your Experience Cloud sites or Salesforce Sites on a domain that you own, add and activate the domain. Add at least one custom URL." and a link to "Salesforce Help".
- Annotation 2:** A callout points to the 'Domain Configuration Option' section, which contains a radio button selected for "Serve the domain with your HTTPS certificate on Salesforce servers". Below this is a lookup field with a magnifying glass icon and the placeholder "Select a certificate to serve this domain".
- Annotation 3:** A callout points to a detailed description of the HTTPS option, which includes two bullet points: "Salesforce hosts an HTTPS certificate that you own, as outlined in [Salesforce Help](#)." and "The public DNS record of the domain name points directly to [domain].00d00000000000maq.live.siteforce.com".
- Bottom Callout:** A yellow callout at the bottom states: "After saving, you will need to activate the domain for your changes to take effect."

The top of the page includes your 18-digit org ID and the format for the canonical name (CNAME) in DNS to point your domain to your org (1). To specify the certificate for this domain, use the certificate field (2). When you select the option to serve your domain with your HTTPS certificate on Salesforce

servers, additional guidance includes the target host name to use for the CNAME record for your domain in DNS (3). Replace **[domain]** with your domain name, such as `www.example.com`.

7. If your domain is a registrable domain such as `https://example.com`, to avoid vulnerabilities during HTTP redirects, select **Allow HSTS preloading registration**.

This setting adds the preload directive to the HSTS header. After you enable this setting, submit your domain at <https://hstspreload.org>. For more information, including how to enable HSTS preloading for a domain with a subdomain, see [Enable HSTS Preloading on a Custom Domain](#).

8. For Associated Org, select the org from which you want this custom domain to serve site content.

For example, to serve the sites in your production org, select **Production**. Or select a sandbox where you want to test this custom domain.

This field only appears in production orgs with associated sandboxes. You can edit this field only from production orgs. For more information, see [Test Your Custom Domains in a Sandbox](#).

9. Save your domain.

The status of your domain changes to Awaiting Custom URL. For domains with this domain configuration option, provisioning starts when you add the first custom URL.

10. To serve your sites via your domain, [add a custom URL](#).

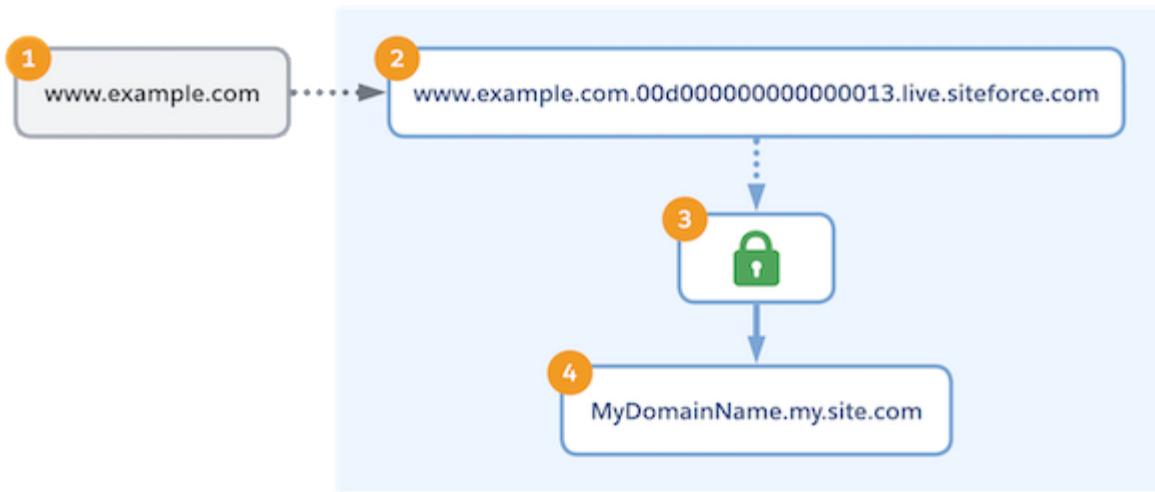
When you add the first custom URL for your domain, Salesforce provisions the domain or gets it ready to be used. The provisioning process can take 4–14 hours. When that process is complete, the domain's status on the Domains Setup page changes to Awaiting Activation and you receive an email.

Newly created custom domains use HTTP, not HTTPS, until you activate the domain.

11. To activate your domain, on the Domains Setup page, click **Activate** next to your custom domain name.

Your site can be unavailable for 5–10 minutes, so activate your custom domain when your site traffic is low. When your site is active, the status changes from Awaiting Activation to Completed and the Current Domain Configuration Option changes to Salesforce Cloud.

This diagram shows the connectivity between your custom domain and your Experience Cloud site content after you set up a custom URL to serve your site and activate your custom domain. Dotted lines () represent DNS configurations, and the solid line () represents user traffic flow through HTTPS. The gray line represents traffic that originates outside Salesforce, and the blue lines represent traffic that originates in Salesforce. In this example, the domain name is `www.example.com` and the 18-digit org ID is `00d000000000000013`.



In DNS, a CNAME record points your custom domain (1) to the Salesforce internal CNAME record for your org (2), which includes your org ID. In Salesforce, your certificate is stored on a secure server (3). Salesforce uses that certificate to serve the content from your Experience Cloud site (4).

See Also

[Custom Domains](#)

[Prerequisites for a Custom Domain That Uses Your HTTPS Certificate](#)

Serve Your Experience Cloud Site with the Salesforce Content Delivery Network (CDN)

Optimize page load times and site performance with our content delivery network. Salesforce partners with a content delivery network (CDN) provider to efficiently deliver publicly cacheable content to users on your Experience Cloud sites. We recommend the Salesforce CDN for custom domains that serve Digital Experiences, including Experiences built with Experience Cloud, Commerce, and Industries licenses.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, and Unlimited** Editions.

Applies to: [LWR, Aura, and Visualforce sites](#)

- Tip** To learn what content delivery networks are, how they work, and how they can help you improve your Experience Cloud site's performance and scalability, watch [Improve Performance and Scalability with Salesforce's Content Delivery Network](#).

[Content Delivery Networks \(CDNs\) and Salesforce](#)

Optimize page load times and site performance with our content delivery network (CDN). Salesforce

partners with a CDN provider to efficiently deliver publicly cacheable content to users on your Experience Cloud sites.

Considerations for the Salesforce CDN

If you host public-facing, cacheable content on your Experience Cloud sites, we recommend that you use the Salesforce content delivery network (CDN) to greatly improve load times.

Prerequisites for the Salesforce CDN

Before you set up a custom domain that uses the Salesforce content delivery network (CDN), complete these steps.

Set Up a Custom Domain That Uses the Salesforce CDN

You can set up the Salesforce content delivery network (CDN) for your custom domain in Experience Builder, Salesforce Tabs, and Visualforce sites. Within each production environment, you can define multiple domains in a sandbox, but only one custom domain associated with a sandbox org can use the Salesforce CDN.

Edit Settings for the Salesforce CDN

Fine-tune the Salesforce content delivery network's (CDN) default settings to address the unique needs of your storefront domains. Turn image optimization and the web application firewall (WAF) managed ruleset off or on. You can also protect your domain from malicious activity with up to 10 custom WAF rules based on autonomous system number (ASN), IP address, and country.

Traffic Allowances for the Salesforce CDN

Each byte of traffic that's requested, whether it's an image or a complex report, counts toward your content delivery network's (CDN) usage amount. The amount of traffic allowed over a CDN depends on the license that you purchase. Understand how to monitor your Salesforce CDN usage and what happens if you exceed the terabyte limit.

See Also

[Custom Domains](#)

[Options to Serve a Custom Domain](#)

Content Delivery Networks (CDNs) and Salesforce

Optimize page load times and site performance with our content delivery network (CDN). Salesforce partners with a CDN provider to efficiently deliver publicly cacheable content to users on your Experience Cloud sites.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, and Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

What's a CDN?

A content delivery network is a geographically distributed network of servers that store cached versions of web assets. To optimize page load times and site performance, a CDN efficiently delivers publicly cacheable content to users. CDNs are the industry standard for web applications because they provide faster and more secure content delivery.

All the assets used to develop your Experience Cloud sites or Salesforce Sites are stored on your Salesforce instance. Your instance is stored on one Salesforce server. The farther away your users are from that server, the longer it takes to get data to their computer and your site's pages.

The CDN minimizes delays in loading web page content by reducing the distance between the server and the user. CDNs also increase the number of requests that the server can respond to because the CDN offloads a large portion of hits to your site. By reducing the distance between the server and the user and increasing content availability and redundancy, CDNs improve website load times and site performance.

Learn more about CDNs and see one in action in this short video.

Watch the video: <https://play.vidyard.com/Pi9NhJTPUXeT9sNYZagXPB>

CDN Options for Sites

For sites, you can get the benefits of a CDN in three ways.

- Use the default Experience Cloud site domain. When you enable Experience Cloud sites, Salesforce creates a system-managed domain for your site. The host name for that system-managed domain ends in `*.my.site.com` and that domain uses the Salesforce CDN.
Government Cloud Plus orgs can opt out of the Salesforce CDN for `*.my.site.com` domains. To opt out, from Setup, in the Quick Find box, enter `My Domain`, and then select **My Domain**. Then in the Routing section, deselect **Use Content Delivery Network (CDN) by default when enhanced domains are enabled for Experience Cloud sites**.
- Set up a custom domain such as `https://www.example.com` that serves your Experience Cloud site content with the Salesforce CDN. This option isn't available for registrable domains. If you have Digital Experiences, including those built with `Experience Cloud`, `Commerce`, and `Industries` licenses, we recommend this option to serve your custom domain. For considerations, prerequisites, and set up instructions, see [Serve Your Experience Cloud Site with the Salesforce Content Delivery Network \(CDN\)](#).
- Set up a custom domain such as `https://www.example.com` that serves content from your Experience Cloud site or Salesforce Site with a third-party CDN outside of Salesforce. A non-Salesforce CDN is the only available option to serve a registrable domain or Salesforce Site via a CDN. Commerce LWR sites can't use a CDN outside of Salesforce. See [Use a Third-Party Service or CDN to Serve Your Custom Domain](#).



Note The Salesforce CDN is the content delivery network recommended for serving sites and stores. Even without a custom domain, you can get the benefits of a CDN. Go to [Enable the Lightning CDN](#)

[to Load Applications Faster](#) enable the Lightning CDN for your Salesforce org.

Who Can Use the Salesforce CDN for a Custom Domain

The Salesforce CDN is available for custom domains that serve Experience Cloud sites. If you use Marketing Cloud Account Engagement (Pardot) in a Professional Edition org, you must use the Salesforce CDN for your domains.

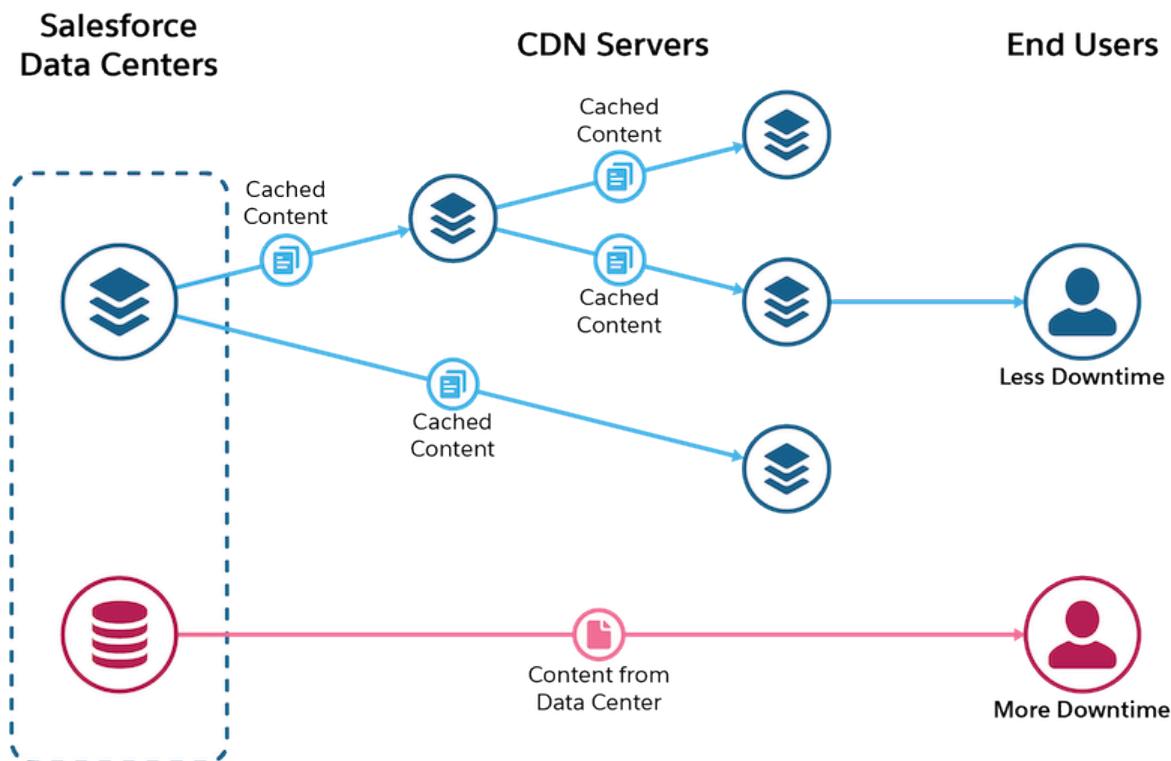
To get the benefits of a CDN for a domain that serves content from a Salesforce Site, use a CDN outside of Salesforce. Also, Salesforce is unable to serve a registrable domain, such as example.com, using our CDN. We only serve subdomains, such as www.example.com or parts.example.com. To serve your site content on a registrable domain served from a CDN, host it on a CDN outside of Salesforce.

If you're creating a Commerce LWR store, you can use the system-managed *.my.site.com URL or a custom domain to enable the Salesforce CDN.

How a CDN Works

By reducing the distance between the server and the user and increasing content availability and redundancy, CDNs improve website load times and [site performance](#).

This diagram illustrates this effect for the Salesforce CDN for Experience Cloud sites. The principle also applies to external CDNs and the Lightning CDN for Salesforce orgs.



The Salesforce CDN includes an [image optimization](#) feature that makes your site pages load faster for guest users, whether they're viewing your site on their phone, tablet, or desktop computer. In addition, the Salesforce CDN comes with configurable availability pages. You can display a custom [Service Not Available page](#) when your site is down or a custom [Too Many Requests page](#) when your site is in high demand.

See Also

[Custom Domains](#)

[Serve Your Experience Cloud Site with the Salesforce Content Delivery Network \(CDN\)](#)

[Use a Third-Party Service or CDN to Serve Your Custom Domain](#)

Considerations for the Salesforce CDN

If you host public-facing, cacheable content on your Experience Cloud sites, we recommend that you use the Salesforce content delivery network (CDN) to greatly improve load times.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: [LWR, Aura, and Visualforce sites](#)

Unfamiliar with terms like DNS, WAF, and CNAME? Want to review the difference between a URI and a URL? See [Custom Domain Terminology](#).

Before you enable our CDN, keep these considerations in mind.

Recommended Option

The Salesforce CDN is the recommended configuration option for custom domains that serve Digital Experiences, including Experiences built with [Experience Cloud](#), [Commerce](#), and [Industries](#) licenses. If the Salesforce CDN option isn't available, make sure that [Digital Experiences](#) is enabled for your org.

-  **Note** The Salesforce CDN is the content delivery network recommended for serving sites and stores. Go to [Enable the Lightning CDN to Load Applications Faster](#) enable the Lightning CDN for your Salesforce org.

The Salesforce CDN Partners

Salesforce uses a CDN partner service to optimize content delivery. All information that's sent to or returned by a domain is stored and transmitted through our CDN.

-  **Note** The Salesforce CDN isn't recommended for sites in China or sites with Chinese users. We recommend using [your own CDN](#) with a local Chinese CDN provider. Please reach out to Salesforce Support for recommendations and work with your local CDN provider on the configuration, performance, and regulatory considerations of your site.

Supported Environments

In each production environment, you can define multiple sandbox custom domains.

Supported Domain Levels and Sites

When you use our CDN, Salesforce is unable to serve a registrable domain, such as `example.com`. The Salesforce CDN only serves subdomains, such as `www.example.com` or `parts.example.com`. If your site needs a registrable domain served from a CDN, host it on a CDN outside of Salesforce.

Likewise, the Salesforce CDN is available for custom domains that serve Experience Cloud site content. To use a CDN to serve content from your Salesforce Site, use a CDN outside of Salesforce. Commerce LWR sites can't use a CDN outside of Salesforce.

Opt Out of the Salesforce CDN

Government Cloud Plus orgs can opt out of the Salesforce CDN for your system-managed `*.my.site.com` domain. To opt out, go to Setup, then in the Quick Find box, enter `My Domain`, and then select **My Domain**. Then in the Routing section, deselect **Use Content Delivery Network (CDN) by default when enhanced domains are enabled for Experience Cloud sites**.

You can also disable your site's Experience Cloud content delivery network (CDN) on your own in sandbox orgs and see if there are any negative impacts before it's disabled in your production org. To test how disabling the CDN for enhanced domains affects your org, from My Domain Settings of your sandbox org, deselect **Use Content Delivery Network (CDN) by default when enhanced domains are enabled for Experience Cloud Sites**.

Activation Timing

To minimize the impact to your users, activate the Salesforce CDN when your site traffic is low.

HTTP/2 Support

The Salesforce CDN supports HTTP/2 protocols. Experience Cloud sites served through our CDN support HTTP/2 for fast content delivery. Salesforce also provides a secure HTTPS site and certificates.

URI Size Limit

The maximum URI size limit for our CDN is 16 kilobytes. If you must exceed this amount, disable the Salesforce CDN.

Provisioning a Domain for Salesforce CDN

Domain provisioning is the process of setting up and configuring a domain name on the internet. In other words, when Salesforce provisions your custom domain, we get it ready for use. Provisioning usually takes fewer than 12 hours, but it can take up to 24 hours to complete. After provisioning, activate your Salesforce domain to create or update the domain canonical name (CNAME). Activation across the internet can take up to 20 minutes.

Until you activate the domain, the domain uses its previous HTTPS configuration. If you add a domain to your org using our CDN, the initial configuration before activation is HTTP.

Provisioning a Domain for a Third-Party CDN

If you use your own CDN and add a domain and provision the CDN at the same time, visitors to your site can see a certificate error. The host name on the certificate doesn't match the custom domain until you activate the domain. Activating the domain resolves the error message.

Until you activate the domain, the domain uses its previous HTTPS configuration.

Switching to the Salesforce CDN

When you change the HTTPS option on an existing domain to the Salesforce CDN, the provisioning process usually takes fewer than 12 hours, but it can take up to 24 hours. After you activate the domain, visitors to your site can see errors for up to 5 minutes:

- Connection reset errors, such as “The site doesn’t load.”
- DNS error messages like “Server DNS address couldn’t be found.”

Caching with the Salesforce CDN

Caching on our CDN improves your site's performance and scale. When your users access a site served by the Salesforce CDN, cached content is served directly from CDN servers. CDN servers are distributed globally and are often closer to your users than Salesforce servers. Because cached content is served directly from CDN servers, your users experience faster load times routinely and in times of high traffic. CDN caching can work with [browser-side caching](#), which also improves performance.

The Aura, LWR, and Visualforce frameworks for building Experience Cloud sites cache content on CDN slightly differently. But no matter which framework you use to build your site, consider caching only public content on our CDN.

Aura caches mostly static content, including Salesforce CMS content, images, javascript, CSS files, font files, and more. In comparison, Lightning Web Runtime (LWR) is built for performance and scale, and it caches more publicly available content in addition to static resources. In LWR, Salesforce caches base documents for a site's pages, public data returned from Salesforce API calls, and public data from Apex methods used internally. Admins and developers can cache their Apex methods that return public data..

Single Versus Shared Server Certificates

We recommend single domain certificates over shared certificates. Shared certificates often include other customer host names in the subject alternative name list. The server certificate for shared certificates also changes frequently, which can cause issues for API clients that require the exact server certificate rather than the root certificate authority. Single certificates offer better security and a single branded experience for your site.

Shared certificates are no longer available for new sites using the Salesforce CDN. If you have an existing custom domain on a shared certificate, you can see a shared option while configuring our CDN. To switch from a shared certificate to a single certificate, on the Domain Edit page, select the **Single certificate for content delivery network (CDN)**. There's no downtime when you switch from a shared certificate to a single certificate.

Orgs that purchase Experience Cloud licenses and use a third-party CDN service get 10 Experience Cloud CDN single certificate domains and 48 terabytes (TB) of traffic per year. Orgs that host LWR Commerce sites or sites hosted on Experience Delivery and use our CDN partne get 5 branded certificates and no limit on traffic per year. If you haven't purchased an Experience Cloud license, your org can provision 5 single certificate domains and get 5 TB of annual traffic. You can contact your account executive to increase the traffic allowances. To increase the number of certificates available per org, contact Salesforce Customer Support.

Security Services Included with the Salesforce CDN

Web application firewall and rate-limiting security features are included for sites that use the Salesforce CDN with single domain certificates. These features improve security by filtering out bad traffic. Focusing on valid traffic improves your site's performance for your customers.

The web application firewall (WAF) catches and rejects attempts to compromise your system. For example, the firewall catches attempts to use SQL injection, command injection, and cross-site scripting. If someone attempts to put malicious content in your site, the firewall filters and rejects the request.

Rate limiting (RL) monitors and blocks abnormal increases in traffic. If the number of requests suddenly increases by a certain threshold, rate limiting blocks the requests. Rate limiting also blocks unusual increases in requests within a time period, for example, a week.

If you notice that your site isn't performing as well or that valid requests are getting blocked, on the Edit Domain page, deselect the **Use the Salesforce CDN's enhanced security services** checkbox.

Domain CNAME Changes

Your domain must use the same CNAME that shows on the Domains Setup page. Changing your CNAME after activating the Salesforce CDN with a shared certificate causes your domain traffic to go directly to your servers. It no longer passes through the CDN.

If you want to resume use of the Salesforce CDN after changing your CNAME, update your domain in Salesforce to the correct CNAME. After you update, provision the domain again with the CDN option.

-  **Note** Don't change single certificate domains with the _acme-challenge CNAME if the single certificate is provisioning or has finished provisioning. If you update or delete the _acme-challenge CNAME before provisioning is complete, it can delay the provisioning process. If it's updated after provisioning the domain, you can have issues when the certificate is due for renewal.

CDN Changes and Single Sign-On

Changing the Salesforce CDN affects SAML Single Sign-On Settings for all custom URLs in that domain. Reconfirm the SAML Single Sign-On Settings for each HTTPS custom URL in that domain after activating a change. Login Settings are available in [Experience Workspaces | Administration | Login & Registration](#).

Data Privacy & Security

When you enable the Salesforce CDN, information sent to or returned by the domain is stored and transmitted by our CDN partner service. For Aura sites, cached content includes static content such as HTML pages, javascript and CSS files, images, and font files.

For LWR sites, cached content includes static content and other publicly cacheable content. Where appropriate, Salesforce caches public content from APIs and Apex methods in standard pages and components. Admins and developers can control caching of Apex methods that use @wire invocation in LWC for their guest users. Only Apex methods annotated with @AuraEnabled(cacheable=true scope='global') are cached. Caching public data from annotated Apex methods in managed packages is enabled by default. To disable this preference, go to [Experience Workspaces | Administration | Preferences](#) and deselect **Cache public data from Apex methods in Managed Packages**.

Our CDN partner manages privacy and security protections for data that is shared. All communications between our CDN partner and Salesforce are conducted over HTTPS.

Our CDN partner supports IPv6. If IP restrictions are configured in Salesforce with only IPv4 addresses, users can see an error when they access your site that ends in *.my.site.com via IPv6. To prevent that error, update your IP allowlists or restrictions to allow IPv6 source addresses for authorized users. In particular, review and update the login IP range restrictions for the relevant profiles, including the site's guest user profile. For more information on setting IP restrictions for Salesforce, see [Network Access, Session Settings, and Profile-based IP restrictions](#).

Image Optimization

The Salesforce CDN includes an [image optimization](#) feature that makes your site's pages load faster for guest users, whether they're viewing your site on their phone, tablet, or desktop computer. The image optimization setting is enabled by default for sites that use the Salesforce CDN. To disable this preference for orgs using our CDN, go to Setup. In the Quick Find box, enter *Domains*, and then select **Domains**. Select **Edit CDN Settings** next to the domain whose settings you want to edit. Toggle off Image Optimization.

See Also

[Custom Domains](#)

[Serve Your Experience Cloud Site with the Salesforce Content Delivery Network \(CDN\)](#)

Prerequisites for the Salesforce CDN

Before you set up a custom domain that uses the Salesforce content delivery network (CDN), complete these steps.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: [LWR, Aura, and Visualforce sites](#)

 **Tip** To learn what content delivery networks are, how they work, and how they can help you improve your Experience Cloud site's performance and scalability, watch  [Improve Performance and Scalability with Salesforce's Content Delivery Network](#).

Unfamiliar with terms like DNS, CAA, and CNAME? See [Custom Domain Terminology](#).

Be sure to read the [Considerations for Using a CDN for Your Experience Cloud Site](#).

Verify the Canonical Name (CNAME) Records for Your Domain

Before you can set up your custom domain with the Salesforce CDN, update DNS to include the required canonical name (CNAME) records for your domain and the _acme-challenge subdomain. See [Point Your Custom Domain to Your Org](#).

Manage Certificate Authority Authorization (CAA) Records

The Salesforce CDN uses Let's Encrypt as its certificate authority. Before you set up our CDN for your domain:

- If your domain doesn't have a current Certification Authority Authorization (CAA) record, leave it as is.
- If your domain or the subdomain has a CAA record, add `letsencrypt.org` to the record.
- If your company decides to set up a CAA record in the future, add `letsencrypt.org` to the record to avoid disruption to your CDN.

For more information, see the [Let's Encrypt](#) website.

 **Note** Let's Encrypt requires consent from the owner of high-value domains before allowing a CAA certificate change for those domains. The list of high-value domains isn't publicly available. If your domain is on the list of high-value domains, contact Let's Encrypt to temporarily remove the block before configuring the Salesforce CDN. If your CDN request fails or it remains in a status of provisioning for more than a day, contact Salesforce Customer Support for assistance.

Use an Existing External Domain

Salesforce uses a partner service to optimize content delivery. All information that's sent to or returned by a domain is stored and transmitted through the CDN service.

If you have an existing domain hosted outside of Salesforce, such as `www.example.com`, and that domain isn't currently registered with the Salesforce CDN partner that will serve your domain, moving to the Salesforce CDN is a two-step process. First, set up a custom domain that points to our CDN partner as a third-party CDN. Commerce LWR sites can't use a CDN outside of Salesforce. See [Use a Third-Party Service or CDN to Serve Your Custom Domain](#).

Then, after you activate the custom domain and your domain successfully serves content from your Experience Cloud site, [change the HTTPS option](#) for that custom domain to the Salesforce CDN.

See Also

[Custom Domains](#)

[Serve Your Experience Cloud Site with the Salesforce Content Delivery Network \(CDN\)](#)

[View and Edit Single Sign-On Settings](#)

Set Up a Custom Domain That Uses the Salesforce CDN

You can set up the Salesforce content delivery network (CDN) for your custom domain in Experience Builder, Salesforce Tabs, and Visualforce sites. Within each production environment, you can define multiple domains in a sandbox, but only one custom domain associated with a sandbox org can use the Salesforce CDN.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: [LWR, Aura, and Visualforce sites](#)

USER PERMISSIONS NEEDED

To create an Experience Cloud site: Create and Set Up Experiences AND View Setup and Configuration

To customize an Experience Cloud site: Be a member of the site AND Create and Set Up Experiences

OR

Be a member of the site AND an experience admin, publisher, or builder in that site

To publish an Experience Cloud site: Be a member of the site AND Create and Set Up Experiences

OR

Be a member of the site AND an experience admin or publisher in that site

To add a domain: Manage Custom Domains

OR

View Setup and Configuration plus either a Site.com Publisher license or Create and Set Up Experiences

To edit a domain: Manage Custom Domains

To add, edit, and delete custom URLs: Manage Custom Domains

OR

View Setup and Configuration AND either Create and Set Up Experiences OR a Site.com Publisher license

Before you add a custom domain that serves your Experience Cloud site with the Salesforce CDN, review these important considerations.

- To minimize disruption for your users, provision and activate our CDN when your site traffic is low.
- To create or activate a custom domain for testing in a sandbox, log in to the production org that owns the sandbox and go to the Domains Setup page.

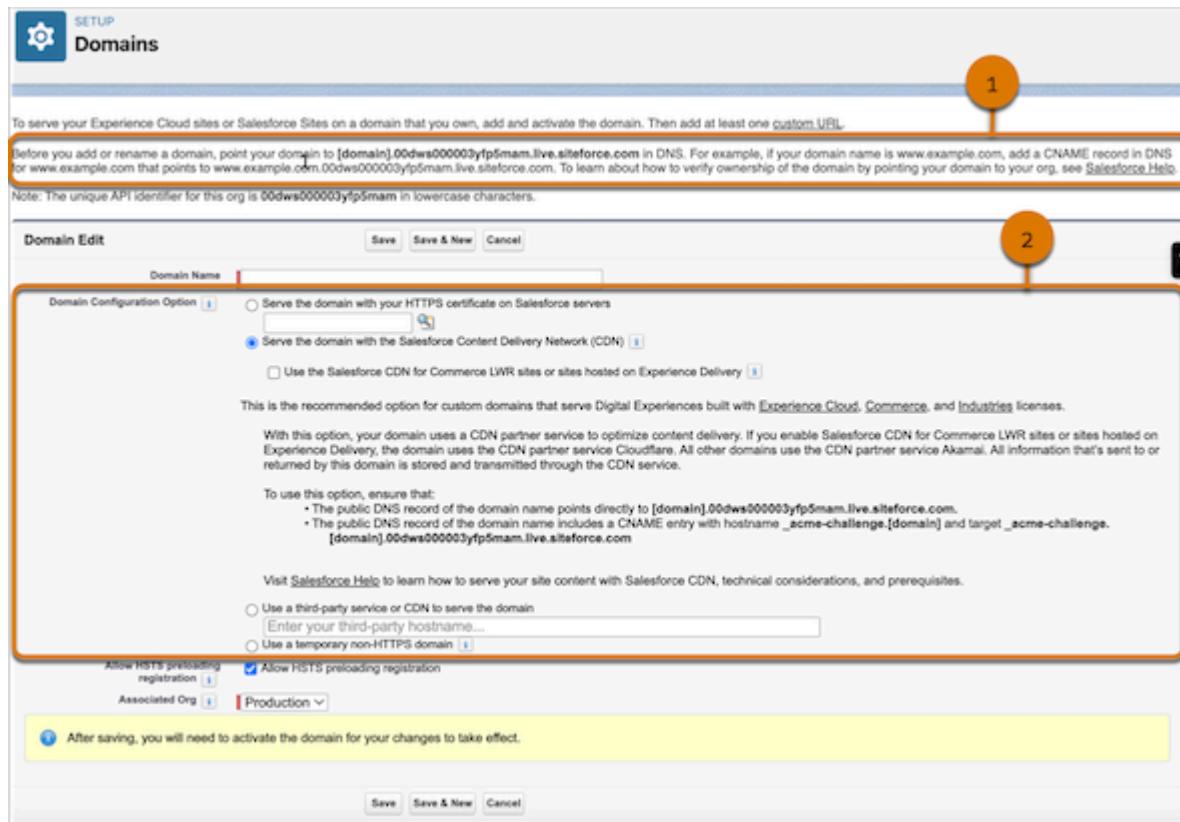
- With this option, Salesforce hosts your domain with one of Salesforce's CDN partners. If a third party hosts your domain or if you use a third-party CDN or a web-application firewall (WAF), see [Use a Third-Party Service or CDN to Serve Your Custom Domain](#).
- When updating Domain Configuration Options for a domain, avoid selecting **Serve the domain with the Salesforce Content Delivery Network (CDN)**, then selecting the checkbox **Use the Salesforce CDN for Commerce LWR sites or sites hosted on Experience Delivery** more than 2 times in one week as it may cause domain onboarding delays.
- Salesforce partners with CDN providers to optimize our content delivery network. If you have an existing domain hosted outside of Salesforce, such as `www.example.com`, and that domain isn't currently registered with the Salesforce CDN partner that will serve your domain, moving to the Salesforce CDN is a two-step process. First, set up a custom domain that points to our CDN partner as a third-party CDN. See [Use a Third-Party Service or CDN to Serve Your Custom Domain](#). Then [change the HTTPS option](#) for that custom domain to the Salesforce CDN.
- This option is unavailable for Salesforce Sites and for registrable domains, such as `example.com` without the `www` subdomain. To serve a registrable domain or a Salesforce Site with a CDN, see [Use a Third-Party Service or CDN to Serve Your Custom Domain](#).
- You don't have to create a custom domain to test the Salesforce CDN in a sandbox. You can also use the system-managed `*.sandbox.my.site.com` Experience Cloud URL, which uses our CDN partner, to serve your content with the Salesforce CDN in sandboxes.

Before you activate this feature, read the [considerations](#) and complete the [prerequisites](#) for the Salesforce CDN. Then complete the [custom domain prerequisites](#).

- On your DNS provider's site, verify that the two required canonical name (CNAME) records exist for your domain and the `_acme-challenge` subdomain. See [Point Your Custom Domain to Your Salesforce Org](#).
- From Setup, in the Quick Find box, enter `Domains`, and then select **Domains**.
- Select **Add a Domain**.
- Enter the domain name.
Salesforce validates ownership based on the fully qualified domain name (FQDN) that you enter when you add a domain to your org. If you get an error message, [point your custom domain to your org](#), and then wait for the changes to propagate. After you update your domain's DNS record, it can take up to 20 hours for that change to take effect worldwide.
- For orgs launching Commerce LWR sites or sites hosted on Experience Delivery, select the Domain Configuration option, **Serve the domain with the Salesforce Content Delivery Network (CDN)**, then select the checkbox **Use the Salesforce CDN for Commerce LWR sites or sites hosted on Experience Delivery**. With this option, your domain uses a single certificate which displays only one host name. Five branded certificates are available for use with the Salesforce CDN partner and unlimited traffic are available. Contact your account representative if you need additional certificates. You can't use the same domain name across multiple organizations. If you delete a domain name from one organization and plan on reusing it in another, wait at least 48 hours between deleting the domain name and adding it back in.
- For all other digital experiences using Experience Cloud, Commerce, or Industries licenses, select the Domain Configuration Option **Serve the domain with the Salesforce Content Delivery Network (CDN)**. With this option, your domain uses a single certificate, which displays only one host name. Ten

branded certificates for use with the Salesforce CDN partner and 48 terabytes of traffic per year are available for Experience Cloud licenses that adopt a single certificate CDN. If you need more certificates, contact your account representative.

Here's the Domain page when you select this domain configuration option.



The top of the page includes your 18-digit org ID and the format for the CNAME in DNS to point your domain to your org (1). When you select the option to serve your domain with the Salesforce CDN, additional guidance includes the targets for both CNAME records in DNS (2). Replace `[domain]` with your domain name, such as `www.example.com`.

7. For Associated Org, select the org from which you want this custom domain to serve site content.

For example, to serve the sites in your production org, select **Production**. Or select a sandbox where you want to test this custom domain.

This field only appears in production orgs with associated sandboxes. You can edit this field only from production orgs. For more information, see [Test Your Custom Domains in a Sandbox](#).

8. Save your domain.

After you save your new domain, Salesforce provisions the domain or gets it ready to be used. The Salesforce CDN provisioning process can take up to 24 hours. When that process is complete, the domain's status on the Domains Setup page changes from Provisioning to Awaiting Activation and you receive an email.

Newly created custom domains use HTTP, not HTTPS, until you activate the domain.

- To activate your domain, on the Domains Setup page, select **Activate** next to your custom domain name.

 **Note** Custom domains in a sandbox are edited and activated in production.

Your site can be unavailable for 5–10 minutes, so activate your custom domain when your site traffic is low. When your site is active, the status changes from Awaiting Activation to Completed, and the Current Domain Configuration Option changes to the Salesforce CDN partner.

- To serve your sites via your activated domain, [add a custom URL](#).

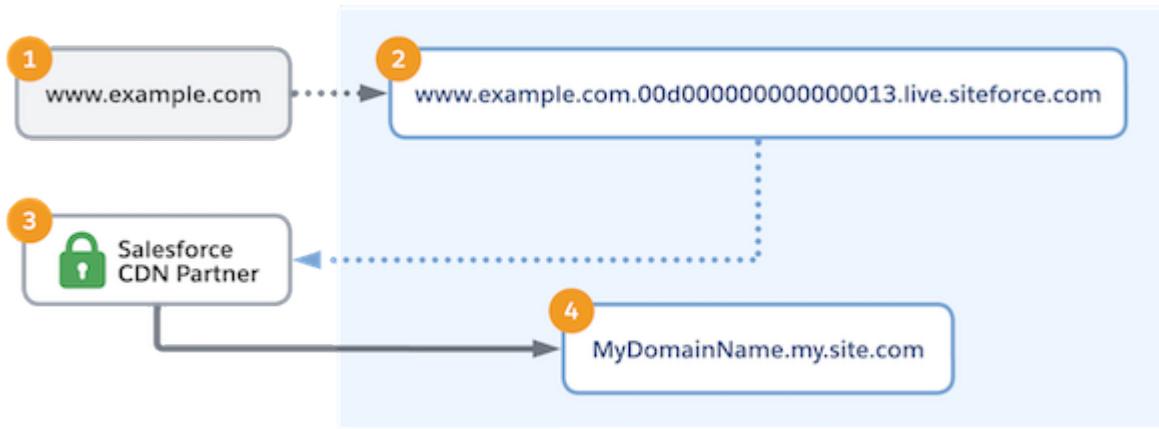
 **Note** For your domain to serve your site, at least one custom URL is required.

Changing the Salesforce CDN affects SAML Single Sign-On Settings for all custom URLs in that domain. Reconfirm the SAML Single Sign-On Settings for each HTTPS custom URL in that domain after activating a change. Login Settings are available in **Experience Workspaces | Administration | Login & Registration**.

- To avoid vulnerabilities during HTTP redirects, [enable HTTP Strict Transport Security \(HSTS\) preloading](#) on the registrable domain for your custom domain.

HSTS preloading ensures that HTTPS connections are always used in supported browsers. Configure this option by adding an HTTP header on the registrable domain for your custom domain. For example, if your custom domain is https://shop.example.com, you add the header to https://example.com. Because the Salesforce CDN doesn't support registrable domains, Salesforce can't configure this header for you, and the Allow HSTS preloading registration option has no effect.

This diagram shows the routing of traffic when Salesforce serves your custom domain with the Salesforce CDN. Dotted lines (••••➡) represent DNS configurations, and the solid line (➡) represents user traffic flow through HTTPS. The gray lines represent traffic that originates outside Salesforce, and the blue line represents traffic that originates in Salesforce. In this example, the domain name is www.example.com and the 18-digit org ID is 00d000000000000013.



With your DNS provider, you point your custom domain (1) to the Salesforce internal CNAME (2), which

includes your org ID. Within Salesforce, user traffic is routed to the Salesforce CDN partner (3), which acts as an intermediary for your Salesforce content (4).

See Also

[Custom Domains](#)

[Serve Your Experience Cloud Site with the Salesforce Content Delivery Network \(CDN\)](#)

[View and Edit Single Sign-On Settings](#)

Edit Settings for the Salesforce CDN

Fine-tune the Salesforce content delivery network's (CDN) default settings to address the unique needs of your storefront domains. Turn image optimization and the web application firewall (WAF) managed ruleset off or on. You can also protect your domain from malicious activity with up to 10 custom WAF rules based on autonomous system number (ASN), IP address, and country.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: [LWR sites](#)

Unfamiliar with terms like DNS, WAF, and CNAME? Want to review the difference between a URI and a URL? See [Custom Domain Terminology](#).

To customize your domain's Salesforce CDN settings, go to Setup. In the Quick Find box, enter *Domains*, and then select **Domains**. Click **Edit CDN Settings** next to the domain whose settings you want to edit.

-  **Note** The Salesforce CDN Settings page is available for Commerce LWR stores sites hosted on Experience Delivery that are provisioned with the Salesforce CDN partner. From the Domains page, the Current Domain Configuration Option column identifies which Salesforce CDN partner serves your domain.

Image Optimization

The image optimization feature makes your site pages load faster for guest users, whether they're viewing your site on their phone, tablet, or desktop computer. For the best site performance, turn on Image Optimization.

Compress Content and Web Resources

Salesforce uses Brotli compression to compress content and web resources for faster page-load times and a smoother delivery. Brotli is a compression algorithm created by Google that can reduce file sizes without sacrificing any significant information.

Accelerate and Secure HTTP Traffic

Quick UDP Internet Connections (QUIC) is a transport layer network protocol designed by Google to improve the performance of web applications over the internet. It's encrypted by default and built on UDP instead of TCP, which makes it more reliable, secure, and faster than traditional internet protocols.

Web Application Firewall (WAF)

A web application firewall protects web applications for potential attacks, such as cross-site scripting (XSS), injection attacks, sensitive data exposure, and security misconfigurations. It analyzes each HTTP or HTTPS request at the application layer.

WAF rules use information about each site visitor to determine whether it's safe to grant the visitor access to your site. With potentially malicious site traffic filtered out by the Salesforce CDN, your site has more bandwidth and better performance to support valid traffic.

Managed Ruleset

Managed Ruleset is a collection of preconfigured WAF rules that address common security vulnerabilities and protect your site from malicious attacks. The Salesforce CDN partner updates the managed ruleset frequently to address new security threats and to reduce false positives. When you turn on Managed Ruleset, your sites are automatically protected with the most up-to-date security.

Custom Firewall Rules

For more control over incoming site traffic, create up to 10 unique WAF rules. To create each rule, write an expression that includes an operator and a value. Then, specify the action for the Salesforce CDN to take when incoming traffic matches that condition—allow the traffic, block the traffic, or run a managed challenge. Custom WAF rules support simple and complex expressions. See [Custom Firewall Rule Expressions on the Salesforce CDN](#).

[Custom Firewall Rule Expressions on the Salesforce CDN](#)

Custom web application firewall (WAF) rule expressions evaluate incoming traffic based on IP address, autonomous system number (ASN), or country. After you write an expression, choose whether to allow, block, or run a managed challenge on incoming traffic that matches the expression.

Custom Firewall Rule Expressions on the Salesforce CDN

Custom web application firewall (WAF) rule expressions evaluate incoming traffic based on IP address, autonomous system number (ASN), or country. After you write an expression, choose whether to allow, block, or run a managed challenge on incoming traffic that matches the expression.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, and Unlimited** Editions.

Applies to: [LWR sites](#)

To write expressions for custom WAF rules, go to the Salesforce CDN Settings page.

The screenshot shows the 'Salesforce CDN Settings' page under the 'DOMAINS > SALESFORCE.EXAMPLE.COM' section. The page is divided into several sections:

- Optimization**: Includes 'Image Optimization' (switched On), which optimizes images based on device type and screen resolution.
- Speed**: Includes 'Compress Content and Web Resources' (switched On), which uses Brotli to compress content and web resources for faster page-load times.
- Accelerate and Secure HTTP Traffic**: Includes 'QUIC' (switched On), which improves performance, security, and reliability by using the QUIC transport layer protocol.
- Web Application Firewall (WAF)**: Includes 'Managed Ruleset' (switched On) and 'Custom Firewall Rules'. The 'Custom Firewall Rules' section allows users to inspect and filter incoming traffic with up to 10 simple or complex rules. A 'New Rule' button is available to add new rules. Below this section, a note states: "After you create a rule, it appears here."

Note The Salesforce CDN Settings page is available for Commerce LWR stores or sites hosted on Experience Delivery that are provisioned with the Salesforce CDN partner. From the Domains page, the Current Domain Configuration Option column identifies which Salesforce CDN partner serves your domain.

From the Salesforce CDN Settings page, click **New Rule**, and write a simple or complex expression in the Condition field. A simple expression defines a value against which incoming traffic is evaluated. A complex expression combines one or more simple expressions to create more focused evaluations for incoming traffic.

A simple expression's syntax has three components.

```
<field> <comparison_operator> <value>
```

Example To allow traffic from a group of specific IP addresses, use the expression `ip.src in { 203.0.113.0 203.0.113.1 }`, and select **Allow** from the Action dropdown menu. In this

example, `ip.src` is the field, `in` is the comparison operator, and `{ 203.0.113.0 203.0.113.1 }` is the value.

To write a complex expression, combine one or more simple expressions using one of the logical operators `and`, `or`, or `not`.

```
<expression> <logical_operator> <expression>
```

-  **Example** To block traffic if the country is Germany and the visitor has a specific IP address, use the expression `(ip.src.country eq "DE") and (ip.src eq 1.1.1.1)`, and then from the Action dropdown menu, select **Block**.

Be careful not to create complex expressions that are too restrictive. Overly restrictive custom WAF rules can unintentionally block a large number of potential visitors from accessing your site.

Fields

The field specifies which of the incoming traffic's properties to evaluate. For example, if you want to allow traffic from a certain country, use the `ip.src.country` field in your custom WAF rule.

Field	Description
<code>ip.src.asnum</code>	Represents the integer representing the autonomous system number (ASN) associated with the site visitor's IP address.
<code>ip.src.country</code>	Represents the two-letter country code in ISO 3166-1 Alpha 2 format.
<code>ip.src</code>	Represents the IP packet's source address.

Comparison Operators

The comparison operator defines how values must relate to actual request data for an expression to remain true.

Operator Name	Operator Notation
Equal	<code>eq</code>
Not equal	<code>ne</code>
Value is within a set of values	<code>in</code>

-  **Note** When you use the `in` operator, surround the value with curly brackets, and use a space to separate the items within the value. For example, `ip.src in { 203.0.113.0 203.0.113.1 }`.

Values

The value represents the data associated with a field. To evaluate a rule, the value is compared with the data from the incoming traffic request.

The Salesforce CDN supports valid values for the `ip.src.asnum`, `ip.src.country`, and `ip.src.fields`.

Traffic Allowances for the Salesforce CDN

Each byte of traffic that's requested, whether it's an image or a complex report, counts toward your content delivery network's (CDN) usage amount. The amount of traffic allowed over a CDN depends on the license that you purchase. Understand how to monitor your Salesforce CDN usage and what happens if you exceed the terabyte limit.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, and Unlimited** Editions.

Applies to: [LWR, Aura, and Visualforce sites](#)

Within each production environment, you can define multiple domains in a sandbox, but only one custom domain associated with a sandbox org can use the Salesforce CDN.

Traffic Allowances by License Type

If you have any of these items, you're entitled to 48 terabytes of traffic per year and 10 single certificate domains.

- Salesforce CMS, Commerce Apps, or External Apps
- A Customer Community license, a Customer Community Plus license, or a Partner Community license

If you require more traffic for your org, contact your account representative to purchase additional terabytes. If you exceed the traffic allowance, a representative contacts you to discuss purchasing more terabytes for your business needs. To request more single certificate domains, you can log a ticket with Salesforce Customer Support.

 **Note** If you exceed the traffic limit, we don't shut down your site or move remove your domain from our CDN.

CDN Usage Reports

You can use two reports to check your Salesforce CDN usage. A CDN Usage report is part of the

Experience Management Package found in AppExchange. You can also create a custom report type with Domains as the primary object.

If you need help with managing your usage, contact Salesforce Customer Support.

See Also

[Custom Domains](#)

[Serve Your Experience Cloud Site with the Salesforce Content Delivery Network \(CDN\)](#)

[Create a Custom Report Type](#)

Use a Third-Party Service or CDN to Serve Your Custom Domain

Use a domain that's hosted by a third-party service to serve your Experience Cloud sites or Salesforce Sites. These third-party services include third-party hosts, web application firewalls (WAFs), and non-Salesforce content delivery networks (CDNs).

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

[Considerations for Custom Domains That Use a Third-Party Service or CDN](#)

Review how IP address tracking works when a third party serves your custom domain. Understand what happens if you update your domain's external configuration after you activate the domain in Salesforce, and clean up unnecessary Domain Name Service (DNS) records.

[Prerequisites for a Custom Domain That Uses a Third-Party Service or CDN](#)

If a non-Salesforce service or content delivery network (CDN) serves your domain, complete these steps before you add your domain in Salesforce. To confirm your domain ownership, update your domain's Domain Name Service (DNS) configuration. Then work with the third-party provider to ensure access to the required HTTP protocols, configure caching and request handling, and review reverse proxy restrictions. If you're using a third-party CDN, update the header in your CDN to track IP addresses.

[Set Up a Custom Domain That Uses a Third-Party Service or CDN](#)

Add a domain in Salesforce that's hosted by a non-Salesforce service or content delivery network (CDN). After you set up your domain in Salesforce, that domain can serve your Experience Cloud sites or Salesforce Sites.

See Also

[Custom Domains](#)

[Options to Serve a Custom Domain](#)

Considerations for Custom Domains That Use a Third-Party Service or CDN

Review how IP address tracking works when a third party serves your custom domain. Understand what happens if you update your domain's external configuration after you activate the domain in Salesforce, and clean up unnecessary Domain Name Service (DNS) records.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, and Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

 **Tip** Unfamiliar with terms like DNS, WAF, and CNAME? Want to review the difference between a DNS resolver and a certificate? See [Custom Domain Terminology](#).

IP Address Tracking and Restrictions

When you use a third party to serve your domain, Salesforce uses and logs the source IP address from the TCP/IP connection to Salesforce.

The `True-Client-IP` request header is used for location-targeting features within an Experience Cloud site. If you use an external content delivery network (CDN) and location-based audience targeting in Experience Cloud, set this header in your CDN. For more information, see [Prerequisites for a Custom Domain That Uses a Third-Party Service or CDN](#).

IP restrictions at the profile level ignore the `True-Client-IP` request header for custom domains served by an external service or CDN. To restrict your guest users' access to valid IP addresses for your custom domain, set the profile-level IP restrictions to allow only the IP addresses for your CDN or reverse-proxy server. For example, allow the addresses that your CDN uses to connect to Salesforce.

Other source IP address request headers, such as the `X-Forwarded-For` (XFF) request header, are ignored. As a result, you can't pass the original source IP address to Salesforce for use with profile IP range restrictions, login history source IP addresses, and event log lines.

Third-Party Changes After Domain Activation

Salesforce validates that your domain points to your org when you add or view your domain in Salesforce. If you update your proxy, web application firewall (WAF), CDN, or the third-party settings after you activate the domain, that change happens outside Salesforce. As a result, Salesforce continues to serve the domain as if it uses HTTPS, even if the third-party configuration doesn't have a valid HTTPS certificate for your custom domain. In this situation, your custom domain can experience disruption.

If you suspect this cause for a disruption, view your custom domain details to validate that your domain points to your org.

DNS Records to Validate Domain Ownership

When you add your domain, Salesforce checks DNS to validate that you own the domain. To pass this verification, you add a canonical name (CNAME) or text (TXT) record for your domain in DNS. After you configure your domain that uses a third-party service or CDN, you can delete the TXT or CNAME record. Deleting unnecessary DNS records can improve performance. Optionally, to make it easier to switch to another domain configuration option in the future, you can keep the CNAME record in DNS. If you choose to keep the record, check with your third-party provider to ensure that their services are supported with that configuration.

Streaming API Features Send CometD HTTP Requests

If your third-party Experience Cloud site includes features that call the Streaming API, the API sends HTTP POST requests to your third-party content delivery network (CDN), not to Salesforce. CDNs have request rate limits, which can be exceeded when your site has many active users at the same time. For more information, see [Streaming API Developer Guide: Using an Experience Cloud Site with Streaming API-Based Features](#).

See Also

[Custom Domains](#)

[Prerequisites for a Custom Domain That Uses a Third-Party Service or CDN](#)

[Set Up a Custom Domain That Uses a Third-Party Service or CDN](#)

[View Your Custom Domain Details](#)

Prerequisites for a Custom Domain That Uses a Third-Party Service or CDN

If a non-Salesforce service or content delivery network (CDN) serves your domain, complete these steps before you add your domain in Salesforce. To confirm your domain ownership, update your domain's Domain Name Service (DNS) configuration. Then work with the third-party provider to ensure access to the required HTTP protocols, configure caching and request handling, and review reverse proxy restrictions. If you're using a third-party CDN, update the header in your CDN to track IP addresses.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

USER PERMISSIONS NEEDED

To view a domain:

[Manage Custom Domains](#)

USER PERMISSIONS NEEDED

OR

[View Setup and Configuration](#)

 **Tip** Unfamiliar with terms like DNS, CNAME, and proxy server? See [Custom Domain Terminology](#).

Update Your Domain's DNS Configuration

When you add a custom domain, Salesforce checks DNS to verify that you own the domain. Before you set up your custom domain that uses a third-party service or CDN, update DNS to include the required canonical name (CNAME) or text (TXT) record for your domain.

- If your domain has an existing A, AAAA, or CNAME record, you can't add a CNAME record. Instead, use a TXT record in DNS to set up a temporary non-HTTPS domain. See [Use a Temporary Non-HTTPS Domain to Serve Your Custom Domain](#).
- Otherwise, add a CNAME record in DNS that points to your org's *.live.siteforce.com CNAME. See [Point Your Custom Domain to Your Org](#).

After you configure your domain that uses a third-party service or CDN, you can delete the TXT or CNAME record. Deleting unnecessary DNS records can improve performance. Optionally, to make it easier to switch to another domain configuration option in the future, you can keep the CNAME record in DNS. If you choose to keep the record, check with your third-party provider to ensure that their services are supported with that configuration.

Verify That Required HTTP Protocols Are Allowed

Some third-party hosts, CDNs, and web application firewalls (WAFs) restrict HTTP protocols. When the third-party provider that serves your custom domain doesn't allow the DELETE and PATCH HTTP protocols, users can't perform actions in Salesforce that rely on those protocols.

For example, assume that a third-party CDN serves a custom domain and that custom domain serves an Experience Cloud site page with an embedded a CRM analytics dashboard. If that third-party CDN doesn't allow the HTTP PATCH and DELETE protocols, users can create a view within that dashboard, but they can't edit or delete that view.

To prevent these issues, verify that your third-party service or CDN allows the POST, PUT, PATCH, and DELETE HTTP protocols. If the provider restricts these protocols by default, work with the provider to allow them for your domain before you activate your custom domain.

Caching

Make sure that when your proxy or CDN service processes an incoming request without a cached

response, the service forwards the request to your custom domain's target host name using HTTPS.

Also, when caching responses, your CDN must honor the Salesforce `Cache-Control` response header. Specifically, ensure that your CDN follows these rules when it operates as a reverse-proxy server.

- Your CDN caches responses only when `public` exists in the Salesforce `Cache-Control` response header.
- If `private`, `no-store`, or `no-cache` exists in the Salesforce `Cache-Control` response header, the CDN doesn't cache that response.
- To determine the cache duration, the CDN uses `s-maxage`, if present in the Salesforce Cache-Control response header. If `s-maxage` isn't present, then the CDN uses `max-age`. The CDN never increases the cache duration, regardless of whether it's derived from `s-maxage` or `max-age`.

Request Configuration: Host HTTP Header

To serve your custom domain with a third-party service or CDN, configure your proxy or CDN service so that the requests sent to Salesforce contain the originally requested `Host` HTTP header. In other words, ensure that your custom domain name—the domain that users see in their original web browser request—is the `Host` HTTP header value in requests sent to Salesforce.

As an example, let's say that a non-Salesforce CDN serves your custom domain, `www.example.com`. When a web browser requests `https://www.example.com/hello/world`, your CDN sends the request to Salesforce at `https://MyDomainName.my.salesforce.com/hello/world` while setting the `Host` header to `www.example.com`. Salesforce then processes the request at `MyDomainName.my.salesforce.com` as a request for `www.example.com` with a path of `/hello/world`. If the `Host` header isn't set to your custom domain, `www.example.com`, then Salesforce can't process the request properly.

URL Paths in Requests

Ensure that your third-party service or CDN processes requests without decoding the path of the requested URLs. For example, if the path includes `%2F`, Salesforce requires that the URL includes `%2F`, not the decoded ASCII value, `/`.

Point Your Custom Domain to Your Org with Your Target Host Name

To point your custom domain that uses a third-party service or CDN to your org, the third-party uses a target host name. A target host name is the host name to which your proxy or CDN forwards requests to your custom domain. In other words, the target host name is how your third-party service or CDN delivers content from your sites in Salesforce.

For custom domains that use a third-party service or CDN, your org's My Domain login URL is the target host name for the domain. Work with your third-party provider to forward your proxy or CDN requests to that host name.

To get the target host name to provide to your third-party service or CDN, in Setup, find and select **Domains**, and then click **Add Domain**.

When you select **Use a third-party service or CDN to serve the domain**, your target host name is included in the guidance for that domain configuration option.

Domain Help for this Page

To serve your Experience Cloud sites or Salesforce Sites on a domain that you own, add and activate the domain. Then add at least one [custom URL](#).

Before you add or rename a domain, point your domain to **[domain].00d0000000000maq.live.siteforce.com** in DNS. For example, if your domain name is www.example.com, add a CNAME record in DNS to www.example.00d0000000000maq.live.sitesforce.com. To learn about how to verify ownership of the domain by pointing your domain to your org, see [Salesforce Help](#).

Note: The unique API identifier for this org is **00d00000000000maq** in lowercase characters.

Domain Edit		Save	Save & New	Cancel
Domain Name	<input type="text"/>			
Domain Configuration Option	<input type="radio"/> Serve the domain with your HTTPS certificate on Salesforce servers <input type="radio"/> Serve the domain with the Salesforce Content Delivery Network (CDN) <input checked="" type="radio"/> Use a third-party service or CDN to serve the domain <input type="text" value="Enter your third-party hostname..."/>			
<p>Use this option if a third-party service or CDN serves your domain. Enter the external hostname that resolves to your proxy, web application firewall (WAF), or CDN service that handles the domain.</p> <p>Before you enable this option, work with your third-party service to use your target hostname, mycompany.my.salesforce.com, and to complete the other prerequisites.</p> <p>To use the recommended CDN for Digital Experiences, select Serve the domain with the Salesforce Content Delivery Network (CDN).</p>				
Allow HSTS preloading registration	<input type="radio"/> Use a temporary non-HTTPS domain <input checked="" type="checkbox"/> Allow HSTS preloading registration			
Associated Org	<input style="width: 150px; height: 20px; border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px;" type="button" value="Production"/>			
<p> After saving, you will need to activate the domain for your changes to take effect.</p>				
Save Save & New Cancel				

Reverse Proxy Restrictions

A reverse proxy is a type of proxy server used to direct client requests to the server that provides the requested resource. Because reverse proxies can increase scalability, performance, resilience, and security, large websites and CDNs often use reverse proxies as part of their load-balancing techniques.

To ensure that traffic is routed to the correct site URL, we recommend that your third-party service or CDN's reverse proxy server forward the full root-relative path of your site. For example, when delivering resources from <https://MyDomainName.my.site.com/store/sales>, the service or CDN's reverse proxy server passes `/store/sales`, not the relative path `/sales`. Otherwise, some pages and features can load resources from paths outside of a site's prefix.

If your third-party service or CDN declines to forward the full root-relative path for all requests, we strongly recommend that you test your custom domain for any resulting issues. During testing, identify the places where resources are incorrectly loaded. Then work with your third-party service or CDN to update their reverse proxy server to correctly handle those requests.

Configure Your CDN to Pass the Origin IP Address

If you use an external CDN and location-based audience targeting in Experience Cloud, set the `True-Client-IP` HTTP header in your external CDN. Without this header, audience targeting can return unexpected results.

If you use an external CDN and you use IP restrictions for location-based audience targeting in Experience Cloud, set the `True-Client-IP` header in your external CDN. This setting helps to pass the IP address of the original client back to Salesforce. Without this header, calls to your site and audience targeting can return unexpected results. For more information about IP address tracking and restrictions with a custom domain, see [Considerations for Custom Domains That Use a Third-Party Service or CDN](#).

For help with setting the `True-Client-IP` header, including any additional recommended settings to protect you against address spoofing, refer to your CDN provider documentation.

See Also

[Salesforce Edge Network](#)

[Custom Domains](#)

[Considerations for Custom Domains That Use a Third-Party Service or CDN](#)

[Set Up a Custom Domain That Uses a Third-Party Service or CDN](#)

Set Up a Custom Domain That Uses a Third-Party Service or CDN

Add a domain in Salesforce that's hosted by a non-Salesforce service or content delivery network (CDN). After you set up your domain in Salesforce, that domain can serve your Experience Cloud sites or Salesforce Sites.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

USER PERMISSIONS NEEDED

To view a domain:

[Manage Custom Domains](#)

OR

USER PERMISSIONS NEEDED

View Setup and Configuration

To add a domain:

Manage Custom Domains

OR

View Setup and Configuration plus either a Site.com Publisher license or Create and Set Up Experiences

To edit or delete a domain:

Manage Custom Domains

To add, edit, and delete custom URLs:

Manage Custom Domains

OR

View Setup and Configuration AND either Create and Set Up Experiences OR a Site.com Publisher license



Tip Unfamiliar with terms like DNS, CNAME, and proxy server? See [Custom Domain Terminology](#).

Before you add a custom domain that's hosted by a third-party service or CDN, review these important considerations.

- With this option, a third-party service or CDN hosts your domain. To use the recommended CDN for Digital Experiences, see [Serve Your Experience Cloud Site with the Salesforce Content Delivery Network \(CDN\)](#). To use an HTTPS certificate that you own to serve your custom domain on Salesforce servers, see [Serve a Custom Domain with Your HTTPS Certificate on Salesforce Servers](#).
- To create or activate a custom domain for testing in a sandbox, log in to the production org that owns the sandbox, and then go to the Domains Setup page.
- To minimize disruption for your users, provision and activate your custom domain when your site traffic is low.

Before you add your domain in Salesforce, review the [considerations](#) for this option, and then complete the [custom domain prerequisites](#).

- Complete the [prerequisites](#) for this option.
- From Setup, in the Quick Find box, enter *Domains*, and then select **Domains**.
- Click **Add a Domain**.
- Enter the domain name.

Salesforce validates ownership based on the fully qualified domain name (FQDN) that you enter when you add a domain to your org. If you get an error message, [point your custom domain to your org](#), and then wait for the changes to propagate. After you update your domain's DNS record, it can take up to

20 hours for that change to take effect worldwide.

- For Domain Configuration Option, select **Use a third-party service or CDN to serve the domain**.

Here's the Domain page when you select the domain configuration option to use a third-party service or CDN to serve the domain.

To serve your Experience Cloud sites or Salesforce Sites on a domain that you own, add and activate the domain. Then add at least one [custom URL](#).

Before you add or rename a domain, point your domain to **[domain].00d0000000000maq.live.siteforce.com** in DNS. For example, if your domain name is [www.example.com](#), add a CNAME record in DNS to [www.example.00d0000000000maq.live.sitesforce.com](#). To learn about how to verify ownership of the domain by pointing your domain to your org, see [Salesforce Help](#).

Note: The unique API identifier for this org is **00d000000000000maq** in lowercase characters.

Domain Edit		Save	Save & New	Cancel
Domain Name	<input type="text" value="www.example.com"/>			
Domain Configuration Option	<input type="radio"/> Serve the domain with your HTTPS certificate on Salesforce servers <input type="radio"/> Serve the domain with the Salesforce Content Delivery Network (CDN) <input checked="" type="radio"/> Use a third-party service or CDN to serve the domain <input type="text" value="Enter your third-party hostname..."/>			
<p>Use this option if a third-party service or CDN serves your domain. Enter the external hostname that v resolves to your proxy, web application firewall (WAF), or CDN service that handles the domain.</p> <p>Before you enable this option, work with your third-party service to use your target hostname, mycompany.my.salesforce.com, and to complete the other prerequisites.</p> <p>To use the recommended CDN for Digital Experiences, select Serve the domain with the Salesforce Content Delivery Network (CDN).</p>				
Allow HSTS preloading registration	<input type="radio"/> <input checked="" type="checkbox"/> Allow HSTS preloading registration			
Associated Org	<input type="text" value="Production"/>			
<p> After saving, you will need to activate the domain for your changes to take effect.</p>				
Save Save & New Cancel				

- Enter your external host name.

If you're hosting the domain yourself, specify the public host name of your host. Typically, the third-party CDN service has a unique canonical name (CNAME). Specify this host name in the external host name field in your domain configuration.

Warning Double check the public host name in the external host name field. If the host name is misspelled or if you don't control the DNS record or the service of the specified host name, an attacker can potentially take over the live service of the custom domain. For example, assume that a third-party CDN serves your custom domain [www.example.com](#) and that the corresponding external host name is [cdn.example.com](#). If you mistakenly enter [cdn.example.com](#) as the external host name, an attacker can register [example.com](#) and use that

incorrect domain to serve content on your custom domain. For steps that you can take to help prevent domain takeovers, see [Maintain Your Custom Domain in Salesforce Help](#).

7. If your domain is a registrable domain such as <https://example.com>, to avoid vulnerabilities during HTTP redirects, select **Allow HSTS preloading registration**.

This setting adds the preload directive to the HSTS header. After you enable this setting, submit your domain at <https://hstspreload.org>. For more information, including how to enable HSTS preloading for a domain with a subdomain, see [Enable HSTS Preloading on a Custom Domain](#).

8. For Associated Org, select the org from which you want this custom domain to serve site content.

For example, to serve the sites in your production org, select **Production**. Or select a sandbox where you want to test this custom domain.

This field only appears in production orgs with associated sandboxes. You can edit this field only from production orgs. For more information, see [Test Your Custom Domains in a Sandbox](#).

9. Save your domain.

After you save your custom domain, Salesforce provisions the domain or gets it ready to be used. The provisioning process can take 4–14 hours. During provisioning, your site can be inaccessible.

When that process is complete, the domain's status on the Domains Setup page changes to Awaiting Activation and you receive an email.

Newly created custom domains use HTTP, not HTTPS, until you activate the domain.

10. Delete the CNAME record that you added in DNS to verify that you own the domain.

Deleting unnecessary DNS records can improve performance. Optionally, to make it easier to switch to another domain configuration option in the future, you can skip this step. Before you opt to keep the CNAME record, check with your third-party provider to make sure that their services are supported with that configuration.

11. To activate your domain, on the Domains Setup page, click **Activate** next to your custom domain name.

 **Note** You edit and activate custom domains for a sandbox in production.

Your site can be unavailable for 5–10 minutes, so activate your custom domain when your site traffic is low. When your site is active, the status changes from Awaiting Activation to Completed and the Current Domain Configuration Option changes to Domain is served by an external host.

12. To serve your sites via your activated domain, [add a custom URL](#).

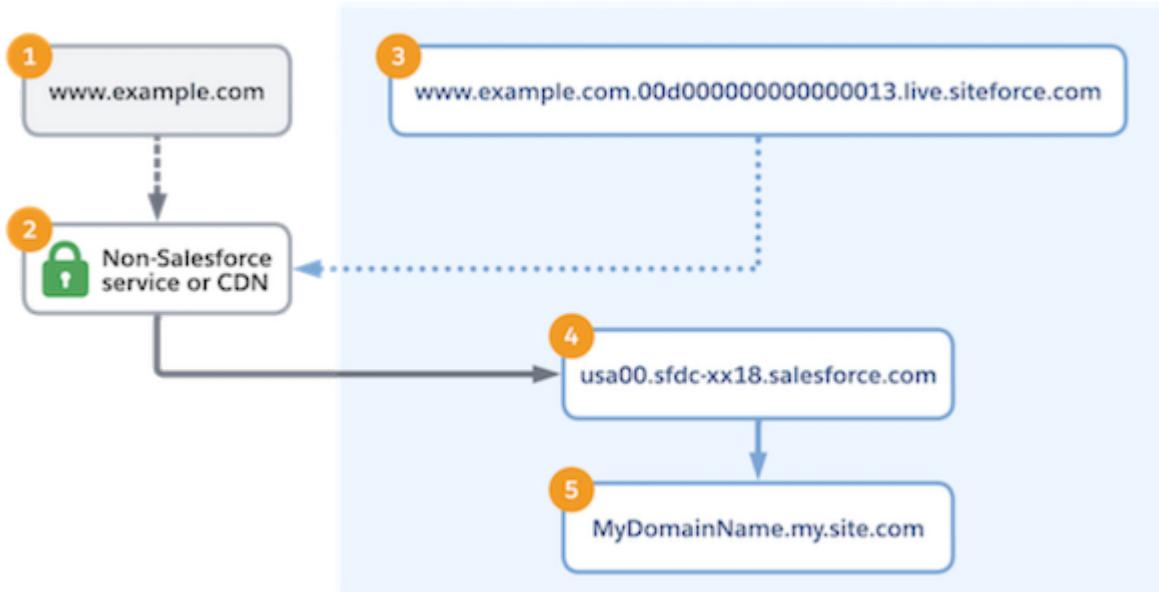
13. If the domain only has legacy Site.com custom URLs unrelated to Experience Cloud sites, manually publish the affected Site.com sites via the Site.com Studio to implement the changes.

If at least one Experience Cloud site or a Salesforce Site exists on the same domain as a legacy Site.com site related to an Experience Cloud site, the changes are automatically published on that domain. For more ways to configure custom domains, see [Options to Serve a Custom Domain](#).

This diagram shows the routing of traffic when a third-party service or CDN serves content from your

Experience Cloud site on your custom domain. In this example, the domain name is www.example.com, the 18-digit org ID is 00d000000000000013, and the org's target host name is usa00.sfcd-xx18.salesforce.com.

The dashed line () represents the configuration that points your domain to your third-party service or CDN. The dotted line () represents routing through DNS, and the solid lines () represent user traffic flow through HTTPS. The gray lines represent traffic that originates outside Salesforce, and the blue lines represent traffic that originates in Salesforce.



Your custom domain (1) points to the third-party service or CDN (2). For example, you point your custom domain to the third party in DNS, or you set up a web application filter (WAF) as a proxy.

The Salesforce CNAME (3) uses the external host name of the domain in Salesforce to point to your non-Salesforce service or CDN (2). In the third-party service or CDN, the configuration points to your org's target host name (4). To identify the domain and serve content from your site (5), Salesforce uses the value passed in the Host HTTP Header of the request from the third-party service or CDN.

See Also

[Custom Domains](#)

[Considerations for Custom Domains That Use a Third-Party Service or CDN](#)

[Prerequisites for a Custom Domain That Uses a Third-Party Service or CDN](#)

[Maintain Your Custom Domain](#)

Use a Temporary Non-HTTPS Domain to Serve Your Custom Domain

Salesforce requires that you serve your custom domain via HTTPS. However, some configuration steps can require a temporary non-HTTPS configuration. If your HTTPS certificate isn't ready to be uploaded to

Salesforce, you can use a temporary domain to start configuring your custom URLs. Also, when your domain serves other content, or when you want to move a custom domain to a new production org, you can use a temporary domain to minimize disruption to your domain.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, and Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

USER PERMISSIONS NEEDED

To view a domain:

Manage Custom Domains

OR

View Setup and Configuration

To add a domain:

Manage Custom Domains

OR

View Setup and Configuration plus either a Site.com Publisher license or Create and Set Up Experiences

To edit or delete a domain:

Manage Custom Domains

To add, edit, and delete custom URLs:

Manage Custom Domains

OR

View Setup and Configuration AND either Create and Set Up Experiences OR a Site.com Publisher license

 **Tip** Unfamiliar with terms like DNS, CNAME, and FQDN? See [Custom Domain Terminology](#).

Before you set up a temporary non-HTTPS domain, complete the custom domain [prerequisites](#). The steps to point your domain to your Salesforce org are included in this topic.

 **Note** This option is a temporary configuration. When your custom domain is served via HTTP, users who attempt to access your custom domain via HTTPS can see a certificate mismatch error and experience a connection timeout.

1. Determine how to serve your domain with HTTPS after you complete the steps that require a temporary non-HTTPS domain. See [Determine How to Serve Your Custom Domain](#).
2. If your fully qualified domain name (FQDN) doesn't have an existing an A, AAAA, or CNAME record in DNS, add a CNAME record that points to your internal Salesforce CNAME. Work with your DNS provider to complete this step. See [Point Your Custom Domain to Your Salesforce Org](#).
3. If an A, AAAA, or CNAME record exists for your FQDN in DNS, add a DNS TXT record to validate your ownership of the domain.

When your domain's DNS record points to a service or server, either through an A or AAAA (address) record that points to an IP address or through an existing CNAME record, removing or updating the existing pointers in DNS can disrupt your website until you set up and activate the custom domain in Salesforce.

To avoid unnecessary downtime, add a DNS TXT record to verify ownership. After you activate your custom domain in Salesforce, you can delete the TXT record.

The domain for which we add the TXT record in DNS varies based on your current configuration.

- a. If an A or AAAA record exists for your FQDN in DNS, add a DNS TXT record for your FQDN that equals your 18-character org ID.

Work with your DNS provider to complete this step.

The TXT record has no effect on the mapping of the existing A or AAAA record.

Here's an example of a DNS TXT record for `www.example.com` that contains an org ID.

Name	TTL	CLASS	TYPE	VALUE
<code>www.example.com.</code>	600	IN	TXT	<code>"00d00000000000map"</code>

- b. If a CNAME record exists for your FQDN in DNS, add a DNS TXT record for the parent domain of your FQDN that equals your 18-character org ID.

Work with your DNS provider to complete this step.

A domain can't have both a CNAME and a TXT record in DNS. So, to avoid downtime for your FQDN, we're going to add a parent domain as a temporary custom domain in Salesforce. Later, the verification process automatically approves your FQDN as a subdomain of an existing custom domain.

To allow Salesforce to verify your ownership of the the parent domain, add a DNS TXT record for the parent domain of your FQDN that points to your 18-digit internal Salesforce CNAME.

For example, if the FQDN that you want to serve your site content is `www.example.com`, add a TXT record for the parent domain `example.com` in DNS.

Here's an example of a DNS TXT record for example.com that contains an org ID.

Name	TTL	CLASS	TYPE	VALUE
example.com.	600	IN	TXT	"00d000000000000map"

4. From Setup, in the Quick Find box, enter *Domains*, and then select **Domains**.
5. Click **Add a Domain**.
6. For Domain Name, enter the FQDN that matches the TXT record that you added in DNS.

Salesforce validates that the domain points to your org via the TXT message.

For example, if you added a TXT record in DNS for example.com a parent domain of www.example.com, enter example.com.

7. For Domain Configuration Option, select **Use a temporary non-HTTPS domain**.

Domain

To serve your Experience Cloud sites or Salesforce Sites on a domain that you own, add and activate the domain. Then add at least one [custom URL](#).

Before you add or rename a domain, point your domain to **[domain].00d000000000000maq.live.siteforce.com** in DNS. For example, if your domain name is www.example.com, add a CNAME record in DNS to www.example.00d000000000000maq.live.sitesforce.com. To learn about how to verify ownership of the domain by pointing your domain to your org, see [Salesforce Help](#).

Note: The unique API identifier for this org is **00d000000000000maq** in lowercase characters.

Domain Edit

Save Save & New Cancel

Domain Name

Domain Configuration Option [i](#)

- Serve the domain with your HTTPS certificate on Salesforce servers
- Serve the domain with the Salesforce Content Delivery Network (CDN) [i](#)
- Use a third-party service or CDN to serve the domain [Enter your third-party hostname](#)
- Use a temporary non-HTTPS domain [i](#)

Allow HSTS preloading registration [i](#) Allow HSTS preloading registration

Associated Org [i](#)

After saving, you will need to activate the domain for your changes to take effect.

Save Save & New Cancel

8. For Associated Org, select the org from which you want this custom domain to serve site content.

For example, to create a temporary custom domain in your production org, select **Production**. Or select a sandbox where you want to test this custom domain.

This field only appears in production orgs with associated sandboxes. You can edit this field only from production orgs. For more information, see [Test Your Custom Domains in a Sandbox](#).

9. Save your domain.

After you save your new domain, Salesforce provisions the domain or gets it ready to be used. Provisioning can take up to 8 hours. During provisioning, your site can be inaccessible and your site visitors can experience errors.

When that process is complete, the domain's status on the Domains Setup page changes to Awaiting Activation and you receive an email.

10. To configure your domain to serve your sites, [add a custom URL](#).

When you add the first custom URL for your domain, Salesforce provisions the domain or gets it ready to be used. The provisioning process can take up to 8 hours. During provisioning, your site can be inaccessible and your site visitors can experience errors.

When that process is complete, the domain's status on the Domains Setup page changes to Awaiting Activation and you receive an email.

11. To activate your domain, on the Domains Setup page, click **Activate** next to your custom domain name.

 **Note** Custom domains for a sandbox are edited and activated in production.

Your site can be unavailable for 5–10 minutes, so activate your custom domain when your site traffic is low. When your site is active, the status changes from Awaiting Activation to Completed and the Current Domain Configuration Option changes to No HTTPS (Temporary).

12. If you added a parent domain, add another temporary non-HTTPS domain for your FQDN that points to another service or server with a CNAME record.

- On the Domains Setup page, click **Add a Domain**.
- For Domain Name, enter the FQDN for your domain.

For example, if you added `example.com` as a temporary non-HTTPS domain so that you can add `www.example.com` as a custom domain, enter `www.example.com`.

This domain name passes the Salesforce verification check because `www.example.com` is a subdomain of `example.com` and `example.com` is an existing domain in Salesforce.

- For Domain Configuration Option, select **Use a temporary non-HTTPS domain**.
- For Associated Org, select the org from which you want this custom domain to serve site content.

For example, to create a temporary custom domain in your production org, select **Production**. Or select a sandbox where you want to test this custom domain.

This field only appears in production orgs with associated sandboxes. You can edit this field only from production orgs. For more information, see [Test Your Custom Domains in a Sandbox](#).

- Save your domain.

After you save your new domain, Salesforce provisions the domain or gets it ready to be used. Provisioning can take up to 8 hours. During provisioning, your site can be inaccessible and your site visitors can experience errors.

When that process is complete, the domain's status on the Domains Setup page changes to Awaiting Activation and you receive an email.

- f. To activate your domain, on the Domains Setup page, click **Activate** next to your custom domain name.



Note Custom domains for a sandbox are edited and activated in production.

Your site can be unavailable for 5–10 minutes, so activate your custom domain when your site traffic is low. When your site is active, the status changes from Awaiting Activation to Completed, and the Current Domain Configuration Option changes to No HTTPS (Temporary).

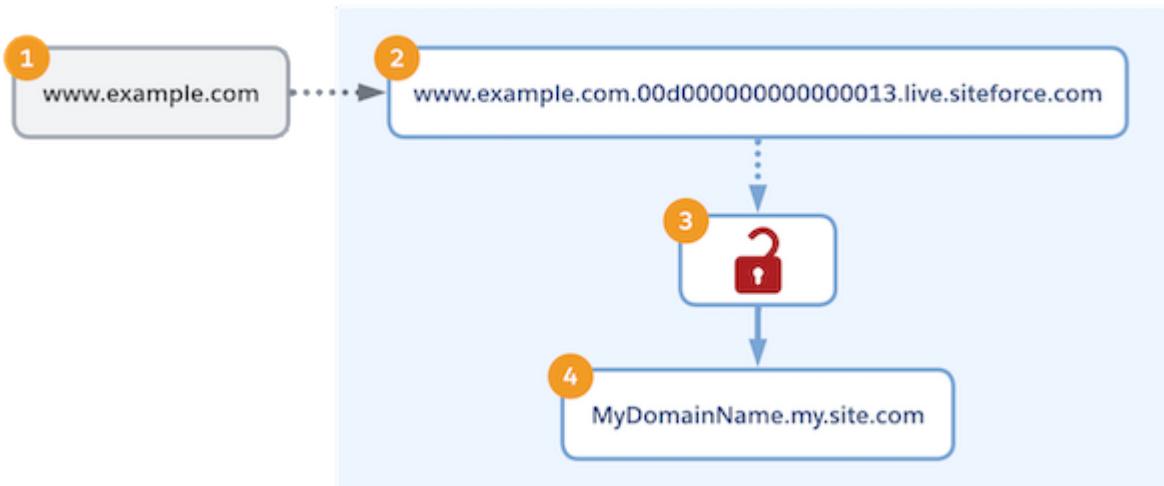
- g. To configure your domain to serve your sites, [add a custom URL](#).

13. Update your domain to use HTTPS.

Until you activate the domain, the configuration uses HTTP.

- a. If your domain serves content from another service, see [Change the Domain Configuration Option for Your Custom Domain](#).
 - b. To move your custom domain to a new production org, see [Move a Domain to Another Production Org](#).
14. When your updated domain is live, remove any temporary configuration in Salesforce and in DNS.
 - a. Delete any temporary non-HTTPS domains that are no longer needed. See [Delete a Domain](#).
 - b. In DNS, delete the TXT records that you added to set up the temporary domain. Work with your DNS provider to complete this step.

This diagram shows the routing of traffic when Salesforce serves your domain as a temporary non-HTTPS domain. Dotted lines (••••→) represent DNS configurations, and the solid line (→) represents user traffic flow through HTTP. The gray line represents traffic that originates outside Salesforce, and the blue lines represent traffic that originates in Salesforce. In this example, the domain name is www.example.com and the 18-digit org ID is 00d000000000000013.



To confirm ownership of your custom domain (1), with your DNS provider, you point the domain to the Salesforce internal CNAME (2), which includes your org ID, via a CNAME or TXT record. If a CNAME record routes traffic to Salesforce, Salesforce uses an HTTP-only endpoint that's served on a secure server (3) to serve the content from your Experience Cloud site (4). However, the hosted certificate (3) supports only HTTP on the custom domain instead of HTTPS. Also, the returned certificate creates a hostname-mismatch error because that certificate doesn't support the custom domain name.

See Also

[Custom Domains](#)

[Custom Domain Management](#)

Add a Custom URL

To specify the site to load when users access your custom domain, add a custom URL. Optionally, you can define additional relationships between your domains and sites by mapping paths for your domain to your sites.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

USER PERMISSIONS NEEDED

To view custom URLs:

Manage Custom Domains

OR

View Setup and Configuration

To add, edit, and delete custom URLs:

Manage Custom Domains

OR

View Setup and Configuration AND either Create and Set Up Experiences OR a Site.com Publisher license

Before you can add a custom URL, set up your custom domain in Salesforce.

A custom URL consists of a domain, a custom path prefix, and a unique site path. For a domain to serve an Experience Cloud site or Salesforce Site, at least one custom URL is required.

In the Custom URLs table, even though the domain name and path are in separate columns, the URLs

consist of the combination of the two.

1. If the custom domain serves a LWR site, associate the domain with a Commerce LWR site or with an LWR site that's hosted on Experience Delivery.
2. From Setup, in the Quick Find box, enter *Custom URLs*, and then select **Custom URLs**.
3. Select **New Custom URL**.
4. Select a domain.
 - a. Enter the name of the domain, or leave the field blank to get a full list of your domains.
 - b. Click the lookup icon () , and then select the name of the domain in the search results.
5. Select a site.
 - a. Enter the name of the site, or leave the field blank to get a full list of your sites.
 - b. Click () , and then click the name of the site in the search results.

Each domain can serve up to 200 sites, and each site can be associated with up to 500 domains. An Experience Cloud site counts as two sites, so if you use only Experience Cloud sites, each domain can serve 100 sites.



Note If you're using a custom domain that has enabled the Salesforce CDN, then the first site you associate it with has to be either a Commerce LWR site or a site hosted on Experience Delivery.

6. Enter a unique path.

To specify the root path, such as `https://www.example.com`, enter a slash (/).

The first character of the path must be a slash (/) to indicate the root. You can extend the path after the slash. For example, if you select the domain, `www.example.com`, and then enter `/products` as the path, the resulting custom URL that serves your selected site is `https://www.example.com/products`.

You can use the same path for more than one domain, but you can use each path only one time within the same domain.

Salesforce reserves some path names, such as `/home` , `/flash` , `/img` , `/servlet`, and `/sfdc`. If you enter a reserved path value, you see an error. In that case, use another relevant prefix to suit your business needs.



Note If you host multiple sites on the same domain, review your site URLs for conflicts because it's possible to configure the same URL for pages on two different sites. Let's say that you host Site A and Site B on the same domain, `www.example.com`. Site A's URL uses the custom URL path prefix `/products`. Site B serves pages from the root path and has a page with the page path `/products`. As a result, both Site A's URL and Site B's page URL are `https://www.example.com/products`. In this scenario, a site visitor can access the Site B page only through a navigation menu on Site B. If a site visitor navigates to `https://example.com/products` any other way, they're directed to Site A. If you identify a potential conflict, either rename the site path or choose a different path for serving the content on your custom domain.

7. To set this custom URL as the preferred URL for authenticated pages and email, select **Site Primary**

Custom URL.

This option is available for the root path for Experience Cloud sites and Salesforce Sites, but not for legacy Site.com sites.

Each site can have only one primary custom URL. If you enable this setting on more than one URL that serves the same site, the last custom URL saved with that setting is used for authenticated pages and email.

For Experience Cloud sites and Salesforce Sites, if you don't select a primary URL, the first HTTPS custom URL in the site, determined alphabetically, is used for authenticated pages and email. If no custom URL uses HTTPS, your system-managed site domain is used. The system-managed domain format for Experience Cloud sites is `MyDomainName.my.site.com` in orgs with enhanced domains and `ExperienceCloudSitesSubdomain.force.com` in orgs without enhanced domains. The system-managed domain format for Salesforce Sites is `MyDomainName.my.salesforce-sites.com` in orgs with enhanced domains and `SalesforceSitesSubdomain.force.com` in orgs without enhanced domains.

8. Save your changes.

If your domain uses your HTTPS certificate on Salesforce servers, saving the first custom URL added to the domain starts the provisioning process for the domain. After the provisioning process is complete, the domain's status on the Domains Setup page changes to Awaiting Activation, and you receive an email. To activate your domain, on the Domains Setup page, click **Activate** next to your custom domain name.

If your custom domain uses another domain configuration option and the domain is active, the path is publicly available.



Example In this example, we want to serve two Experience Cloud sites via our custom domain. The system-managed URL for the site named Storefront is `https://MyDomainName.my.site.com/store`, and the URL for the site named Jobs is `https://MyDomainName.my.site.com/joblisting`. We add the `www.example.com` domain in Salesforce, and then we add two custom URLs. To drive users to the Storefront site, we create a custom URL for the root path of the domain.

- For domain, we select our active domain, `www.example.com`.
- For site, we select **Storefront**.
- For path, we enter `/`.

When we save this custom URL, users who visit `https://www.example.com` see the content from `https://MyDomainName.my.site.com/store`, but the address bar in their browser shows `https://www.example.com`. So users can apply for positions within the company, we create a custom URL for the Jobs site.

- For domain, we select our active domain, `www.example.com`.
- For site, we select **Jobs**.

- For path, we enter `/apply`.

When we save this custom URL, users who visit `https://www.example.com/apply` see the content from `https://MyDomainName.my.site.com/joblistings`, but the address bar in their browser shows `https://www.example.com/apply`.

See Also

[Custom Domains](#)

[Custom Domain Build Example](#)

Custom Domain Build Example

Domains and sites can have a many-to-many relationship through custom URLs. See this relationship in action with an example where three custom domains serve one parent site with two different brands.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

In this example, Example Company owns two clothing brands: Red Example and Blue Example. Their Experience Cloud sites include two sites for each brand. One is a storefront, and the other provides job listings for the brand.

The parent company owns three domains: `https://www.example.com`, `https://www.redexample.com`, and `https://blueexample.com`. They want visitors to access the sites for both brands from each of those domains.

Build the Site

First, the admin builds an Experience Cloud site to serve as a landing page. Because they want to drive traffic to their storefronts, this site highlights the latest trends for both brands and includes navigation links to the Red Example storefront site and the Blue Example storefront sites. The sites for Red Example and Blue Example include a header that links to the other brand's storefront.

In Salesforce, there are five Experience Cloud sites.

- Home—The landing page. Site path: `/`.
Resulting URL: `https://www.example.com/`
- Red Storefront—Storefront for Red Example brand. Site path: `/shop`.
Resulting URL: `https://www.redexample.com/shop`.
- Red Jobs—Job listings for Red Example brand. Site path: `/apply`.

- Resulting URL: <https://www.redexample.com/apply>.
- Blue Storefront—Storefront for Blue Example brand. Site path: /shop.
Resulting URL: <https://www.blueexample.com/shop>.
 - Blue Jobs—Job listings for Blue Example brand. Site path: /apply.
Resulting URL: <https://www.blueexample.com/apply>.

Update DNS

To add the three custom domains as records on the Domain Setup page, Salesforce requires that the domains point to their org. The admin visits the page where they can add a domain to verify the canonical name (CNAME) records values to use when they configure DNS.

The admin works with their DNS provider to add a CNAME record for each domain that points to the corresponding *.live.siteforce.com value.

To serve the three custom domains over the Salesforce Content Delivery Network (CDN), the admin also adds a second _acme-challenge CNAME record for each domain as required by our Salesforce CDN partner.

Add the Domains

After each CNAME record in DNS for each domain target points to their Salesforce org, the admin can add the domains in Salesforce.

From the Domains Setup page, the admin adds three domains: one with the domain name www.example.com, one with the domain name www.redexample.com, and one with the domain name www.blueexample.com. For each domain, the admin selects the domain configuration option, Serve the domain with the Salesforce Content Delivery Network (CDN).

After the admin saves each domain record, Salesforce provisions the domains. In other words, Salesforce gets the new domains ready for activation. When the admin receives an email that each domain is ready for activation, they activate each domain.

Add Custom URLs for the Root Domains

To specify the site to load when users access their root domains, the admin sets up Custom URLs. For each Custom URL, the admin specifies the domain record, the site, and the path. The admin adds Custom URLs with these values.

Domain	Site	Path
www.example.com	Home	/
www.redexample.com	Home	/
www.blueexample.com	Home	/

When the admin adds the custom URL for www.example.com, they also select **Site Primary Custom URL**.

When the admin saves these changes, users who visit `https://www.example.com`, `https://www.redexample.com`, or `https://blueexample.com` see the content from the landing page. However, the address bar in the user's browser shows the corresponding branded URL. Also, because the admin selected Site Primary Custom URL for www.example.com, `https://www.example.com` is the preferred URL for the site's authenticated pages and email.

Add Custom URLs for the Brand-Specific Site Pages

To load the brand-specific sites for Red Example and Blue Example, the admin adds Custom URLs for the storefront and jobs sites.

Domain	Site	Path
www.redexample.com	Red Storefront	/shop
www.redexample.com	Red Jobs	/apply
www.blueexample.com	Blue Storefront	/shop
www.blueexample.com	Blue Jobs	/apply

When the admin saves these custom URLs, users who visit `https://www.redexample.com/shop` or `https://www.blueexample.com/shop` see the corresponding brand's storefront. And users who visit `https://www.redexample.com/apply` or `https://www.blueexample.com/apply` see the corresponding brand's job listings.

See Also

[Custom Domains](#)

[Options to Serve a Custom Domain](#)

[Add a Custom URL](#)

Custom Domain Management

Keep your custom domain up to date with your business needs. Learn how to manage your domain during org migrations and certificate changes. Edit your domain's configuration option, move to a different domain name, or enable HSTS preloading. And when you no longer need a custom domain, delete the domain and its custom URLs in Salesforce.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: Salesforce Sites and LWR, Aura, and Visualforce sites

Maintain Your Custom Domain

To keep your custom domain running smoothly, follow these guidelines. If you serve your domain with your HTTPS certificate, update the certificate before it expires. Optimize your domain that uses the Salesforce content delivery network (CDN). Minimize downtime when you update your domain, and take steps to help prevent domain takeover attacks. And review recommendations for communicating to your users when you change your domain.

View Your Custom Domain Details

Get information about your custom domain. For example, scan the expiration dates for your HTTPS certificates that serve your domains. Or get the target host name for your domain that's served by a third-party service or content delivery network (CDN). Also, you can view and manage the custom URLs associated with the domain, and view alerts about the domain's DNS configuration.

Redirect System-Managed Site URLs to Your Custom Domain

To reinforce your brand, redirect users that visit a system-managed `.my.site.com` or `.my.salesforce-sites.com` URL to the custom domain that serves that site's content.

Enable HSTS Preloading on a Custom Domain

As a security best practice, enable and submit your custom domain for HTTP Strict Transport Security (HSTS) preloading. Connections can be vulnerable when HTTP requests are redirected to HTTPS. An example is a user attempting to access your custom domain using the HTTP protocol. By adding your registrable domain to the third-party HSTS preload list, supported browsers always use HTTPS, protecting your users from attacks during those HTTP redirections.

Update an Expiring Certificate for Your Custom Domain

If you serve your domain with your HTTPS certificate on Salesforce servers, avoid disruption to your domain by renewing or replacing your certificate before it expires. You can find the expiration date for your HTTPS certificates on the Certificate and Key Management Setup page. Also, admins receive an expiring certificate notification email before the certificate expires.

Change the Domain Configuration Option for Your Custom Domain

Change the method for serving your domain. For example, switch from a third-party service or content delivery network (CDN) to the Salesforce CDN. Update your domain in Salesforce to use a different third-party service or CDN to serve your site. Or update a temporary non-HTTPS domain to use HTTPS as required by Salesforce.

Change the Domain Name for a Custom Domain

Whether your brand changed or you want to use a different domain to serve your site content, you can change the domain name of an existing custom domain in Salesforce. Review these important steps to ensure a smooth transition and reduce the risk of takeover attacks.

Move a Domain to Another Production Org

If you purchase a new production org, your old org continues to serve your sites with your custom domain. To move the domain to your new org, set up the domain, and then delete the domain in your old org. Or if you have multiple production orgs, you can move an existing domain to an org of your choice.

Delete a Domain

To stop serving a site via your custom domain, such as `https://www.example.com`, first delete the

custom URLs for that domain. Then delete the domain in Salesforce. Or to point the domain to a different site, set up new custom URLs.

See Also

- [Use a Temporary Non-HTTPS Domain to Serve Your Custom Domain](#)
- [Custom Domains](#)
- [Troubleshoot Common Custom Domain Issues](#)

Maintain Your Custom Domain

To keep your custom domain running smoothly, follow these guidelines. If you serve your domain with your HTTPS certificate, update the certificate before it expires. Optimize your domain that uses the Salesforce content delivery network (CDN). Minimize downtime when you update your domain, and take steps to help prevent domain takeover attacks. And review recommendations for communicating to your users when you change your domain.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

Update an Expiring Certificate

If Salesforce serves your custom domain with your HTTPS certificate, avoid disruption by renewing or replacing your certificate before it expires. Admins receive an expiring certificate notification email before the certificate expires. See [Update an Expiring Certificate for Your Custom Domain](#).

Optimize the Salesforce CDN

When your custom domain serves your Experience Cloud site with the Salesforce CDN, each byte of traffic that's requested on your custom domain counts toward your CDN usage amount. To learn how to monitor your Salesforce CDN usage and what happens if you exceed the terabyte limit, see [Traffic Allowances for the Salesforce CDN](#).

To improve the performance of your LWR Experience Cloud site that's served on the Salesforce CDN, cache Apex methods on the Salesforce CDN.

Change the Domain Configuration Option for Your Domain

Change the method for serving your domain. Switch from using a third-party service to serving your domain with the Salesforce CDN, or change the third party that serves your domain. See [Change the Domain Configuration Option for Your Custom Domain](#).

Help Prevent Domain Takeovers

A domain takeover occurs when a malicious actor controls someone else's domain. They then point the domain to a site that performs malicious activity. When a malicious attacker takes over your domain, your users' trust in your company makes them more susceptible to the attack, and the resulting impact on your users can damage your brand.

A common reason for a domain takeover is a DNS record that points to a resource that's no longer available. Such DNS records are also known as dangling DNS entries. To help prevent these attacks, if you change your domain name, review and update the DNS record for the old domain.

In Salesforce, a domain takeover can also occur when you use a third-party service or CDN to serve your domain, and then you change to another service or domain name. If you no longer have control over the DNS or service for the previous external host name, update your domain in Salesforce.

- If you removed your custom domain from the third-party service or CDN, update the external hostname field for your domain to the new service or [change the domain configuration option](#) for your custom domain.
- If you no longer use the custom domain, [delete it](#).

For example, a third-party CDN serves your custom domain and the corresponding external host name for the domain is `cdn.example.com`. You remove your custom domain from that CDN, but the domain record in Salesforce isn't updated to remove the pointer from `www.example.com` to `cdn.example.com`. In this situation, an attacker can potentially create an account with that CDN and then set up your custom domain in that CDN to serve content that's under their control.

Move a Custom Domain to a New Org

If you purchase a new production org, your old org continues to serve your sites with your custom domain. To move the domain to your new org, set up the domain, and then delete the domain in your old org. Or if you have multiple production orgs, you can move an existing domain to an org of your choice. See [Move a Domain to Another Production Org](#).

Use a Different Domain Name to Serve Your Site Content

Whether your brand changed or you simply want to use a different URL to serve your sites, you have two options.

- To serve your site content on an additional domain, add the domain in Salesforce. More than one custom domain can serve the same site content.
- To switch an existing custom domain to serve the site content on a different domain name and stop using the current domain, see [Change the Domain Name for a Custom Domain](#).

Communicate Site Domain Changes

A change to your domain can impact external users, such as visitors to your Experience Cloud sites. To review recommendations about communicating to these groups before and after you activate an updated domain, see [Notify Users and Customers About a My Domain Change](#).

See Also

[Custom Domains](#)

[Custom Domain Management](#)

View Your Custom Domain Details

Get information about your custom domain. For example, scan the expiration dates for your HTTPS certificates that serve your domains. Or get the target host name for your domain that's served by a third-party service or content delivery network (CDN). Also, you can view and manage the custom URLs associated with the domain, and view alerts about the domain's DNS configuration.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

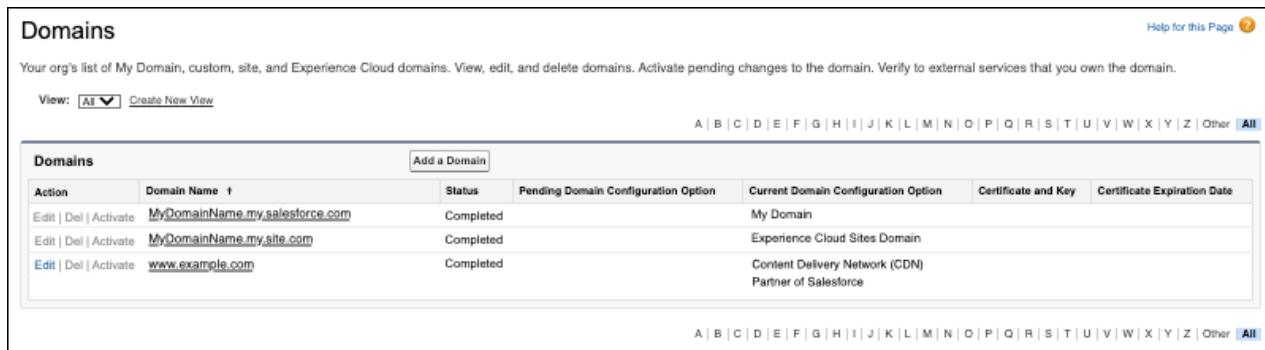
USER PERMISSIONS NEEDED

To view a domain:	View Setup and Configuration
-------------------	------------------------------

To edit or delete a domain:	Manage Custom Domains
-----------------------------	-----------------------

- From Setup, in the Quick Find box, enter *Domains*, and then select **Domains**.

The domains list includes your custom domains and the system-managed domains. System-managed domains are the domains that Salesforce serves for your org, including your Experience Cloud sites, Salesforce Sites, and My Domain.



The screenshot shows the 'Domains' page in Salesforce. At the top, there's a header with 'Domains' and a 'Help for this Page' link. Below the header, a message says 'Your org's list of My Domain, custom, site, and Experience Cloud domains. View, edit, and delete domains. Activate pending changes to the domain. Verify to external services that you own the domain.' There are buttons for 'View' (dropdown), 'Create New View', and 'Add a Domain'. A navigation bar at the bottom includes links for A through Z and 'All'. The main table has columns for Action, Domain Name, Status, Pending Domain Configuration Option, Current Domain Configuration Option, Certificate and Key, and Certificate Expiration Date. Three rows are listed:

Action	Domain Name	Status	Pending Domain Configuration Option	Current Domain Configuration Option	Certificate and Key	Certificate Expiration Date
Edit Del Activate	MyDomainName.my.salesforce.com	Completed		My Domain		
Edit Del Activate	MyDomainName.my.site.com	Completed		Experience Cloud Sites Domain		
Edit Del Activate	www.example.com	Completed		Content Delivery Network (CDN)		

At the bottom of the page, another navigation bar includes links for A through Z and 'All'.

In the domain list, you can see the status of your domain.

Status	What It Means
Awaiting Activation	The provisioning process is complete. To use the updated domain, click Activate .
Awaiting Custom URL	The domain requires a custom URL to start provisioning. Applies to domains that use the Salesforce Cloud domain configuration option only.
Completed	The domain is active and in use.
Provisioning	Salesforce is provisioning the domain. In other words, Salesforce is getting the domain ready for activation.

You can also view the domain's current configuration option. If you saved a change to your domain, the Pending Domain Configuration Option column lists the configuration option that takes effect when you activate the domain.

Domain Configuration Option	Definition
Content Delivery Network (CDN) Partner of Salesforce	A custom domain that uses the domain configuration option, Serve the domain with the Salesforce Content Delivery Network (CDN).
Experience Cloud Sites Domain	If enhanced domains are deployed, the system-managed domain for your Experience Cloud sites that ends in <code>*.my.site.com</code> .
Experience Cloud Sites Force.com Domain	If enhanced domains aren't deployed, the system-managed domain for your Experience Cloud sites that ends in <code>*.force.com</code> .
Domain is served by an external host	A custom domain that uses the domain configuration option, Use a third-party service or CDN to serve the domain.
My Domain	Your org's My Domain login URL.
No HTTPS (Temporary)	A custom domain that uses the domain configuration option, Use a temporary non-HTTPS domain.
Salesforce Cloud	<p>A custom domain that uses the domain configuration option, Serve the domain with your HTTPS certificate on Salesforce servers.</p> <p>For domains with this option in the Current Domain Configuration Option column, the Certificate and Key column includes a hyperlink to the certificate that serves the domain. Also, in the Certificate Expiration Date column, you can see when that certificate expires.</p>
Salesforce Sites	If enhanced domains are deployed, the system-managed domain for your

Domain Configuration Option	Definition
Domain	Salesforce Sites that ends in <code>*.my.salesforce-sites.com</code> .
Salesforce Sites Force.com Domain	If enhanced domains aren't deployed, the system-managed domain for your Salesforce Sites that ends in <code>*.force.com</code> .

For system-managed domains, you can't edit the domain, such as domains that end in `.my.site.com` or `.my.salesforce.com`. To update the subdomain name of these domains, change your My Domain. For more information, see My Domain and Plan for a My Domain Change in Salesforce Help. If you click the domain name for those domains, you see the corresponding Setup page. For example, if you click your My Domain name, you see the My Domain Setup page.

- To view additional information about your custom domain, in the Domain Name column, click the domain name.

The Domain Detail page shows additional information about your domain. The fields vary based on the configuration option for your domain. Here's an example of the Domain Detail page for a domain that's served by a third-party service or CDN.

Domain
www.example.com

Help for this Page ?

Domain Detail

An issue with the external hostname, cdn.example.com, prevents the hostname from serving this domain, www.example.com. To correct this, click Edit and update the external hostname field.

Domain Name	www.example.com	Current Domain Configuration Option	Domain is served by an external host
Allow HSTS preloading registration	<input checked="" type="checkbox"/>	External Hostname	cdn.example.com
Target Host Name	mycompany.my.salesforce.com	Provisioning Status	Completed
Created By	Admin User, 7/31/2023, 11:07 PM	Associated Org	Production
	<input type="button" value="Edit"/>	Modified By	Admin User, 1/8/2024, 3:02 PM
	<input type="button" value="Activate"/>		

Custom URLs

New Custom URL

Custom URLs Help ?

Action	Path	Status	Site Label	Site Type
Edit Del View	/	Published	Storefront	Community
Edit Del View	/store	Published	Storefront	Community

Get Help with Custom Domain Issues

When you access this page, Salesforce validates that your domain points to your Salesforce org. If that validation fails, a warning message provides next steps (1). If that validation succeeds, a message reminds you to keep the DNS record pointed to the internal CNAME for this org.

If your domain is served by an external host, the target host name field shows the value that your third party provider uses to point your domain to your org (2).

To browse or search for information on custom domain setup, maintenance tasks, and common

configuration issues, click **Get Help with Custom Domain Issues (4)**.

In the Custom URLs list (3), you can manage the custom URLs associated with this domain. To go to the site associated with a custom URL, click the site label.

These are the custom URL statuses.

Status	What It Means
Offline	The custom URL is associated with an inactive site.
Preview	The custom URL is active for an Experience Cloud site with a status of Preview.
Published	The custom URL is active and is associated with a site that is live.
Publish Failed	There was an unexpected issue with the background publishing operation for site associated with the custom URL. This status is assigned to the custom URL after the background publishing operation was attempted every hour for two weeks without success.
Publishing	Salesforce is publishing the custom URL in the background. If this custom URL was previously published, the custom URL remains available throughout the publishing process.
Awaiting Publish	<p>The custom URL is ready to be published from within Site.com Studio.</p> <p>This status applies only to legacy Site.com sites when no sites of other types exist on the same domain.</p>
Awaiting Unpublish	<p>The custom URL was removed but is still published. To finish deleting the custom URL, click Unpublish in Site.com Studio.</p> <p>This status applies to legacy Site.com sites only.</p>
In Development	<p>The custom URL is ready to be published.</p> <p>This status applies to legacy Site.com sites.</p>

See Also

[Custom Domains](#)

[Custom Domain Management](#)

[My Domain](#)

[Plan for a My Domain Change](#)

Redirect System-Managed Site URLs to Your Custom Domain

To reinforce your brand, redirect users that visit a system-managed `.my.site.com` or `.my.salesforce-sites.com` URL to the custom domain that serves that site's content.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

USER PERMISSIONS NEEDED

To edit site settings:

Customize Application

OR

Create and Set Up Experiences



Warning If multiple custom domains serve this site, this feature isn't recommended. The site option assumes that only one custom domain serves the site content. There's no method to specify which custom domain to use for redirections.

1. If Salesforce Sites are enabled, from Setup, in the Quick Find box, enter `Sites`, and then select **Sites**. The Sites list at the bottom of the Salesforce Sites page includes both Salesforce Sites and Experience Cloud sites.
2. If only Experience Cloud sites are enabled, access the Force.com sites list.
 - a. From Setup, in the Quick Find box, enter `All Sites`, and then select **All Sites**.
 - b. Click **Workspaces** next to the site name.
 - c. Click **Administration | Pages | Go to Force.com**.
3. Click the site label.
4. On the Site Details page, click **Edit**.
5. Select **Redirect to custom domain**, and then save your changes.

When users visit a system-managed site URL, they're redirected to the equivalent URL on the custom domain.

If no custom domain serves content from this site, this setting has no effect.



Example The custom domain `www.example.com` serves content for the Experience Cloud site `mycompany.my.site.com`. When this setting is enabled, users that visit `https://mycompany.my.site.com/shop/store.html` are redirected to `https://www.example.com/shop/store.html`.

See Also

Custom Domain Management

Enable HSTS Preloading on a Custom Domain

As a security best practice, enable and submit your custom domain for HTTP Strict Transport Security (HSTS) preloading. Connections can be vulnerable when HTTP requests are redirected to HTTPS. An example is a user attempting to access your custom domain using the HTTP protocol. By adding your registrable domain to the third-party HSTS preload list, supported browsers always use HTTPS, protecting your users from attacks during those HTTP redirections.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

USER PERMISSIONS NEEDED

To view a domain:

Manage Custom Domains

OR

View Setup and Configuration

To edit a domain:

Manage Custom Domains

The `Strict-Transport-Security` HTTP header informs browsers to always use HTTPS, a secure connection, to access the domain. However, the first time a user accesses a domain, an HTTP-only connection can be vulnerable while the browser interprets that instruction.

HSTS preloading helps to mitigate this issue. If your domain is on that list, browsers that use the list always treat that domain as requiring a secure connection. For a list of the browsers that support HSTS preloading, see <https://hstspreload.org>. To add your domain to the list, you enable the `Strict-Transport-Security` HTTP header on the registrable domain. Then you add that domain to the third-party HSTS preload list.

-  **Note** Only registrable domains are eligible for HSTS preloading. A registrable domain, –sometimes called a root domain or naked domain–is the domain's public suffix, such as `.com` or `.org`, plus the label to the left of that suffix. An example is `example.com` without the `www` subdomain. So, `example.com` and `example.co.uk` are eligible for HSTS preloading, but `www.example.com`, `www.example.co.uk`, and `sub.example.com` aren't eligible.

To enable HSTS on a custom domain that serves your site content, complete these steps.

1. Enable HSTS preloading on the `Strict-Transport-Security` HTTP header for your custom domain's registrable domain.
 - a. If your domain in Salesforce is a registrable domain such as `https://example.com`, select **Allow HSTS preloading registration** on the domain. To access this setting, edit your domain from the Domains Setup page.
When that setting is enabled, Salesforce includes the `preload` directive in the HSTS header for your custom domain.
 - b. If your domain in Salesforce includes a subdomain, complete the prerequisites for the related registrable domain.
Examples of domains with a subdomain include `https://www.example.com`, `https://shop.example.com`, and `https://shop.example.co.uk`. In all three examples, the registrable domain is `example.com`.
 - You can find the prerequisites on <https://hstspreload.org>. To determine any steps required to qualify for HSTS preloading, use the form on that website. Salesforce can't complete those prerequisites for you.
- If your custom domain is a registrable domain and the Allow HSTS preloading registration option is enabled on your domain, Salesforce adds the required HTTP header. Otherwise, Salesforce can't complete the prerequisites for your domain.
2. To add your domain to the HSTS preload list, go to <https://hstspreload.org>, verify your domain's eligibility, and then submit your domain.
When your registrable domain is on the HSTS preload list, browsers that check that list always use HTTPS to access your domain and its subdomains.
-  **Note** A third party defines and manages the HSTS preload list and its prerequisites. Salesforce can't add your domain to the list for you.

-  **Note** HSTS preloading is enabled on all Salesforce and Visualforce pages, and for all system-managed domains for Experience Cloud sites and Salesforce Sites. The system-managed domain format for Experience Cloud sites is `MyDomainName.my.site.com` in orgs with enhanced domains and `ExperienceCloudSitesSubdomain.force.com` in orgs without enhanced domains. The system-managed domain format for Salesforce Sites is `MyDomainName.my.salesforce-sites.com` in orgs with enhanced domains and `SalesforceSitesSubdomain.force.com` in orgs without enhanced domains. No action is required to enable HSTS preloading on those domains.

See Also

[Custom Domains](#)

[Custom Domain Management](#)

Update an Expiring Certificate for Your Custom Domain

If you serve your domain with your HTTPS certificate on Salesforce servers, avoid disruption to your domain by renewing or replacing your certificate before it expires. You can find the expiration date for your HTTPS certificates on the Certificate and Key Management Setup page. Also, admins receive an

expiring certificate notification email before the certificate expires.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

USER PERMISSIONS NEEDED

To create, edit, and manage certificates: Customize Application

To edit a domain: Manage Custom Domains

 **Note** These instructions apply only to certificates for custom domains that you serve with your HTTPS certificate on Salesforce servers. If your custom domain uses the Salesforce content delivery network (CDN) to serve your Digital Experiences, our CDN partner automatically renews the certificate. If a third-party service or CDN serves your domain, work with that third party to make sure that your certificates remain valid.

Unfamiliar with terms like DNS and certificate? See [Custom Domain Terminology](#).

By default, when you generate a certificate authority (CA)-signed HTTPS certificate in Salesforce, that certificate expires in a year, after which the certificate isn't trusted. If you serve your site with a certificate that you uploaded to your Experience Cloud site, that certificate also expires periodically. To avoid downtime and allow time for testing, update your certificate at least one week before the certificate expires.

1. Download your existing certificate.

It's a good idea to save a copy of any key before you delete it. If there's an issue after you delete a certificate, you can import the key later.

- a. From Setup, in the Quick Find box, enter *Certificate and Key Management*, and then select **Certificate and Key Management**.
- b. In the Label column, select the certificate to download.
- c. On the Certificate and Key Detail page, select Download Certificate.
- d. Save the *.crt file to a safe location.

2. Create and upload a new certificate to Salesforce.

Although it's technically possible to update your existing certificate, we recommend that you create a certificate in Salesforce when your certificate is about to expire. This approach increases the security of your certificate because adding a certificate generates a new public-private key pair for encryption.

- a. If your CA uses intermediate certificates, see the instructions in the knowledge article, [Merge a complete certificate chain for custom HTTPS domains](#).
- b. To upload your certificate to Salesforce, see [Generate a Certificate Signed by a Certificate Authority](#). You can get the required CA signature as a part of that process.
- c. To import an existing certificate that is already signed, see the knowledge article, [Use HTTPS](#)

[certificate that exists within your Community domain.](#)

3. To use the new certificate, update your custom domain.

- a. From Setup, in the Quick Find box, enter *Domains*, and then select **Domains**.
- b. Next to your domain, select **Edit**.
- c. Under the domain configuration option, serve the domain with your HTTPS certificate on Salesforce servers, and clear the certificate field. Then select the lookup icon ().

Domain

To serve your Experience Cloud sites or Salesforce Sites on a domain that you own, add and activate the domain. Then add at least one [custom URL](#).

Before you add or rename a domain, point your domain to **[domain].00d0000000000maq.live.siteforce.com** in DNS. For example, if your domain name is www.example.com, add a CNAME record in DNS to www.example.00d0000000000maq.live.siteforce.com. To learn about how to verify ownership of the domain by pointing your domain to your org, see [Salesforce Help](#).

Note: The unique API identifier for this org is **00d0000000000maq** in lowercase characters.

Domain Edit

Save | Save & New | Cancel

Domain Name:

Domain Configuration Option: Serve the domain with your HTTPS certificate on Salesforce servers
 Serve the domain with the Salesforce Content Delivery Network (CDN) 
 Use a third-party service or CDN to serve the domain 
 Use a temporary non-HTTPS domain 

Select a certificate to serve this domain.

Use this HTTPS option if

- Salesforce hosts an HTTPS certificate that you own, as outlined in [Salesforce Help](#).
- The public DNS record of the domain name points directly to [domain].00d0000000000maq.live.siteforce.com.

Before you enable this option, complete the [prerequisites](#).

To use the recommended CDN for Digital Experiences, select **Serve the domain with the Salesforce Content Delivery Network (CDN)**.

If a third party hosts your domain, if you use a web application firewall (WAF), or if you use a non-Salesforce CDN, select **Use a third-party service or CDN to serve the domain instead**.

Serve the domain with the Salesforce Content Delivery Network (CDN) 
 Use a third-party service or CDN to serve the domain 
 Enter your third-party hostname...

Allow HSTS preloading registration 
 Allow HSTS preloading registration

Associated Org:  Production 

After saving, you will need to activate the domain for your changes to take effect.

Save | Save & New | Cancel

- d. In the lookup window, select the label of the new certificate.
 - e. Save your changes.
- It can take up to 4 hours for the updated certificate to take effect.
4. Validate your domain.
- Because it can take up to 4 hours for the change to take effect, we recommend that you validate your domain both after you save your changes and then again the next day.
5. Optionally, delete your old certificate.
 - a. From Setup, in the Quick Find box, enter *Certificate and Key Management*, and then select

Certificate and Key Management.

- b. For the expired certificate, select **Del**.

This option is available only when no domain, identity provider, single sign-on (SSO) setting, or connected app uses the certificate.

See Also[Custom Domains](#)[Custom Domain Management](#)[Certificates and Keys](#)

Change the Domain Configuration Option for Your Custom Domain

Change the method for serving your domain. For example, switch from a third-party service or content delivery network (CDN) to the Salesforce CDN. Update your domain in Salesforce to use a different third-party service or CDN to serve your site. Or update a temporary non-HTTPS domain to use HTTPS as required by Salesforce.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, and Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

USER PERMISSIONS NEEDED

To view a domain:

[Manage Custom Domains](#)

OR

[View Setup and Configuration](#)

To add a domain:

[Manage Custom Domains](#)

OR

[View Setup and Configuration](#) plus either a Site.com Publisher license or Create and Set Up Experiences

To edit or delete a domain:

[Manage Custom Domains](#)

To add, edit, and delete custom URLs:

[Manage Custom Domains](#)

OR

USER PERMISSIONS NEEDED

View Setup and Configuration AND either Create and Set Up Experiences OR a Site.com Publisher license

 **Tip** Unsure which domain configuration option to use? See [Determine How to Serve Your Custom Domain](#).

1. Complete the prerequisites for your chosen domain configuration option.
 - a. To serve your domain with your HTTPS certificate on Salesforce servers, complete the [prerequisites](#) for that option.
 - b. To serve your Experience Cloud site with the Salesforce CDN, complete the [prerequisites](#) for the Salesforce CDN.
 - c. To use a third-party service or CDN to serve your domain, complete the [prerequisites](#) for that option. These third-party services include third-party hosts, web application firewalls (WAFs), and non-Salesforce CDNs.
2. [Add custom URLs](#) or your custom domain.
At this point, you kept your website up and running while you prepared to use the new domain configuration option. When you're ready to change the domain configuration option for your domain, complete these three steps.
3. If necessary, update the DNS record of your domain. Work with your DNS provider to complete these steps.
 - a. If your new domain configuration option serves your domain with your HTTPS certificate or with the Salesforce CDN, validate that the canonical name (CNAME) record [points to your org](#). If the CNAME record points to your org and you completed the other prerequisites for your new domain configuration option, no further action is required.
 - b. Reduce the Time to Live (TTL) of your domain in DNS. Use a small value, such as 300, for 300 seconds or 5 minutes.

Think of your TTL as a timer for calls to your domain. This value tells servers how long to cache, or keep, a web page before calling DNS again to refresh the page. When you temporarily reduce this value, you minimize the chance of your users calling the previous IP address during the move.

After you make this change, wait for the time duration of the previous TTL before you make other changes to DNS.

- c. To switch to serving your custom domain with your HTTPS certificate on Salesforce servers, update the CNAME record to [point to your org](#).
- d. To switch to serving your custom domain with the Salesforce CDN, update the CNAME record to [point to your org](#).
- e. To switch to using a third-party service or CDN to serve your domain, work with your third-party provider to [use your target host name](#) to forward requests to your domain to your org.

After you update your DNS record, the changes are updated across the internet. That process, called DNS propagation, typically takes a few hours, but it can take up to 72 hours.

4. From Setup in production, in the Quick Find box, enter *Domains*, and then click **Domains**.
5. For your existing custom domain, click **Edit**.
6. Select your new domain configuration option.
 - a. If you have a certificate authority (CA)-signed certificate using Certificate and Key Management for your domain, select **Serve the domain with your HTTPS certificate on Salesforce servers**, and then choose the certificate to serve the domain.
 - b. To use the Salesforce CDN partner to host an Experience Cloud site on your custom domain, select **Serve the domain with the Salesforce Content Delivery Network (CDN)**.

 **Note** This option is unavailable for registrable domains, such as example.com without the www subdomain, and for Salesforce Sites. To serve a registrable domain or a Salesforce Site on a CDN, serve your custom domain with a third-party service or CDN.

For more information and important considerations about the Salesforce CDN, see [Serve Your Experience Cloud Site with the Salesforce Content Delivery Network \(CDN\)](#).

- c. If a third-party service or CDN serves your domain, select **Use a third-party service or CDN to serve this domain**, and then enter the external host name.
7. If your domain is a registrable domain such as https://example.com, to avoid vulnerabilities during HTTP redirects, select **Allow HSTS preloading registration**.
This setting adds the preload directive to the HSTS header. After you enable this setting, submit your domain at <https://hstspreload.org>. For more information, including how to enable HSTS preloading for a domain with a subdomain, see [Enable HSTS Preloading on a Custom Domain](#).
8. For Associated Org, select the org from which you want this custom domain to serve site content.

For example, to serve the sites in your production org, select **Production**. Or select a sandbox where you want to test this custom domain.

This field only appears in production orgs with associated sandboxes. You can edit this field only from production orgs. For more information, see [Test Your Custom Domains in a Sandbox](#).

9. Save your domain.

When you save the domain, Salesforce provisions the domain in your new org. Provisioning can take up to 14 hours for the Salesforce CDN and up to 8 hours for other domain configuration options. During provisioning, your site can be inaccessible and your site visitors can experience errors.

When that process is complete, the status of the domain on the Domains Setup page changes to **Awaiting Activation** and you receive an email.

Until you activate the domain, the configuration uses your previous domain configuration option.

10. To activate your updated domain, on the Domains Setup page, click **Activate** next to your custom domain.
Your site can be unavailable for 5–10 minutes after activation, so activate your custom domain when your site traffic is low. When your site is active, the status changes from Awaiting Activation to **Completed**.
11. When your updated domain is live, reset the TTL of your domain in DNS to your previous setting.

12. If you used a temporary non-HTTPS domain to configure this domain, remove any temporary configuration in Salesforce and DNS.
- [Delete](#) any temporary non-HTTPS domains that are no longer needed.
 - In DNS, delete any TXT records that you added to set up the temporary domain.

See Also

[Custom Domains](#)

[Custom Domain Management](#)

Change the Domain Name for a Custom Domain

Whether your brand changed or you want to use a different domain to serve your site content, you can change the domain name of an existing custom domain in Salesforce. Review these important steps to ensure a smooth transition and reduce the risk of takeover attacks.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

USER PERMISSIONS NEEDED

To view a domain:

[Manage Custom Domains](#)

OR

[View Setup and Configuration](#)

To add a domain:

[Manage Custom Domains](#)

OR

[View Setup and Configuration](#) plus either a Site.com Publisher license or Create and Set Up Experiences

To edit or delete a domain:

[Manage Custom Domains](#)

-  **Note** These instructions apply when an existing custom domain serves your site content and you want to switch to serving the site content on a different domain name. To serve your site content on an additional domain, add the domain in Salesforce. More than one domain can serve the same site content. For an example where three custom domains serve one parent site with two different brands, see [Custom Domain Build Example](#). To minimize the downtime for your site, consider

setting up and activating your new domain in Salesforce, and then [delete](#) the old domain.

To update an existing custom domain to serve your site content on a different domain name and stop using the current domain, complete these steps.

Before you start, determine how to process visits to your old domain after the change. Common approaches include redirecting traffic to your new domain or displaying a landing page that directs visitors to the new page.

 **Tip** To minimize site downtime, update the domain in Salesforce and the DNS records when your org receives minimal traffic, such as during the weekend.

1. Complete the prerequisites for your chosen domain configuration option to serve the domain.
 - a. To serve your domain with your HTTPS certificate on Salesforce servers, complete the [prerequisites](#) for that option
 - b. To serve your Digital Experiences with the Salesforce content delivery network (CDN), complete the [prerequisites](#) for the Salesforce CDN.
 - c. To use a third-party service or CDN to serve your domain, complete the [prerequisites](#) for that option. These third-party services include third-party hosts, web application firewalls (WAFs), and non-Salesforce CDNs.
2. Reduce the Time to Live (TTL) of your domain in DNS. Use a small value, such as 300, for 300 seconds or 5 minutes.

Work with your DNS provider to complete this step.

Think of your TTL as a timer for calls to your domain. This value tells servers how long to cache, or keep, a web page before calling DNS again to refresh the page. When you temporarily reduce this value, you minimize the chance of your users calling the previous IP address during the move.

After you make this change, wait for the time duration of the previous TTL before you make other changes to DNS.

3. If a third-party service or CDN serves your new domain, work with your third-party provider to [use your target host name](#) to forward requests to your domain to your org.
4. If you plan to serve your new domain with your HTTPS certificate on Salesforce servers on with the Salesforce CDN, [point the new domain to your org](#).
5. From Setup in production, in the Quick Find box, enter *Domains*, and then select **Domains**.
6. For the domain with your old domain name, click **Edit**.
7. Clear the Domain Name field, and then enter your new domain name.
Salesforce validates ownership based on the fully qualified domain name (FQDN) that you enter when you add a domain to your org. If you get an error message, [point your custom domain to your org](#), and then wait for the changes to propagate. After you update your domain's DNS record, it can take up to 72 hours for that change to take effect worldwide.
8. If you serve your domain with your HTTPS certificate, update the certificate.
 - a. Under the domain configuration option, Serve the domain with your HTTPS certificate on Salesforce servers, clear the certificate field. Then click the lookup icon ()

Domain

To serve your Experience Cloud sites or Salesforce Sites on a domain that you own, add and activate the domain. Then add at least one [custom URL](#).

Before you add or rename a domain, point your domain to `[domain].00d0000000000maq.live.siteforce.com` in DNS. For example, if your domain name is `www.example.com`, add a CNAME record in DNS to `www.example.00d0000000000maq.live.siteforce.com`. To learn about how to verify ownership of the domain by pointing your domain to your org, see [Salesforce Help](#).

Note: The unique API identifier for this org is `00d000000000000maq` in lowercase characters.

Domain Edit

Save | Save & New | Cancel

Domain Name: www.example.com

Domain Configuration Option: Serve the domain with your HTTPS certificate on Salesforce servers
CA_Signed_Cert_Aug2

Select a certificate to serve this domain.

Use this HTTPS option if:

- Salesforce hosts an HTTPS certificate that you own, as outlined in [Salesforce Help](#).
- The public DNS record of the domain name points directly to `[domain].00d0000000000maq.live.siteforce.com`.

Before you enable this option, complete the [prerequisites](#).

To use the recommended CDN for Digital Experiences, select **Serve the domain with the Salesforce Content Delivery Network (CDN)**.

If a third party hosts your domain, if you use a web application firewall (WAF), or if you use a non-Salesforce CDN, select **Use a third-party service or CDN to serve the domain instead**.

Serve the domain with the Salesforce Content Delivery Network (CDN) [i](#)
 Use a third-party service or CDN to serve the domain [i](#)
 Enter your third-party hostname...
 Use a temporary non-HTTPS domain [i](#)

Allow HSTS preloading registration [i](#) Allow HSTS preloading registration

Associated Org: Production

After saving, you will need to activate the domain for your changes to take effect.

Save | Save & New | Cancel

- b. In the lookup window, select the label of the certificate for your new domain.
9. If you use a third-party service or CDN to serve your domain, update the external host name.

This step is required to serve your new domain. If you no longer have control over the DNS or service of the previous external host name, this action also reduces the risk of a domain takeover attack.

 **Note** Without these steps, an attacker can potentially take over the custom domain's live service. For example, a third-party CDN serves your custom domain and the corresponding external host name for the domain is `cdn.example.com`. You remove your custom domain from that CDN, and the domain record in Salesforce isn't updated to remove the pointer from `www.example.com` to `cdn.example.com`. In this situation, an attacker can potentially create an account with that CDN and then set up your custom domain in that CDN to serve content that's under their control.

10. Save your domain.

After you save your custom domain, Salesforce provisions the domain or gets it ready to be used. The provisioning process can take 4–14 hours. During provisioning, your site can be inaccessible.

When that process is complete, the domain's status on the Domains Setup page changes to Awaiting Activation and you receive an email.

Until you activate the domain, the configuration uses your previous domain name.

11. To activate your domain, on the Domains Setup page, click **Activate** next to your custom domain name.

 **Note** Custom domains for a sandbox are edited and activated in production.

Your site can be unavailable for 5–10 minutes, so activate your custom domain when your site traffic is low. When your site is active, the status changes from Awaiting Activation to Completed.

12. In DNS, remove the records that point to your Salesforce org from your old domain. Work with your DNS provider to complete this step.
 - a. If you use a registrable domain, work with your DNS provider to remove the alias record or canonical name (CNAME) flattening setting that points to your Salesforce org.
 - b. If you serve your domain with your HTTPS certificate, remove the CNAME record for your old domain that points to your internal Salesforce CNAME.
 - c. If you serve the Salesforce CDN, remove the CNAME record for your old domain that points to your internal Salesforce CNAME. Also remove the acme-challenge CNAME for your old domain.
 - d. If a third-party service or CDN serves your domain, work with that third party to remove the target host name that points to your Salesforce org.

For more information on these settings, see [Point Your Custom Domain to Your Salesforce Org](#) and [Prerequisites for a Custom Domain That Uses a Third-Party Service or CDN](#).

13. Update your old domain to redirect or inform users.

Update DNS or work with your third party to establish a redirection to your new domain, or to display a page informing visitors about the new domain.

If you still have control over the old domain, you can reset the TTL in DNS after your users transition to the new site. You can also remove redirections or the informational page about the move when that redirection is no longer required.

See Also

[Custom Domains](#)

[Custom Domain Management](#)

Move a Domain to Another Production Org

If you purchase a new production org, your old org continues to serve your sites with your custom domain. To move the domain to your new org, set up the domain, and then delete the domain in your old org. Or if you have multiple production orgs, you can move an existing domain to an org of your choice.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, and Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

USER PERMISSIONS NEEDED

To view a domain:

Manage Custom Domains

OR

[View Setup and Configuration](#)

To add a domain:

Manage Custom Domains

OR

[View Setup and Configuration plus either a Site.com Publisher license or Create and Set Up Experiences](#)

To edit or delete a domain:

Manage Custom Domains

To add, edit, and delete custom URLs:

Manage Custom Domains

OR

[View Setup and Configuration AND either Create and Set Up Experiences OR a Site.com Publisher license](#)

To move a custom domain from one production org to another production org, complete these steps.



Warning This process requires downtime for your custom domain.

If you set up a custom domain in a sandbox and want to move that domain to production, see [Activate a Sandbox Custom Domain in Production](#).

Prepare the Custom Domain

Keep your website up and running while you prepare to move your domain. Configure the domain in the new production org, but don't activate it yet.

1. Verify that the Experience Cloud sites, Salesforce Sites, and corresponding pages exist in the new org.

If not, move them to the new org.

2. To configure your custom domain in the new org, add a [temporary non-HTTPS domain](#).

With a temporary non-HTTPS domain, your existing domain keeps serving your sites while you configure the domain and custom URLs in the new org.

You can't add the same domain to two orgs on the same Salesforce instance. If your new org is on the same instance as your old org, contact Salesforce Customer Support to move the org to another instance.

3. Complete the prerequisites for your selected domain configuration option to serve the domain.
 - a. To serve your domain with your HTTPS certificate on Salesforce servers, complete the [prerequisites](#) for that option.
 - b. To serve your Digital Experiences with the Salesforce content delivery network (CDN), complete the [prerequisites](#) for the Salesforce CDN.
 - c. To use a third-party service or CDN to serve your domain, complete the [prerequisites](#) for that option. These third-party services include third-party hosts, web application firewalls (WAFs), and non-Salesforce CDNs.
4. [Add custom URLs](#) for your temporary non-HTTPS domain.

Activate the Domain in the New Org

When you're ready, activate it in the new org.

 **Tip** To minimize site downtime, perform these steps when your org receives minimal traffic, such as during the weekend.

1. If necessary, update the DNS record of your domain. To complete these steps, work with your DNS provider or third-party provider.
 - a. Reduce the Time to Live (TTL) of your domain in DNS. Use a small value, such as 300, for 300 seconds or 5 minutes.

Think of your TTL as a timer for calls to your domain. This value tells servers how long to cache or keep a web page before calling DNS again to refresh the page. When you temporarily reduce this value, you minimize the chance of your users calling the previous IP address during the move.

After you make this change, wait for the time duration of the previous TTL before you make other changes to DNS.

- b. If you serve your domain with your HTTPS certificate on Salesforce servers or with the Salesforce CDN, update the canonical name (CNAME) record to [point to your new org](#).
 - c. If a third-party service or CDN serves your domain, work with your third-party provider to [use your target host name](#) to forward requests to your domain to your new org.
 - After you update your DNS record, the changes are updated across the internet. That process, called DNS propagation, typically takes a few hours, but it can take up to 72 hours.
2. From Setup in production, in the Quick Find box, enter *Domains*, and then select **Domains**.

3. For your existing temporary non-HTTPS domain, click **Edit**.
4. Select your domain configuration option.
 - a. If you have a certificate authority (CA)-signed certificate that uses Certificate and Key Management for your domain, select **Serve the domain with your HTTPS certificate on Salesforce servers**. Then select the certificate to serve the domain.
 - b. To use the Salesforce CDN partner to host an Experience Cloud site on your custom domain, select **Serve the domain with the Salesforce Content Delivery Network (CDN)**.

If you use Marketing Cloud Account Engagement (Pardot) in a Professional Edition org, the Salesforce CDN is the only domain configuration option available for your custom domains.

This option is unavailable for registrable domains such as example.com without the www subdomain, and for Salesforce Sites. To serve a registrable domain or a Salesforce Site on a CDN, serve your custom domain with a third-party service or CDN.

For more information and important considerations about the Salesforce CDN, see [Serve Your Experience Cloud Site with the Salesforce Content Delivery Network \(CDN\)](#).

- c. If a third-party service or CDN serves your domain, select **Use a third-party service or CDN to serve this domain**, and then enter the external host name.
5. If your domain is a registrable domain such as https://example.com, to avoid vulnerabilities during HTTP redirects and to have supported web browsers always use secure HTTPS connections for your domain, select **Allow HSTS preloading registration**.
This setting adds the preload directive to the HSTS header. After you enable this setting, submit your domain at <https://hstspreload.org>. For more information, including how to enable HSTS preloading for a domain with a subdomain, see [Enable HSTS Preloading on a Custom Domain](#).
6. For Associated Org, select the org from which you want this custom domain to serve site content.

For example, to serve the sites in your production org, select **Production**. Or select a sandbox where you want to test this custom domain.

This field only appears in production orgs with associated sandboxes. You can edit this field only from production orgs. For more information, see [Test Your Custom Domains in a Sandbox](#).

7. Save your domain.

When you save a domain that uses your HTTPS certificate on Salesforce servers, the status of the domain on the Domain Setup page changes to Awaiting Activation.

When you save a domain that uses another domain configuration option, Salesforce provisions the domain in your new org. Provisioning can take up to 14 hours for the Salesforce CDN and up to 8 hours for other domain configuration options. During provisioning, your site can be inaccessible and your site visitors can experience errors.

When the provisioning process is complete, the status of the domain on the Domains Setup page changes to Awaiting Activation and you receive an email.

8. To serve your sites via your domain, [add a custom URL](#).

When you add the first custom URL for a domain that uses your HTTP certificate on Salesforce servers, Salesforce provisions the domain. In other words, Salesforce gets it ready. The provisioning process can take 4–14 hours. When that process is complete, the domain's status on the Domains Setup page changes from Awaiting Custom URL to Awaiting Activation and you receive an email.

9. To activate your updated domain, on the Domains Setup page, click **Activate** next to your custom domain.

Newly created custom domains use HTTP, not HTTPS, until you activate the domain.

Your site can be unavailable for 5–10 minutes after activation, so activate your custom domain when your site traffic is low. When your site is active, the status changes from Awaiting Activation to Completed.

Clean Up References to the Old Domain

To prevent issues and potential outages, remove references to the old org in Salesforce and DNS.

1. [Delete](#) the domain in your old production org.

! **Important** While the domain exists in two orgs, these issues can occur.
 - Users can be routed to the old IP address and experience errors or view outdated content.
 - When someone deletes the old org, parts of the Salesforce-owned configuration for the custom domain are deleted, too. That deletion causes an outage for the custom domain in the new org.
2. When your updated domain is live, remove any temporary configuration in Salesforce and DNS.
 - a. Work with your DNS provider or third-party service provider to reset the TTL of your domain in DNS to your previous setting.
 - b. [Delete](#) any temporary non-HTTPS domains that are no longer needed.
 - c. In DNS, delete the TXT records that you added to set up the temporary domain.

See Also

[Custom Domains](#)

[Custom Domain Management](#)

Delete a Domain

To stop serving a site via your custom domain, such as <https://www.example.com>, first delete the custom URLs for that domain. Then delete the domain in Salesforce. Or to point the domain to a different site, set up new custom URLs.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: Salesforce Sites and LWR, Aura, and Visualforce sites

USER PERMISSIONS NEEDED

To view custom URLs:

Manage Custom Domains

OR

View Setup and Configuration

To add, edit, and delete custom URLs:

Manage Custom Domains

OR

View Setup and Configuration AND either Create and Set Up Experiences OR a Site.com Publisher license

To view a domain:

Manage Custom Domains

OR

View Setup and Configuration

To edit or delete a domain:

Manage Custom Domains

To delete a custom domain configured with the Salesforce CDN, first ensure the following:

- DNS entries are pointing to the correct organization.
- The domain is using a temporary non-HTTPS domain.
- The domain is activated.

1. If you use a legacy site.com site, deactivate the site in Salesforce.

If you use Experience Cloud sites or Salesforce Sites, skip this step.

2. Delete the custom URLs for your domain.

- a. From Setup, in the Quick Find box, enter *Custom URLs*, and then select **Custom URLs**.
- b. For each custom URL associated with your domain, click **Del** next to the URL name, and then click **OK**.

For Experience Cloud sites and Salesforce Sites, the URL is deleted immediately, and users who visit that URL no longer have access to the site.

With legacy Site.com sites, the URL remains active until you republish or delete the site.

3. To use the Salesforce-hosted domain for your site, or if you set up custom URLs for a different domain to serve your site, activate your site again.

- a. For Experience Cloud sites, see [Activate Your Site](#).
 - b. For Salesforce Sites, from Setup, in the Quick Find box, enter *Sites*, select **Sites**, and then click **Deactivate**.
 - c. For legacy Site.com sites, see [Publish and Manage Live Sites](#).
4. Optionally, delete your domain.
- a. From Setup, in the Quick Find box, enter *Domains*, and then select **Domains**.
 - b. Next to the domain name, click **Del**, and then click **OK**.

For Experience Cloud sites and Salesforce Sites, the domain is deleted immediately, and users who visit that URL no longer have access to any site connected to that domain.

For legacy Site.com sites unrelated to Experience Cloud sites, the domain remains active until you republish or delete the site.



Note When you delete a domain with an unpublished legacy Site.com site attached, it also deletes any custom URLs associated with that site.

5. Work with your DNS provider to update the DNS record for your domain. Remove pointers from the domain to Salesforce.
- For information about the initial configuration, see [Point Your Custom Domain to Your Salesforce Org](#).
6. Optionally, review and revert other configuration related to your domain's configuration option.
- a. If your domain used your HTTPS certificate on Salesforce servers, remove the certificate from Salesforce. On the Certificate and Key Management Setup page, next to your certificate, click **Del**. This option is available only when no domain, identity provider, single sign-on (SSO) setting, or connected app uses the certificate.
 - b. If your domain used the Salesforce CDN, see [Prerequisites for the Salesforce CDN](#).
 - c. If a third-party service or CDN served your domain, see [Prerequisites for a Custom Domain That Uses a Third-Party Service or CDN](#).

See Also

[Custom Domains](#)

[Custom Domain Management](#)

Troubleshoot Common Custom Domain Issues

Review the potential causes and troubleshooting steps for common issues that you can encounter when you set up a custom domain.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions.

Applies to: Salesforce Sites and [LWR, Aura, and Visualforce sites](#)

Unfamiliar with terms like DNS, CDN, certificate, and CNAME? See [Custom Domain Terminology](#)

 **Tip** You can browse or search for information on custom domain set up, maintenance tasks, and common configuration issues without leaving Setup. On the Domain Detail page of a custom domain, click **Get Help with Custom Domain Issues**.

Behavior	Potential Causes and Troubleshooting Steps
When adding a domain and entering the domain name, an admin gets the error, Unable to verify the CNAME target .	<p>The canonical name (CNAME) record is missing or incorrect in the domain's DNS record. Verify that the CNAME record is present and correct for your fully qualified domain name (FQDN). See Point Your Custom Domain to Your Salesforce Org.</p> <p>The change to your domain's DNS record is still being propagated. It can take up to 72 hours for that change to take effect worldwide. Wait for propagation to finish, then try again.</p>
A custom domain is stuck in provisioning.	<p>Provisioning a domain can take 4–14 hours for the Salesforce content delivery network (CDN) and 4–8 hours for other options.</p> <p>If that time has elapsed, check your domain's DNS record for any DNS changes. Then contact Salesforce Customer Support.</p>
A custom domain has a status of Awaiting Custom URL . Provisioning didn't start.	<p>For custom domains that use your HTTPS certification on Salesforce servers, provisioning starts when you add the first custom URL. In this case, the status on the Domains Setup page is Awaiting Custom URL. To start provisioning, add a custom URL for the domain.</p>
When attempting to visit the custom domain, users get the ERR_NAME_NOT_RESOLVED error in Chrome.	<p>The CNAME record is missing or incorrect in the domain's DNS record. Verify that the CNAME record is present and correct for your fully qualified domain name (FQDN). See Point Your Custom Domain to Your Salesforce Org.</p> <p>The domain isn't activated in Salesforce. Check the domain status on the Domains Setup page.</p>
When users attempt to access a custom domain that uses your HTTPS certificate, they see an error that indicates a security issue.	<p>The common name (CN) on the certificate is incorrect. A certificate is valid only if the request host name matches the certificate CN.</p> <p>A mismatch between the certificate and the provisioned domain name. For example, the certificate is for <code>www.example.com</code> and the domain name in Salesforce is <code>example.com</code>.</p> <p>Make sure that your custom domain points to its own <code>*.live.siteforce.com</code> CNAME target. For example, CNAME record for <code>www.example.com</code> points at <code>www.example.com.18CharOrgId.live.siteforce.com</code>, and the CNAME record for <code>products.example.com</code> points at <code>products.example.com.18CharOrgId.live.siteforce.com</code>.</p>

Behavior	Potential Causes and Troubleshooting Steps
When users attempt to access a custom domain that's hosted by a third-party service or CDN, they see an SSL connection error.	<p>Verify that your proxy or CDN forwards requests to your My Domain login URL. For more information, see Prerequisites for a Custom Domain That Uses a Third-Party Service or CDN.</p> <p>The most common cause is the use of a site host name, such as <code>MyDomainName.my.site.com</code>, as a target host name. That configuration fails.</p> <p>Also, before Spring '26, some target host names included your Salesforce instance or Hyperforce cell. Those host names can change during maintenance, such as during an org migration. With those host names, when your instance or cell changes, your custom domain stops working. To fix this issue, work with your third party to update the target host name to your org's My Domain login URL.</p>
	Make sure that the SSL certificate is installed properly on the third party's origin server.
In Setup, the custom domain isn't reflected as the Site URL on the All Sites Setup page.	<p>The custom domain configuration prevents the association. This issue can happen if you use a proxy or point indirectly to the <code>*.live.sitesforce.com</code> CNAME target.</p> <p>Two custom domains serve the same site. Only one is shown on the All Sites Setup page.</p>
Users can't authenticate via the custom domain with methods such as single-sign-on (SSO), connected apps, and auth providers.	<p>If you have two custom domains that serve the same site, verify that the authentication method is configured for both custom domains.</p> <p>Often when the authentication method isn't configured for one domain, authentication methods that use the default site URL ending in <code>*.force.com</code> or <code>*.my.site.com</code> work.</p> <p>If you deployed enhanced domains, see Restore Access to Sites After Enhanced Domains Auto-Deployment.</p>
Some users can't connect to the site via a custom domain that uses the Salesforce CDN.	Check the common issues in the knowledge article Restore Access to Sites After Enhanced Domains Auto-Deployment .
Users see unexpected content when they visit a custom URL.	If you host multiple sites on the same domain, review your site URLs for conflicts. It's possible to configure the same URL for pages on two different sites.

Behavior	Potential Causes and Troubleshooting Steps
	<p>Let's say that you host Site A and Site B on the same domain, <code>https://www.example.com</code>. Site A's URL uses the custom URL path prefix <code>/products</code>. Site B serves pages from the root path and has a page with the page path <code>/products</code>. As a result, both Site A's URL and Site B's page URL are <code>https://www.example.com/products</code>.</p> <p>In this scenario, a site visitor can access the Site B page only through a navigation menu on Site B. If a site visitor navigates to <code>https://example.com/products</code> any other way, they're directed to Site A.</p> <p>If you identify a potential conflict, either rename the site path or select a different path for serving the content on your custom domain.</p>
<p>When users visit my temporary non-HTTPS domain, the browser address bar shows a version of my internal site URL.</p>	<p>This behavior is expected. Because Salesforce requires HTTPS, HTTP-only requests to your site are redirected to a working HTTPS URL.</p> <p>For example, an admin adds <code>www.example.com</code> in Salesforce as a Domain with the “Use a temporary non-HTTPS domain” domain configuration option. The admin activates the domain and adds a Custom URL to load their Experience Cloud site with the CNAME <code>00d000000000000013.live.siteforce.com</code>. When users visit <code>www.example.com</code>, the browser's address bar shows <code>https://00d000000000000013.my.site.com</code> instead of <code>http://www.example.com</code>.</p> <p>To get your custom domain name to persist in the browser's address bar, update the temporary non-HTTPS domain to use another domain configuration option.</p>
<p>Users can't access my custom domain via a link in a different Salesforce org.</p>	<p>You restrict external redirections and the custom domain isn't on the allowlist. See Manage Redirections to External URLs.</p>

See Also

[Custom Domains](#)

[Custom Domain Management](#)

Custom Domain Terminology

Review the terms related to setting up a custom domain in Salesforce.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Professional, Enterprise, Performance, and Unlimited** Editions. Domains created in **Professional** Edition must use the Salesforce CDN.

- A Record—A record added to a domain’s DNS record that maps the domain to an IPv4 address. A DNS record can contain multiple A records.
- AAAA Record (quad A)—A record added to a domain’s DNS record that maps the domain to an IPv6 address. A domain’s DNS record can contain multiple AAAA records.
- Canonical Name (CNAME)—A record in DNS that points to a domain name, rather than an IP address. Often, when sites have subdomains such as `blog.example.com` or `shop.example.com`, those subdomains have CNAME records that point to a root domain (`example.com`). CNAME records follow changes made to the root domain. Use CNAME when you want to make one domain name the alias of another name.
- Certificate Authority (CA)—A trusted entity that stores, signs, and issues digital certificates to authenticate content sent from web servers. Sometimes referred to as a certification authority.
- Content Delivery Network (CDN)—A geographically distributed network of servers that stores cached versions of web assets to optimize page load times and site performance. For more information, see [Content Delivery Networks \(CDNs\) and Salesforce](#).
- Certificate—A file provided by a certificate authority (CA) that authenticates a website’s identity and enables an encrypted connection. The file contains a public key and other data that identifies the private key owner.
For custom domains, CDNs use single and shared certificates. Single certificates display only one brand, while shared certificates display multiple brands. The Salesforce CDN encourages single certificates because they’re more secure and branded. As of Winter ’21, all new sites using the Salesforce CDN use single certificates by default. To switch from a shared to a single certificate, edit your domain, and then select **Single certificate for content delivery network (CDN)**.
- Common Name (CN)—Represents the host name protected by the SSL certificate. The CN serves as a host name identifier of a certificate.
- Custom URL—in Salesforce, a custom URL maps a full path URL to an Experience Cloud site or Salesforce Site in your org. At least one custom URL is required per domain. For more information, see [Add a Custom URL](#).
- DNS Resolver—A server designed to receive DNS queries from web browsers and other applications. The resolver receives a host name and uses the DNS record for the host name to return an IP address.
- Domain Name—A company’s identity on the web. Domain names are human-friendly alphabetic versions of numeric IP addresses. In `https://www.example.com`, the domain name is `example.com`. When you set up a custom domain in Salesforce, unless you’re setting up a registrable domain, include the `www.` prefix in your domain name. See [Point Your Custom Domain to Your Salesforce Org](#).
- Domain Name System (DNS)—A hierarchical and decentralized naming system for computers, services, or other resources connected to the internet or a private network.
- Fully Qualified Domain Name (FQDN)—All the parts of a domain required to look up this authority by name unambiguously using the internet’s DNS system. For example, `www.example.com`.

- HTTP, HTTPS—Hypertext Transfer Protocol, a communications protocol used to connect to web servers on the internet or on a local network (intranet). Hypertext Transfer Protocol Secure (HTTPS) is a protocol that secures communication and data transfer between a user's web browser and a website.
- HTTP Strict Transport Security (HSTS)—A response header that declares that others access the website using HTTPS only, and instructs requesters to convert any future attempts to access the website using HTTP automatically to HTTPS. This highly recommended option helps to protect against man-in-the-middle attacks. For more information, see [Enable HSTS Preloading on a Custom Domain](#).
- IP Address—A unique string of characters that identifies a device using the Internet Protocol to communicate over the internet or a local network. IP addresses come in two formats: IPv4 addresses such as 255.255.255.0, which contain only numbers, or IPv6 addresses such as 2001:0db8:95a3:0000:0000:8c2f:0730:9155, which can contain numbers and letters.
- Naked Domain—See registrable domain.
- Path—In a URL, the path refers to the exact location of a page, post, file, or other asset. It can be analogous to the underlying file structure of the website. The path resides after the host name and is separated by “/”.
- Proxy Server—A proxy server is a dedicated computer or a software system that acts as an intermediary between an endpoint device, such as a computer, and another server. When a user or client makes a request—for example, by visiting a URL or loading an image—the proxy routes the request. The proxy server can exist in the same machine as a firewall server or it can exist on a separate server, which forwards requests.
- Public-Private Key Pair—Two related encryption keys used in public-key cryptography. When data is encrypted with the public key, that data can be decrypted only with the matching private key. The owner of the key pair makes the public key available to anyone but keeps the private key secret. CA-signed HTTPS certificates use a public-private key pair to establish the security of data transmission.
- Registrable Domain—Your unique domain name that you obtained from a domain registrar. The registrable domain of your site is likely your homepage and the highest page in your site hierarchy. Subdomains or pages can be built off the registrable domain, but each page's URL must include the registrable domain to be a part of your site. An example of a registrable domain name is example.com.
- Root Domain—See registrable domain.
- Subdomain—A division of your main domain, often used for a different content type or for a separate business unit. For example, `https://shop.example.com` points to your site's store, while `blog.example.com` points to the site's blog. A subdomain is considered a standalone site. It's related to, but distinct from, the main domain.



Note www is a subdomain that identifies your site as part of the World Wide Web.

- Subdirectory or Subfolder—Subfolders are a further division of your domain to organize and navigate to different sections of your website. For instance: `shop.example.com/newproducts` or `blog.example.com/faq`.
- Top-Level Domain (TLD)—The level of the domain hierarchy that identifies sites based on their commonality of geography (.ca, .au) or purpose (.com, .org).
- Uniform Resource Identifier (URI)—A string that identifies a resource on the internet or an intranet. There are two types of URIs: URLs and URNs.
- Uniform Resource Locator (URL)—
A unique identifier that specifies the location of a resource on the internet or an intranet.

- Uniform Resource Name (URN)—A persistent and location-independent identifier that specifies the name of a resource on the internet or an intranet.
- Web Application Firewall (WAF)—A web application firewall protects web applications for potential attacks, such as cross-site scripting (XSS), injection attacks, sensitive data exposure, and security misconfigurations. It analyzes each HTTP or HTTPS request at the application layer.

See Also

[Custom Domains](#)

Extend the Reach of Your Organization

Sometimes your users need to work with data and services that are outside your Salesforce org. There's a variety of ways you can provide seamless access across org boundaries.

[Provide Actions, Buttons, and Links](#)

Buttons and links let users interact with Salesforce data and with external websites and services, such as search engines and online maps. Salesforce includes several standard buttons and links. You can also create custom ones. Actions let users do tasks, such as create records in the Chatter publisher and in the Salesforce app.

[Manage Your MuleSoft Anypoint Platform Connections](#)

Enable cross-platform features between Salesforce and Anypoint Platform by establishing a tenant relationship.

[MuleSoft API Catalog for Salesforce](#)

Bring your APIs into MuleSoft API Catalog for Salesforce, a single place to view your APIs from different sources and make them available to use in Salesforce. With just a few clicks, activate actions to use in your automations, such as flows, Apex code, and topic and agent actions.

[External Services](#)

Connect your Salesforce org to an external API by using zero lines of code. Use declarative tools and OpenAPI specifications to describe the external API functionality, and External Services automatically creates invocable actions within Salesforce. Use External Services for outbound integrations from Salesforce by using low-code, process-based integrations or to turbocharge your Apex integrations. Call the invocable actions natively from Apex, or create a custom agent action, flow, or Einstein bot that interacts with the external API source.

[Access External Data With Salesforce Connect](#)

Salesforce Connect lets your users view, search, and modify data that's stored outside your Salesforce org. Instead of copying the data into standard or custom objects, use external objects to access the data in real time via web service callouts.

[Work with External Data Sources](#)

An external data source specifies how to access an external system. Salesforce Connect uses external data sources to access data that's stored outside your Salesforce organization. Files Connect uses external data sources to access third-party content systems. External data sources have associated external objects, which your users and the Lightning platform use to interact with the external data and content.

Connect Business Processes with Real-Time Events

Publish and subscribe to platform events to connect business processes in Salesforce and external sources through the exchange of real-time event data. Also, use event relays to integrate platform events and change data capture events with Amazon EventBridge.

Event Studio Overview

Salesforce Eventing encompasses Platform Events and Change Data Capture. Event Studio allows you to track existing event publishing and consumption activity on your Salesforce Event Bus based on event publishers and event subscribers.

Sync Data Between Salesforce and Heroku

Heroku Connect lets you sync data between Salesforce and Heroku Postgres.

Extend Salesforce to External Apps with Lightning Out 2.0

Embed custom Lightning web components (LWC) into external, non-Salesforce apps by using Lightning Out 2.0, a special Salesforce app that you create and configure in Setup. Developers can embed the generated Lightning Out 2.0 app script into a page of your company's external app. Lightning Out 2.0 uses session-based authentication that logs users into Salesforce without leaving the external app, so authenticated users can seamlessly interact with the embedded components as if they were in Salesforce.

Provide Actions, Buttons, and Links

Buttons and links let users interact with Salesforce data and with external websites and services, such as search engines and online maps. Salesforce includes several standard buttons and links. You can also create custom ones. Actions let users do tasks, such as create records in the Chatter publisher and in the Salesforce app.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: All Editions

USER PERMISSIONS NEEDED

To create or edit buttons, links, and actions: Customize Application

Action Types

There are several categories of actions, such as standard Chatter actions, default actions, mobile smart actions, custom actions, and productivity actions. The types of actions that you see depend on the age and configuration of your org.

Custom Buttons and Links

Custom buttons and links help you integrate Salesforce data with external URLs, applications, your company's intranet, or other internal office systems.

Action Types

There are several categories of actions, such as standard Chatter actions, default actions, mobile smart actions, custom actions, and productivity actions. The types of actions that you see depend on the age and configuration of your org.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Quick actions available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

Custom canvas actions available in: **Professional (with Canvas enabled), Enterprise, Performance, Unlimited, and Developer Editions**

Which actions are available in the full Salesforce site depends on whether your org has Chatter, feed tracking, and actions in the publisher enabled. Actions in the Salesforce mobile app don't rely on whether Chatter or actions in the publisher are enabled. For how Chatter enablement affects action visibility, see [Actions With and Without Chatter](#).

Standard Chatter Actions

Standard Chatter actions are available only when Chatter is enabled. The standard Chatter actions are Post, Poll, Question, and Announcements (groups only). Salesforce Classic standard actions also include File, Link, and Thanks (WDC). Standard actions are supported in both the full Salesforce site and in the Salesforce mobile app.

Default Actions

Default actions are sets of predefined actions to get you and your users started using actions in your org. Add default actions to publisher layouts to make them available to your users in the full Salesforce site and the action bar in the Salesforce mobile app.

Mobile Smart Actions

Mobile smart actions are a set of preconfigured quick actions. They're available for account, case, contact, lead, and opportunity pages and on the global publisher layout in the Salesforce mobile app. You can use them to set up quick actions for mobile users with little effort.

Productivity Actions

Productivity actions are predefined and attached to a limited set of objects. Productivity actions include Send Email, Call, Map, View Website, and Read News. Except for the Call action, you can't edit productivity actions.

Quick Actions

Quick actions enable users to do more in Salesforce and in the Salesforce mobile app. With custom quick actions, you can make your users' navigation and workflow as smooth as possible by giving them convenient access to information that's most important. For example, you can let users create or update records and log calls directly in their Chatter feed or from their mobile device.

Mass Quick Actions

With mass quick actions, users can create or update up to 100 records from a list view or related list. Users can select one record in the list to update only that record, or multiple records to perform bulk updates. To help users understand the changes they're making, the action window includes a Fields to update section when users update more than one record.

[Actions in Lightning Experience](#)

In Lightning Experience, actions appear in the Global Actions menu in the header, on related lists, and on list view items. Actions also appear on a record page, in one of several places depending on the action's type.

[Salesforce Mobile App Action Bar](#)

The productivity actions, standard and custom record buttons, and quick actions appear in the action bar and the action menu.

Standard Chatter Actions

Standard Chatter actions are available only when Chatter is enabled. The standard Chatter actions are Post, Poll, Question, and Announcements (groups only). Salesforce Classic standard actions also include File, Link, and Thanks (WDC). Standard actions are supported in both the full Salesforce site and in the Salesforce mobile app.

You can customize the order in which standard Chatter actions appear, but you can't edit their properties. To see standard Chatter actions on an object, the object must have feed tracking enabled.

Only standard Chatter actions appear on the user profile page, regardless of which actions are assigned to the User Page Layout or the global publisher layout. Similarly, only standard Chatter actions appear on reports, regardless of which other actions are assigned to the global publisher layout.

Which actions are available in the full Salesforce site depends on whether your org has Chatter, feed tracking, and actions in the publisher enabled. Actions in the Salesforce mobile app don't rely on whether Chatter or actions in the publisher are enabled. For how Chatter enablement affects action visibility, see [Actions With and Without Chatter](#).

Default Actions

Default actions are sets of predefined actions to get you and your users started using actions in your org. Add default actions to publisher layouts to make them available to your users in the full Salesforce site and the action bar in the Salesforce mobile app.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Essentials, Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, and Developer Editions**

Each default action has a predefined set of fields. Use the page layout editor or global publisher layout to

remove actions or to change the order in which the actions appear. Default actions are supported on the account, case, contact, lead, and opportunity objects.

This table lists the available default actions. Italicized actions are standard Chatter actions.

-  **Note** In orgs created after Winter '14, Salesforce adds default actions to the global publisher layout and to the account, case, contact, lead, and opportunity object page layouts. In orgs created before Winter '14, default actions are available in the palette on the page layout editor, but they're not automatically added to the page layouts.

Layout Page	Default Actions
Global layout (also applies to custom objects)	<ul style="list-style-type: none"> • <i>Post</i> • <i>File</i> • New Event • New Task • New Contact • Log a Call (logged calls are saved as completed tasks) • New Opportunity • New Case • New Lead • <i>Thanks</i> • <i>Link</i> • <i>Poll</i> • <i>Question</i>
Account	<ul style="list-style-type: none"> • <i>Post</i> • <i>File</i> • New Event • New Task • New Contact • New Case • Log a Call (logged calls are saved as completed tasks) • New Note • New Opportunity • <i>Thanks</i> • <i>Link</i> • <i>Poll</i> • <i>Question</i>

Layout Page	Default Actions
Case	<ul style="list-style-type: none"> • <i>Post</i> • <i>File</i> • New Child Case • Log a Call (logged calls are saved as completed tasks) • New Task • New Event • <i>Thanks</i> • <i>Link</i> • <i>Poll</i> • <i>Question</i>
Contact, Lead, and Opportunity	<ul style="list-style-type: none"> • <i>Post</i> • <i>File</i> • New Task • Log a Call (logged calls are saved as completed tasks) • New Case • New Note • New Event • <i>Thanks</i> • <i>Link</i> • <i>Poll</i> • <i>Question</i>

 **Note** Using record types in your org can affect the availability of global default actions for your users. For more information, see [Quick Actions and Record Types](#).

Default Action Fields

Each default action includes a pre-defined set of fields, which lets you make actions available to your users without spending much time on setup.

See Also

[Quick Actions](#)

[Create Dynamic Actions in Lightning App Builder](#)

Default Action Fields

Each default action includes a pre-defined set of fields, which lets you make actions available to your users without spending much time on setup.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Essentials, Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, and Developer Editions**

These fields are included on each default action and appear in the action layout in the order listed.

Action	Fields
Log a Call	<ul style="list-style-type: none"> • Subject (default value: Call) • Comment (description) • Name • Related To
New Case and New Child Case	<ul style="list-style-type: none"> • Contact Name • Status • Subject • Description
New Contact	<ul style="list-style-type: none"> • Name • Email • Phone • Account Name • Title
New Event	<ul style="list-style-type: none"> • Subject • Start • End • All-Day Event • Name • Related To • Assigned To • Location
New Lead	<ul style="list-style-type: none"> • Name • Email • Phone • Company

Action	Fields
	<ul style="list-style-type: none"> • Title
New Note	<ul style="list-style-type: none"> • Title • Public (checkbox) • Body
New Opportunity	<ul style="list-style-type: none"> • Opportunity Name • Account • Close Date (default value: 30 days from today) • Stage • Amount
New Task	<ul style="list-style-type: none"> • Subject • Due Date • Name • Related To • Assigned To • Status

You can change the fields that appear on each action layout using the action layout editor.

See Also

[Default Actions](#)

[Quick Actions](#)

Mobile Smart Actions

Mobile smart actions are a set of preconfigured quick actions. They're available for account, case, contact, lead, and opportunity pages and on the global publisher layout in the Salesforce mobile app. You can use them to set up quick actions for mobile users with little effort.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

 **Note** Mobile smart actions don't appear in the full Salesforce site, regardless of which page layouts

you add them to. They appear only to users in the Salesforce mobile app.

Mobile smart actions are populated with all your org's required fields on the relevant object, regardless of how many fields there are. For example, the New Case action in the mobile smart action bundle includes all required case fields. You can't edit the fields on mobile smart actions. The fields that appear change only if you change which fields on an object are required.

You also can't change which actions are included as part of a mobile smart actions bundle—removing New Event or adding a custom action, for example. To create a more customized set of actions, create the actions you want, add them to the relevant page layouts, and remove the mobile smart actions bundle.

Mobile smart actions appear as a single action element in the page layout editor. In the Salesforce mobile app, the Mobile Smart Actions element expands to distinct create actions that enable users to create records directly from the action bar.

Here's what the mobile smart action element on each supported object expands to include. The actions appear in the action bar and menu in the order shown.

Layout	Actions Included in Mobile Smart Action Bundle
Global layout (also applies to custom objects)	<ul style="list-style-type: none">• New Task• New Contact• Log a Call (logged calls are saved as completed tasks)• New Opportunity• New Case• New Lead
Account	<ul style="list-style-type: none">• New Task• New Contact• New Case• Log a Call (logged calls are saved as completed tasks)• New Note• New Opportunity
Case	<ul style="list-style-type: none">• New Child Case• Log a Call (logged calls are saved as completed tasks)• New Task• New Event

Layout	Actions Included in Mobile Smart Action Bundle
Contact	<ul style="list-style-type: none"> • New Task • Log a Call (logged calls are saved as completed tasks) • New Case • New Note • New Event
Lead	<ul style="list-style-type: none"> • New Task • Log a Call (logged calls are saved as completed tasks) • New Case • New Note • New Event
Opportunity	<ul style="list-style-type: none"> • New Task • Log a Call (logged calls are saved as completed tasks) • New Case • New Note • New Event

See Also

[Quick Actions](#)

[Set Up Actions with Chatter Enabled](#)

Productivity Actions

Productivity actions are predefined and attached to a limited set of objects. Productivity actions include Send Email, Call, Map, View Website, and Read News. Except for the Call action, you can't edit productivity actions.

You can customize the layout of the Call productivity action. The Call action uses the layout of the global Log a Call quick action, which you can edit. You can also customize the Call action for different objects. When you create a Log a Call quick action for an object, you see your custom Log a Call action's layout when you tap **Call** from that object in the Salesforce mobile app.

Productivity actions appear on these objects.

- Account
- Contact

- Event
- Lead
- User
- User Profile

Productivity actions are supported in Lightning Experience and the Salesforce mobile app.

 **Tip** When selecting a record type, don't use Master. Master acts as a placeholder record type that isn't accessible to most profiles. Instead, select a specific record type.

Quick Actions

Quick actions enable users to do more in Salesforce and in the Salesforce mobile app. With custom quick actions, you can make your users' navigation and workflow as smooth as possible by giving them convenient access to information that's most important. For example, you can let users create or update records and log calls directly in their Chatter feed or from their mobile device.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Quick actions available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer** Editions

Custom canvas actions available in: **Professional** (with Canvas enabled), **Enterprise, Performance, Unlimited, and Developer** Editions

Quick actions can also invoke Lightning components, flows, Visualforce pages, or canvas apps with functionality that you define. For example, you can create a custom action so that users can write comments that are longer than 5,000 characters. Or create one that integrates a video-conferencing application so that support agents can communicate visually with customers.

Create quick actions, and add them to your Salesforce Classic home page, to the Chatter tab, to Chatter groups, and to record detail pages. Choose from standard quick actions, such as create and update actions, or create custom actions based on your company's needs.

 **Note** Custom quick actions aren't supported in Chatter groups with customers.

In Salesforce Classic, quick actions appear in the Chatter publisher when Chatter Settings are enabled. In Lightning Experience, they appear in different areas of the user interface, depending on the action's type. In the Salesforce mobile app, actions of all types appear in the action bar, the action bar's action menu, and as list-item actions.

Quick actions come in two flavors.

Object-specific quick actions

Object-specific actions have automatic relationships to other records. Users can quickly create or update records, log calls, send emails, and more in the context of a particular object. For example, you add an object-specific action on the Account object that creates contacts. If a user creates a contact with that action on the detail page for the Acme account, that new contact is associated with Acme.

Global quick actions

You create global quick actions in a different place in Setup than object-specific actions. They're called global actions because they can be put anywhere actions are supported. Users can log call details, create records, or send email, all without leaving the page they're on.

There are several types of object-specific and global quick actions.

- *Create actions* let users create records—like New Contact, New Opportunity, and New Lead. Global create actions enable users to create object records, but the new record has no direct relationship with other records. Object-specific create actions create records that are automatically associated with related records. They're different from the Quick Create and Create New features on the Salesforce Classic home page, because create actions respect validation rules and field requiredness, and you can choose each action's fields.
- *Update actions* let users make changes to a record.
- *Log a Call actions* let users record the details of phone calls or other customer interactions. These call logs are saved as completed tasks. Object-specific Log a Call actions let users enter notes about calls, meetings, or other interactions that are related to a specific record.
- *Custom actions* invoke Lightning components, flows, Visualforce pages, or canvas apps with functionality that you define. Use a Visualforce page, Lightning component, or a canvas app to create global custom actions for tasks that don't require users to use records that have a relationship to a specific object. Object-specific custom actions invoke Lightning components, flows, Visualforce pages, or canvas apps that let users interact with or create records that have a relationship to an object record.
- Object-specific Send Email actions, available only on cases, give users access to a simplified version of the Case Feed Email action in the Salesforce mobile app. You can use the case-specific Send Email action in Salesforce Classic, Lightning Experience, and the Salesforce mobile app.
- Global *Send Email actions* are supported only in Lightning Experience. You can't add them to the Cases layout or use them with cases.
- *Question actions* enable users to ask and search for questions about the records that they're working with.

For create, Log a Call, and custom actions, you can create either [object-specific actions](#) or [global actions](#). Update actions must be object-specific.

Agent Quick Actions

Agent quick actions are AI-powered, and allow your users to take advantage of Agentforce directly from record pages. Clicking an agent quick action opens the Agentforce panel and directs Agentforce

to use the topic underlying the utterance in the agent panel.

Global Quick Actions

You can add global quick actions to almost any page that supports actions. Use global actions to let users log call details, create or update records, or send email, all without leaving the page they're on. Global create actions enable users to create object records, but the new record has no direct relationship with other records.

Object-Specific Actions

Object-specific actions let users quickly create or update records, log calls, send emails, and more, in the context of a particular object.

Lightning Component Actions

Lightning component actions are custom actions that invoke a Lightning component. They support Apex and JavaScript and provide a secure way to build client-side custom functionality. Lightning component actions are supported only in the Salesforce mobile app and Lightning Experience.

Lightning Web Component Actions

Lightning web component (LWC) actions are custom quick actions that invoke a Lightning web component. They support JavaScript and provide a secure way to build client-side functionality. LWC actions are supported only on record pages in Lightning Experience.

Start a Flow from a Quick Action

With this feature, users can run a screen flow with the click of a button on a Lightning record page. Create a quick action on an object and configure it to run an active screen flow. Then, add the quick action button to the Lightning record page layout.

Visualforce Pages as Object-Specific Custom Actions

A Visualforce page added as a custom action on an object is invoked in the context of a record of that object type. The custom action is passed a specific record ID—the record the user was looking at when the user clicked the custom action. Design the page to act on that specific record type.

Enable Actions in the Chatter Publisher

Enabling actions in the publisher lets you add actions that you've created to the Chatter publisher. Actions appear on the Home page, on the Chatter tab, in Chatter groups, and on record detail pages.

Action Layout Editor

Just as object record pages have page layouts that can be customized, actions have action layouts that can be customized. When you create an action, Salesforce populates its layout with a default set of fields. You can add, remove, or reorder fields on the action layout to present only the essential items your users need when they use the action.

Custom Success Messages for Quick Actions

For Create a Record, Update a Record, and Log a Call action types, you can create a custom message that displays when the action executes successfully.

Customize Actions with the Enhanced Page Layout Editor

Use the page layout editor to customize which actions show up in Salesforce and in the Salesforce mobile app.

Set Predefined Field Values for Quick Action Fields

When you create actions, use predefined field values to set a value for a field. Predefined values can help ensure consistency and make it faster and easier for users to create records.

Global Publisher Layouts

Global publisher layouts determine the global actions that appear in the various Salesforce interfaces. In Salesforce Classic and Lightning Experience, these layouts customize the actions on global pages (like the Home page) and on the Chatter page. Lightning Experience also uses these layouts to populate the Global Actions menu. And in the Salesforce mobile app, these layouts drive the actions that appear in the action bar on the Feed and People pages. Global publisher layouts can include global actions only.

Quick Actions and Record Types

Using record types in your organization can affect the availability of quick actions for your users.

Actions Best Practices

Use these tips as you set up actions.

Quick Action Considerations

Keep these considerations in mind when working with quick actions.

Troubleshooting Actions

Use these tips to help you solve problems that arise with actions.

Action Limits and Limitations

Actions can work differently than you expect in certain situations and for different objects. Keep these limits and limitations in mind when working with actions.

See Also

- [Set Up Actions with Chatter Enabled](#)
- [Actions Best Practices](#)
- [Quick Action Considerations](#)
- [Create Dynamic Actions in Lightning App Builder](#)

Agent Quick Actions

Agent quick actions are AI-powered, and allow your users to take advantage of Agentforce directly from record pages. Clicking an agent quick action opens the Agentforce panel and directs Agentforce to use the topic underlying the utterance in the agent panel.

REQUIRED EDITIONS

Available in: Lightning Experience and the Salesforce mobile app

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions with the Einstein for Sales, Einstein for Service, or Einstein Platform add-ons.

You can add agent quick actions to an object's page layout using the page layout editor, or to the Lightning page using Lightning App Builder.

In order to use agent quick actions in your org, first [enable generative AI](#). Your org must also have an active Agentforce (Default) agent or at least one Agentforce Employee agent. If there are multiple agents active in an org, the agent being used must contain at least one action whose description matches that utterance. For example, for a “summarize contact” agent quick action to work, the “Summarize Record”

action must be added to a topic to the active agent currently being used.

Users must have access to at least one agent in your org in order to use the agent quick action functionality on record pages. For information on granting users access, see [Give Users Access to Agentforce \(Default\)](#) or [Manage Employee Agent Access](#) in Salesforce Help.

If they don't have access to an agent, they still see the agent quick action on record pages by default, but can't access the functionality. Admins can use page layouts or visibility rules to limit access to agent quick actions to users who don't have access to agents.

[Create an Agent Quick Action](#)

Create agent quick actions to give your users the ability to streamline their work by using AI-powered actions.

Create an Agent Quick Action

Create agent quick actions to give your users the ability to streamline their work by using AI-powered actions.

REQUIRED EDITIONS

Available in: Lightning Experience and the Salesforce mobile app

Available in: **Enterprise**, **Performance**, and **Unlimited** Editions with the Einstein for Sales, Einstein for Service, or Einstein Platform add-ons.

USER PERMISSIONS NEEDED

To create actions:	Customize Application
--------------------	-----------------------

1. From Setup, click **Object Manager**, click the object that you want to create the action for, and then click **Buttons, Links, and Actions**.
2. Click **New Action**.
3. Select **Agent Quick Action**.
4. In the User Utterance field, add an utterance that tells the agent what to do. For example, to summarize the record in question, enter *Summarize record*. You can also use the record ID merge field with the syntax *Summarize record with ID \${RECORD_ID}*.
5. Enter a label for the action. Users see this label as the name of the action.
6. Save your work.

When you finish creating your action, add it to the desired page layout. In the Page Layout Editor, find the newly created action **Mobile & Lightning Actions**.

Users must have access to at least one active Agentforce (Default) agent or Agentforce Employee agent in your org in order to use the agent quick action functionality on record pages. For information on granting users access, see [Give Users Access to Agentforce \(Default\)](#) or [Manage Employee Agent Access](#) in Salesforce Help.

Global Quick Actions

You can add global quick actions to almost any page that supports actions. Use global actions to let users log call details, create or update records, or send email, all without leaving the page they're on. Global create actions enable users to create object records, but the new record has no direct relationship with other records.

REQUIRED EDITIONS

 **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

 **Note** Global quick actions aren't context-aware, so fields can't be populated with information from the current page. Instead, add the appropriate quick action to a page to specify predefined fields.

Global actions live on a special layout of their own, known as the *global publisher layout*. The layout isn't associated with an object, and it populates the global actions menu in Lightning Experience. Users can access the global actions menu by clicking  in the Salesforce header.

Check out this video to see how global actions work.

Watch the video: <https://salesforce.vidyard.com/watch/1F2Pb54GkLKhgFTqF61hCd>

If an object page layout isn't customized with actions, the actions on those object record pages are inherited from the global publisher layout.

You can do lots of different things with global quick actions.

- To let users send emails from anywhere in Lightning Experience, add a Send Email action to the global layout. Add a Send Email action to any page layout for objects that are enabled for activities. Global Send Email actions are supported only in Lightning Experience. You can't add a global Send Email action to the Cases layout or use the action with cases.

 **Note** To let users send emails from anywhere in Lightning Experience, keep a Send Email action on the global publisher layout, in the Salesforce Mobile and Lightning Experience Actions section.

- To let users record call details, add Log a Call actions to global layouts. For example, users can log calls from global pages in Salesforce Classic, such as the Home page and the Chatter tab, or in the Salesforce mobile app from the Feed or Groups pages. In Lightning Experience, Log a Call actions on global layouts display in the Global Actions menu.
If you have multiple Log a Call actions and use simpler task forms on the Salesforce mobile app, mobile users see the first valid Log a Call action listed for the mobile publisher layout.
- Use a Visualforce page, Lightning component, or a canvas app to create global custom actions for

- tasks that don't require users to use records that have a relationship to a specific object. Canvas apps that you want to use as custom actions require Publisher as a location. Visualforce pages that you want to use as global custom actions can't use standard controllers. For example, you want a custom action that lets users enter a street address and see a map, the local time, and the local weather. For this action, create a Visualforce page that doesn't use any of the standard controllers, and add it as a custom global action.
- You can also use a global Create a Record quick action to enable your Salesforce for Outlook users to create records directly from the Salesforce side panel.

 **Note**

- Chatter groups with customers don't support global create, log a call, or custom actions and display only standard Chatter actions, such as Post, File, Link, and Poll.
- Actions to create records for an object that is the detail object in a master-detail relationship must be object-specific, not global.

Supported Objects for Create Actions

You can create global actions to let users create many kinds of records, including:

- Account
- Asset
- Badge
- Campaign
- Case
- Contact
- Contract
- Custom objects
- Event (without invitees)
- Goal
- Group
- Knowledge object
- Lead
- Note
- Opportunity
- Orders
- Person Account
- Question
- Reward
- Task
- Work Order

Create Global Quick Actions

You can add global actions to any page that supports actions, like the Home page, the Chatter tab,

object pages, and custom Lightning app pages. For example, add a Log a Call action to global layouts so users can record call details right from a Chatter thread.

See Also

- [Object-Specific Actions](#)
- [Actions With and Without Chatter](#)
- [Quick Action Considerations](#)
- [Quick Actions](#)
- [Create Dynamic Actions in Lightning App Builder](#)

Create Global Quick Actions

You can add global actions to any page that supports actions, like the Home page, the Chatter tab, object pages, and custom Lightning app pages. For example, add a Log a Call action to global layouts so users can record call details right from a Chatter thread.

REQUIRED EDITIONS

USER PERMISSIONS NEEDED

To create actions:	Customize Application
--------------------	-----------------------

1. From Setup, in the Quick Find box, enter *Actions*, and then select **Global Actions**.
2. Click **New Action**.
3. Select the type of action to create.
4. Customize the action.

For a Create a Record action, select the type of object to create. If the object has more than one record type, select the one that you want to use for records created through this action.

For a Custom Visualforce or Custom Canvas action, select a Visualforce page or canvas app, and then specify the height of the action window. The width is fixed.

For a Lightning Component action, select the component to be called by the action.

 **Note** While you can create Lightning Web Component global actions and add them to the Global Publisher Layout, these actions are available for use only in the Salesforce Field Service (SFS) mobile app. These actions aren't available in Lightning Experience on mobile or desktop.

5. Enter a label for the action. Users see this label as the name of the action.

 **Tip** You can choose an option from the Standard Label Type list to have Salesforce generate the label. For the labels in this list that include [Record] and [Record Type], Salesforce fills in the type of object or the record type the action creates. For example, if you choose the Create New [Record] standard label on a create contact action, the generated label is Create New Contact.
6. If necessary, change the name of the action. If you selected a standard label type in the previous step, you must enter the name.

This name is used in the API and managed packages. It must begin with a letter and use only alphanumeric characters and underscores, and it can't end with an underscore or have two consecutive underscores. Unless you're familiar with working with the API, we don't recommend editing this field.

7. Type a description for the action.

The description appears on the detail page for the action and in the list on the Buttons, Links, and Actions page. The description isn't visible to your users. If you're creating several actions on the same object, use a detailed description, such as "Create Contact on Account using New Client record type."

8. Select to post a feed item in the record feed when the action's completed.

When enabled, **Create Feed Item** causes the creation of a feed item when the action is performed. The target object's compact layout defines the feed item's fields.

When feed tracking is enabled for the object, and **All Related Objects** is selected, performing the action causes the creation of a feed item. The feed item is created, regardless of whether **Create Feed Item** is selected.

9. For a Create a Record, Update a Record, or Log a Call action, you can add a custom success message.

The success message displays after the action executes successfully.

10. Optionally, to select a different icon for the action, click **Change Icon**.

Custom images used for action icons must be less than 1 MB

11. Save your changes.

After you create a quick action, customize its layout, [add predefined values](#), and then [add the action to page layouts](#).



Tip If you delete an action, the action is removed from all layouts that it's assigned to.

See Also

- [Set Predefined Field Values for Quick Action Fields](#)
- [Customize Actions with the Enhanced Page Layout Editor](#)
- [Custom Success Messages for Quick Actions](#)
- [Visualforce Pages as Object-Specific Custom Actions](#)
- [Create Dynamic Actions in Lightning App Builder](#)

Object-Specific Actions

Object-specific actions let users quickly create or update records, log calls, send emails, and more, in the context of a particular object.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Quick actions available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

Custom canvas actions available in: **Professional** (with Canvas enabled), **Enterprise, Performance, Unlimited, and Developer Editions**

For example, you add an object-specific action on the Account object that creates contacts. If a user creates a contact with that action on the detail page for the Acme account, that new contact is associated with Acme.

Object-specific actions are only available on page layouts for that object. For example, you can add the New Group Member action only to the group publisher layout.

When a user creates a record by using an object-specific create action, a feed item for that record appears:

- In the feed for the record on which the new record was created
- As the first entry in the feed for the new record
- In the Chatter feed of the user who created the record
- In the user profile feed for the user who created the record
- In the Chatter feed of any users who follow the record on which the record was created
- In the Chatter feed of any users who, through custom triggers or auto-follow rules for new records, automatically follow the new record

There are several types of object-specific quick actions.

- Object-specific Create a Record actions create records that are associated with related records.
- Object-specific Log a Call actions let users enter notes about calls, meetings, or other interactions that are related to a specific record.



Note Make sure that you have only one Log A Call action on your page layout. If there are multiple Log a Call actions, Salesforce mobile app users see the Task layout when they click **Call** on their mobile devices. The required and editable fields from the layout can appear in a different order.

- Object-specific Update a Record actions make it easy for users to edit records. You can define the fields that are available for update.
- Object-specific custom actions invoke Lightning components, flows, Visualforce pages, or canvas apps that let users interact with or create records that have a relationship to an object record. The Visualforce page for an object-specific custom action must include the standard controller for the relevant object. For example, use the standard contact controller to create a custom action that lets users import a contact's Twitter profile and add that information to a contact record.
- Object-specific Send Email actions, available only on cases, give users access to a simplified version of the Case Feed Email action in the Salesforce mobile app. You can use the case-specific Send Email action in Salesforce Classic, Lightning Experience, and the Salesforce mobile app.

Supported Objects

When you create an object-specific action, you can choose as a target object an event, a task, or any object that has a parent-child or lookup relationship to the host object. You can't choose Quote as a target object from Opportunity. However, you can still create quotes from an opportunity by going to the opportunity's Quotes related list and clicking **New**.

You can create object-specific actions on many objects, including:

- Account
- Campaign
- Case
- Contact
- ContentNote
- Custom objects
- Group
- Lead
- Opportunity

[Create Object-Specific Quick Actions](#)

Object-specific actions let users create records with automatic relationships to other records, make updates to specific records, and interact with records in ways that you define.

See Also

- [Global Quick Actions](#)
- [Actions With and Without Chatter](#)
- [Quick Action Considerations](#)
- [Quick Actions](#)
- [Send Email Action Considerations for Cases](#)
- [Create Dynamic Actions in Lightning App Builder](#)

Create Object-Specific Quick Actions

Object-specific actions let users create records with automatic relationships to other records, make updates to specific records, and interact with records in ways that you define.

REQUIRED EDITIONS

 **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

USER PERMISSIONS NEEDED

To create actions:

Customize Application

1. From the management settings for the object for which you want to create an action, go to Buttons, Links, and Actions.
2. Click **New Action**.
3. Select the type of action to create.
4. Customize the action.

For a Create a Record action, select the type of object to create.

- If the object has more than one record type, select the one you want to use for records created through this action.
- If the object for which you're creating the action has multiple relationships with the target object, select the field to populate when a record's created. If the two objects have a master-detail relationship, you can't select which field to populate. The master-detail Relationship field is selected by default, and you can't change this setting. Set the Relationship field to Read-Only so that users can view the field on the object.
- You can't choose Quote as a target object from Opportunity. However, you can still create quotes from an opportunity by going to the opportunity's Quotes related list and clicking **New**.

For a Custom Visualforce action, select the Visualforce page, and then specify the height of the action window. The width is fixed.

For a Lightning Component action, select the component called by the action.

For a flow action, select the flow to render.

5. Enter a label for the action. Users see this label as the name of the action.

 **Tip** You can choose an option from the Standard Label Type list to have Salesforce generate the label. For the labels in this list that include "Record" and "Record Type," Salesforce fills in the type of object or the record type the action creates. For example, if you choose the Create New "Record" standard label on a create contact action, the generated label is Create New Contact.

6. If necessary, change the name of the action. If you selected a standard label type in the previous step, you must enter the name.

This name is used in the API and managed packages. It must begin with a letter and use only alphanumeric characters and underscores, and it can't end with an underscore or have two consecutive underscores. Unless you're familiar with working with the API, we suggest not editing this field.

7. Type a description for the action.

The description appears on the detail page for the action and in the list on the Buttons, Links, and Actions page. The description isn't visible to your users. If you're creating several actions on the same object, use a detailed description, such as "Create Contact on Account using New Client record type."

8. Select to post a feed item in the record feed when the action's completed.

When enabled, **Create Feed Item** causes the creation of a feed item when the action is performed. The target object's compact layout defines the feed item's fields.

When feed tracking is enabled for the object, and **All Related Objects** is selected, performing the action causes the creation of a feed item. The feed item is created, regardless of whether **Create Feed Item** is selected.

9. For a Create a Record, Update a Record, or Log a Call action, you can add a custom success message. The success message displays after the action executes successfully.
10. Optionally, click **Change Icon** to select a different icon for the action.

Custom images used for action icons must be less than 1 MB in size.

11. Click **Save**.

After you create a quick action, customize its layout, [add predefined values](#), and then [add the action to page layouts](#).



Note If you delete an action, the action is removed from all layouts that it's assigned to.

See Also

- [Set Predefined Field Values for Quick Action Fields](#)
- [Customize Actions with the Enhanced Page Layout Editor](#)
- [Visualforce Pages as Object-Specific Custom Actions](#)
- [Custom Success Messages for Quick Actions](#)

Lightning Component Actions

Lightning component actions are custom actions that invoke a Lightning component. They support Apex and JavaScript and provide a secure way to build client-side custom functionality. Lightning component actions are supported only in the Salesforce mobile app and Lightning Experience.

REQUIRED EDITIONS

Available in: both the Salesforce mobile app and Lightning Experience

Available in: **Essentials, Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, and Developer Editions**

You can add Lightning component actions to an object's page layout using the page layout editor, or to the global publisher layout using the Global Publisher Layout editor. If you have Lightning component actions in your org, you can find them in the Mobile & Lightning Actions category in the page layout editor's palette.

Lightning component actions can't call just any Lightning component in your org. For a component to work as a Lightning component action, it must be configured for that purpose and implement either the `force:LightningQuickAction` or `force:LightningQuickActionWithoutHeader` interfaces. You must also set a default value for each component attribute marked as required.

If you plan on packaging a Lightning component action, the component the action invokes must be marked as `access=global`.

[Create a Lightning Component Action](#)

Creating a Lightning component action is similar to creating a regular quick action, and you do it in the

same place in Setup. All you need is a Lightning component in your org for the quick action to trigger.

See Also

[Lightning Aura Components Developer Guide](#)

[Actions in Lightning Experience](#)

[Quick Action Considerations](#)

Create a Lightning Component Action

Creating a Lightning component action is similar to creating a regular quick action, and you do it in the same place in Setup. All you need is a Lightning component in your org for the quick action to trigger.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

USER PERMISSIONS NEEDED

To create actions:	Customize Application
--------------------	-----------------------

1. In Setup, click **Object Manager**, click the object that you want to create the action for, and click **Buttons, Links, and Actions**.
2. Click **New Action**.
3. For Action Type, select **Lightning Component**.
4. Select the component that you want the action to call.
5. Enter a label for the action. Users see this label as the name of the action.



Tip You can choose an option from the Standard Label Type list to have Salesforce generate the label. For the labels in this list that include “Record” and “Record Type,” Salesforce fills in the type of object or the record type the action creates. For example, if you choose the Create New “Record” standard label on a create contact action, the generated label is Create New Contact.

6. If necessary, change the name of the action. If you selected a standard label type in the previous step, you must enter the name.

This name is used in the API and managed packages. It must begin with a letter and use only alphanumeric characters and underscores, and it can't end with an underscore or have two consecutive underscores. Unless you're familiar with working with the API, we suggest not editing this field.

7. Type a description for the action.

The description appears on the detail page for the action and in the list on the Buttons, Links, and Actions page. The description isn't visible to your users. If you're creating several actions on the same object, we recommend using a detailed description, such as “Create Contact on Account using New Client record type.”

8. Optionally, click **Change Icon** to select a different icon for the action.

When you're finished creating your action, add it to the Salesforce Mobile and Lightning Experience Actions section of the desired page layout, such as the case page layout, and your users can start using it!

See Also

[Add Quick Actions to the Case Page Layout for Lightning Experience](#)

Lightning Web Component Actions

Lightning web component (LWC) actions are custom quick actions that invoke a Lightning web component. They support JavaScript and provide a secure way to build client-side functionality. LWC actions are supported only on record pages in Lightning Experience.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Essentials, Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, and Developer Editions**

You can add LWC actions to a record's page layout using Dynamic Actions in the Lightning App Builder. If you have LWC actions in your org, you can find them in the Lightning Actions category in the page layout editor's palette.

For a component to work as an LWC action, it must be configured for that purpose and implement either the `Action` or `ScreenAction` target. For more information, see [Configure a Component for Quick Actions](#) in the Lightning Web Components Developer Guide.

 **Note** Lightning Web Component actions don't work on external objects because those objects are built on the Aura framework.

[Create a Lightning Web Component Action](#)

Creating a Lightning web component (LWC) action is similar to creating a regular quick action, and you do it in the same place in Setup for most supported objects. All you need is an LWC in your org for the quick action to trigger.

See Also

[Lightning Web Components Developer Guide](#)

[Actions in Lightning Experience](#)

[Quick Action Considerations](#)

Create a Lightning Web Component Action

Creating a Lightning web component (LWC) action is similar to creating a regular quick action, and you do it in the same place in Setup for most supported objects. All you need is an LWC in your org for the quick action to trigger.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

USER PERMISSIONS NEEDED

To create actions: Customize Application

You can create a Lightning web component action for custom objects and for Salesforce record home objects that are enabled for LWC. If you can't create actions with the Lightning Web Component action type, they're not currently supported for that object. For more information, see [LWC Migration for Record Home Pages](#) in the *Lightning Web Components Developer Guide*.

-  **Note** When feed tracking is enabled for cases, you can't use the page layout to add a Lightning web component action to the Highlights Panel of a case record. Add Lightning web component actions to a case record page's Highlights Panel as dynamic actions in Lightning App Builder.

1. From Setup, click **Object Manager**, click the object that you want to create the action for, and then click **Buttons, Links, and Actions**.
2. Click **New Action**.
3. For Action Type, select **Lightning Web Component**.
4. Select the component that you want the action to call.
5. Enter a label for the action. Users see this label as the name of the action.



Tip You can choose an option from the Standard Label Type list to have Salesforce generate the label. For the labels in this list that include Record and Record Type, Salesforce fills in the type of object or the record type the action creates. For example, if you choose the Create New Record standard label on a create contact action, the generated label is Create New Contact.

6. If necessary, change the name of the action. If you selected a standard label type in the previous step, you must enter the name.

This name is used in the API and managed packages. It must begin with a letter and use only alphanumeric characters and underscores, and it can't end with an underscore or have two consecutive underscores. Unless you're familiar with working with the API, we suggest not editing this field.

7. Type a description for the action.

The description appears on the detail page for the action and in the list on the Buttons, Links, and Actions page. The description isn't visible to your users. If you're creating several actions on the same object, we recommend using a detailed description, such as "Create Contact on Account using New Client record type."

8. Optionally, to select a different icon for the action, click **Change Icon**.

When you finish creating your action, add it to the Lightning Experience Actions section of the desired page layout, such as the opportunity page layout, and your users can start using it.

 **Note** Headless Lightning web component quick actions are available only on record pages in Lightning Experience and LWR sites created in Experience Cloud. They're not supported in Aura Experience Builder sites or on the Salesforce mobile app. Orgs with the Salesforce Field Service (SFS) mobile app support Lightning web component actions on additional objects. These actions appear only in the SFS mobile app and not in Lightning Experience on mobile or desktop.

See Also

- [Lightning Web Components Developer Guide: Use Quick Actions](#)
- [Add Quick Actions to the Case Page Layout for Lightning Experience](#)
- [Create Dynamic Actions in Lightning App Builder](#)

Start a Flow from a Quick Action

With this feature, users can run a screen flow with the click of a button on a Lightning record page. Create a quick action on an object and configure it to run an active screen flow. Then, add the quick action button to the Lightning record page layout.

REQUIRED EDITIONS

[View supported editions.](#)

USER PERMISSIONS NEEDED

To create quick actions:	Customize Application
To create flows:	Manage Flows

For instructions on how to complete this task, see [Run a Flow from a Quick Action on a Record Page](#)

Visualforce Pages as Object-Specific Custom Actions

A Visualforce page added as a custom action on an object is invoked in the context of a record of that object type. The custom action is passed a specific record ID—the record the user was looking at when the user clicked the custom action. Design the page to act on that specific record type.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

Visualforce pages you create to use as object-specific actions must use a standard object controller. Use controller extensions to add custom code, including `@RemoteAction` methods you can call using JavaScript remoting.

Your custom code could do more than make updates to the originating record. For example, the Create Quick Order custom action searches for matching merchandise. It then creates an invoice and line item, all as part of creating an order for a part. That logic occurs in the context of the originating account record—the invoice is related to the account record where the quick order action was invoked.

The following code sample shows a page designed to be used as a custom action on the account object, so it uses the standard Account controller. This action lets users create cases from account detail pages, and it has a different user interface from standard create actions.

```
public with sharing class CreateCaseExtension {  
    private final SObject parent;  
    public Case theCase {get; set;}  
    public String lastError {get; set;}  
  
    public CreateCaseExtension2(ApexPages.StandardController controller) {  
        parent = controller.getRecord();  
        theCase = new Case();  
        theCase.accountId = parent.id;  
        lastError = '';  
    }  
  
    public PageReference createCase() {  
        createNewCase();  
        theCase = new Case();  
        theCase.accountId = parent.id;  
        return null;  
    }  
  
    private void createNewCase() {  
        try {  
            insert theCase;  
  
            FeedItem post = new FeedItem();  
            post.ParentId = ApexPages.currentPage().getParameters().get('id');  
            post.Body = 'created a case';  
            post.type = 'LinkPost';  
            post.LinkUrl = '/' + theCase.id;  
            post.Title = theCase.Subject;  
            insert post;  
        } catch(System.Exception ex){  
            lastError = ex.getMessage();  
        }  
    }  
}
```

```
<apex:page standardcontroller="Account" extensions="CreateCaseExtension" showHeader="false">
    <script type='text/javascript' src='/canvas/sdk/js/publisher.js'/>
    <style>
        .requiredInput .requiredBlock, .requiredBlock {background-color: white;}
        .custompubblock div {display: inline-block;}
        .custompublabel {width:54px;}
    </style>
    <script>
        function refreshFeed() {
            Sfdc.canvas.publisher.publish({name : 'publisher.refresh', payload : {feed:true}});
        }
    </script>
    <div>
        <apex:form >
            <apex:actionFunction action="{!!createCase}" name="createCase" render="out"
                oncomplete="refreshFeed();"/>
            <apex:outputPanel id="out" >
                <div class="custompubblock">
                    <div class="custompublabel">Account:</div><apex:inputField value="{!!theCase.accountId}" style="margin-left:0;"/>&ampnbsp&ampnbsp&ampnbsp
                    <div>Contact:&ampnbsp</div><apex:inputField value="{!!theCase.contactId}" />
                </div>
                <apex:inputField value="{!!theCase.description}" style="width:538px;height:92px;margin-top:4px;" />
                <div class="custompubblock" style="margin-top:5px;">
                    <div>Status:&ampnbsp</div><apex:inputField value="{!!theCase.status}" />&ampnbsp&ampnbsp&ampnbsp
                    <div>Priority:&ampnbsp</div><apex:inputField value="{!!theCase.priority}" />&ampnbsp&ampnbsp&ampnbsp
                    <div>Case Origin:&ampnbsp</div><apex:inputField value="{!!theCase.origin}" />
                </div>
            </apex:outputPanel>
        </apex:form><br/>
        <button type="button" onclick="createCase();"
            style="position:fixed;bottom:0px;right:0px;padding:5px 10px;
            font-size:13px; font-weight:bold; line-height:18px;background-color:#0271BF;background-image:-moz-linear-gradient(#2
            
```

```
DADD, #0271BF);background-repeat:repeat-x; border-color:#096EB3;"  
    id="addcasebutton">Create Case</button>  
</div>  
</apex:page>
```

-  **Note** When you redirect to a URL internal to your org, the action dialog closes upon completion or programmatically navigating away. If you set up the redirect to point to an external URL, the behavior can vary because an external URL opens in a new browser tab.

Requirements for Refreshing Host Pages

If you want an object-specific or global custom action to refresh the feed on the page that hosts it, the Visualforce page you create to use as that action must:

- Reference the publisher JavaScript file: `<script type='text/javascript' src='/canvas/sdk/js/publisher.js' />`. (Creating custom Visualforce actions doesn't require the Canvas SDK.)
- Include this JavaScript call: `sfdc.canvas.publisher.publish({name : 'publisher.refresh', payload : {feed:true}});`.

Visualforce Pages as Global Custom Actions

Visualforce pages used as global actions can be invoked in many different places and don't have a specific record associated with them. They have complete freedom of action, which means it's up to you to write the code.

Hide the Action Header for Visualforce Custom Actions

When creating a Visualforce page to use as a custom action, you can choose to hide the action's header. Hiding the action header helps prevent user confusion, especially if you have your own buttons specified in the Visualforce page.

See Also

[Visualforce Pages as Global Custom Actions](#)

[Create Object-Specific Quick Actions](#)

[Quick Actions](#)

Visualforce Pages as Global Custom Actions

Visualforce pages used as global actions can be invoked in many different places and don't have a specific record associated with them. They have complete freedom of action, which means it's up to you to write the code.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager,**

Database.com, and Developer Editions

Visualforce pages you create to use as global actions can't use a standard object controller. You must write a custom controller to handle the page.

When a global action completes, the user is either redirected to a parent record created as part of the action or returned to where they started.

This code sample shows a Visualforce page designed to be used as a custom action on any object that supports actions. This action lets users create cases from record detail pages, Chatter, Chatter groups (except customer groups), or the home page. It has a different user interface from standard create actions. As with all global actions, the records created through this action aren't associated with other records.

```
<!-- Custom controller -->
public with sharing class CreateCaseController {
    public Case theCase {get; set;}
    public String lastError {get; set;}
    public CreateCaseController() {
        theCase = new Case();
        lastError = '';
    }

    public PageReference createCase() {
        createNewCase();
        theCase = new Case();
        return null;
    }

    private void createNewCase() {
        try {
            insert theCase;

            FeedItem post = new FeedItem();
            post.ParentId = ApexPages.currentPage().getParameters().get('id');
            post.Body = 'created a case';
            post.type = 'LinkPost';
            post.LinkUrl = '/' + theCase.id;
            post.Title = theCase.Subject;
            insert post;
        } catch(System.Exception ex) {
            lastError = ex.getMessage();
        }
    }
}
```

```
}
```

```
<apex:page controller="CreateCaseController" showHeader="false">
    <script type='text/javascript' src='/canvas/sdk/js/publisher.js' />
    <style>
        .requiredInput .requiredBlock, .requiredBlock {background-color: white;}
        .custompubblock div {display: inline-block;}
        .custompublabel {width:54px;}
    </style>

    <script>
        function refreshFeed() {
            Sfdc.canvas.publisher.publish({name : 'publisher.refresh', payload : {feed: true}});
        }
    </script>

<div>
    <apex:form >
        <apex:actionFunction action="{!createCase}" name="createCase" render="out" oncomplete="refreshFeed();"/>
        <apex:outputPanel id="out" >
            <div class="custompubblock">
                <div>Subject: </div><apex:inputField value="{!!theCase.subject}" style="width:500px;" />
            </div>
            <div class="custompubblock">
                <div class="custompublabel">Account:</div><apex:inputField value="{!!theCase.accountId}" value="margin-left:0;"/>&nbsp;&nbsp;&nbsp;
                <div>Contact: </div><apex:inputField value="{!!theCase.contactId}" />
            </div>
            <apex:inputField value="{!!theCase.description}" style="width:500px;height:92px;margin-top:4px;" />
            <div class="custompubblock" style="margin-top:5px;">
                <div>Status: </div><apex:inputField value="{!!theCase.status}" />&nbsp;&nbsp;&nbsp;
                <div>Priority: </div><apex:inputField value="{!!theCase.priority}" />&nbsp;&nbsp;&nbsp;
                <div>Case Origin: </div><apex:inputField value="{!!theCase.origin}" />
            </div>
        </apex:outputPanel>
    </apex:form>
</div>
```

```
</div>
<div style="color:red;">{!!lastError}</div>
</apex:outputPanel>
</apex:form><br/>
<button type="button" onclick="createCase();" style="position:fixed;bottom:0px;right:0px;padding:5px 10px; font-size:13px; font-weight:bold; line-height:18px;background-color:#0271BF;background-image:-moz-linear-gradient(#2DADD, #0271BF);background-repeat:repeat-x; border-color:#096EB3;" id="addcasebutton">Create Case</button>
</div>
</apex:page>
```

Requirements for Refreshing Host Pages

To have an object-specific or global custom action refresh the feed on the page that hosts it, the Visualforce page you create to use as that action must:

- Reference the publisher JavaScript file: `<script type='text/javascript' src='/canvas/sdk/js/publisher.js' />`. (Creating custom Visualforce actions doesn't require the Canvas SDK.)
- Include this JavaScript call: `Sfdc.canvas.publisher.publish({name : 'publisher.refresh', payload : {feed:true}});`.

See Also

- [Visualforce Pages as Object-Specific Custom Actions](#)
- [Create Global Quick Actions](#)
- [Quick Actions](#)

Hide the Action Header for Visualforce Custom Actions

When creating a Visualforce page to use as a custom action, you can choose to hide the action's header. Hiding the action header helps prevent user confusion, especially if you have your own buttons specified in the Visualforce page.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

To hide the header, add `showQuickActionVfHeader="false"` to the `<apex:page>` tag of the custom action's Visualforce page. When the Visualforce custom action renders in the Salesforce mobile app, the header and the Cancel and Save buttons are hidden. Using this attribute doesn't affect how the

action displays in the full Salesforce site.

If you don't specify `showQuickActionVfHeader`, its value defaults to `true`.

The `showQuickActionVfHeader` attribute isn't supported in Experience Cloud sites.

See Also

[Visualforce Pages as Object-Specific Custom Actions](#)

[Visualforce Pages as Global Custom Actions](#)

Prerequisites for Using Canvas Apps as Custom Actions

Using canvas apps as custom actions makes it easy to give users access to the functionality of your apps in Chatter and elsewhere in Salesforce.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Quick actions available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer** Editions

Custom canvas actions available in: **Professional** (with Canvas enabled), **Enterprise, Performance, Unlimited, and Developer** Editions

You can use as a custom action any canvas app that uses Publisher as a location. For example, you can use an expense report app as a custom action to make it easy for salespeople to submit expense reports directly from feeds. Or a custom action that includes a video conferencing canvas app could help support agents communicate with customers visually for easier troubleshooting of technical issues.

Before creating a custom action with a canvas app, be sure that the app uses Publisher as a location. Also ensure that you give the users who you want to be able to use the action access to the app.

See Also

[Visualforce Pages as Object-Specific Custom Actions](#)

[Quick Actions](#)

Enable Actions in the Chatter Publisher

Enabling actions in the publisher lets you add actions that you've created to the Chatter publisher. Actions appear on the Home page, on the Chatter tab, in Chatter groups, and on record detail pages.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

USER PERMISSIONS NEEDED

To enable actions in the Chatter publisher: Customize Application

By default, the Salesforce Classic Chatter publisher includes the standard actions Post, File, Link, Poll, Question, and Thanks. With the actions in the publisher setting enabled, you can include nonstandard actions in the Chatter publisher too. Nonstandard actions include Create, Update, Log a Call, custom actions, and Mobile Smart Actions.

In organizations created after the Winter '14 release, actions in the publisher is enabled automatically.

1. From Setup, enter *Chatter Settings* in the **Quick Find** box, then select **Chatter Settings**.

2. Click **Edit**.

3. Select **Enable Actions in the Publisher**.

Although the Enable Actions setting appears in Lightning Experience, it has no effect there.

4. Click **Save**.

 **Note** You aren't required to enable actions in the publisher to use them in the Salesforce app or in third-party apps. See [Actions With and Without Chatter](#) for more information.

Set Up Actions with Chatter Enabled

Global and object-specific actions enhance the Chatter experience for your users in Salesforce Classic and in the Salesforce app.

Set Up Actions Without Chatter Enabled

You can set up object-specific and global actions for the Salesforce mobile app or third-party apps, even if your org doesn't have Chatter enabled.

Actions With and Without Chatter

Use actions regardless of whether Chatter or actions in the publisher are enabled.

Set Up Actions with Chatter Enabled

Global and object-specific actions enhance the Chatter experience for your users in Salesforce Classic and in the Salesforce app.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

USER PERMISSIONS NEEDED

To set up actions:

Customize Application

Chatter must be enabled for your organization.

By default, the Chatter publisher in Salesforce Classic includes the standard actions Post, File, Link, Poll, Question, and Thanks. To set up more actions in Chatter:

1. Enable feed tracking for the objects for which you want to make actions available.
2. Enable actions in the publisher if you want to see both standard actions and non-standard actions in the Chatter publisher.
3. Optionally, enable feed updates for related records to display feed items on a record detail page when related records are created.
4. Create object-specific actions or global actions.
5. Customize the action layout with the fields that you want users to see when they use the action.
6. Add the actions to page layouts or global publisher layouts.

Salesforce automatically adds default actions to the page layouts for account, case, contact, lead, and opportunity, and to the global publisher layout in organizations that were created after Winter '14.

See Also

- [Enable Actions in the Chatter Publisher](#)
[Create Object-Specific Quick Actions](#)
[Create Dynamic Actions in Lightning App Builder](#)

Set Up Actions Without Chatter Enabled

You can set up object-specific and global actions for the Salesforce mobile app or third-party apps, even if your org doesn't have Chatter enabled.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

USER PERMISSIONS NEEDED

To set up actions:

Customize Application

When Chatter is disabled in an org, only the nonstandard actions appear in the action bar in the Salesforce mobile app or in third-party apps that use action lists. Nonstandard actions include Create, Update, Log a Call, custom actions, and Mobile Smart Actions.

 **Note** You can create actions in Salesforce Classic when Chatter is disabled. Those actions don't

appear in Salesforce Classic, but can appear in Lightning Experience and the Salesforce mobile app if you don't customize the Salesforce Mobile and Lightning Experience Actions section of the page layout editor.

Follow these steps to set up actions for use in the Salesforce mobile app or third-party apps.

1. Create object-specific actions or global actions.
2. Customize the action layout with the fields that you want users to see when they use the action.
3. Add the actions to page layouts or global publisher layouts.

Salesforce automatically adds default actions to the page layouts for account, case, contact, lead, and opportunity, and to the global publisher layout in organizations that were created after Winter '14.

See Also

- [Create Object-Specific Quick Actions](#)
- [Create Global Quick Actions](#)
- [Customize Actions with the Enhanced Page Layout Editor](#)

Actions With and Without Chatter

Use actions regardless of whether Chatter or actions in the publisher are enabled.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Quick actions available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer** Editions

Custom canvas actions available in: **Professional** (with Canvas enabled), **Enterprise, Performance, Unlimited, and Developer** Editions

The actions that are available in the full Salesforce site (Salesforce Classic and Lightning Experience) or in the Salesforce mobile app depend on your org's settings. To enable or disable Chatter for your organization, from Setup, enter *Chatter Settings* in the **Quick Find** box, then select **Chatter Settings**. If Chatter is enabled, the **Enable Actions in the Publisher** option controls whether the actions that you create display in the Chatter publisher.

	Chatter Off, Actions Off	Chatter On, Actions Off	Chatter On, Actions On
You can create global actions and customize global action lists	Yes	Yes	Yes
You can create object-specific	Yes	Yes	Yes

	Chatter Off, Actions Off	Chatter On, Actions Off	Chatter On, Actions On
actions and customize object-specific action lists			
Actions appear on the Home page and Chatter home page in the full Salesforce site	No	Yes ¹	Yes
Actions appear in object feeds in the full Salesforce site	No	Yes ^{1,2}	Yes ²
The action bar is available in the Salesforce mobile app feed	No ³	Yes ⁴	Yes
The action bar is available on the record view in the Salesforce mobile app	Yes ⁵	Yes ⁶	Yes ⁶
The action bar is available on Lightning pages in the Salesforce mobile app	Yes ⁵	Yes	Yes

Footnotes:

1. If actions in the publisher aren't enabled, only standard Chatter actions (Post, File, Link, Poll, and Thanks) appear in the Chatter publisher in the full Salesforce site.
2. The Chatter feed appears on an object's detail page in the full Salesforce site only for objects that have feed tracking enabled.
3. When Chatter is disabled, the Feed item isn't available in the Salesforce mobile app.
4. When Chatter is enabled but actions in the publisher aren't, standard Chatter actions and nonstandard actions appear in the Salesforce mobile app action bar and in third-party apps that use action lists. Nonstandard actions include Create, Update, Log a Call, custom actions, and Mobile Smart Actions.
5. When Chatter and actions in the publisher are disabled, only nonstandard actions appear in the action bar in the Salesforce mobile app or in third-party apps that use action lists. Nonstandard actions include Create, Update, Log a Call, custom actions, and Mobile Smart Actions.
6. If feed tracking isn't enabled on an object, only nonstandard actions appear in the Salesforce mobile app action bar and in third-party apps that use action lists. Nonstandard actions include Create, Update, Log a Call, custom actions, and Mobile Smart Actions.

See Also

- [Set Up Actions with Chatter Enabled](#)
- [Set Up Actions Without Chatter Enabled](#)
- [Quick Actions](#)

Action Layout Editor

Just as object record pages have page layouts that can be customized, actions have action layouts that can be customized. When you create an action, Salesforce populates its layout with a default set of fields. You can add, remove, or reorder fields on the action layout to present only the essential items your users need when they use the action.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

For example, a Post action needs just one large text area for user input. By contrast, an object-related Create action must include multiple fields; otherwise, Salesforce can't create the record.

Find the Action Layout Editor

To view and edit the layouts for global actions, from Setup, in the Quick Find box, enter *Actions*, and then select **Global Actions**. Next to the action's name, click **Layout**.

To view and edit the layouts for object-specific actions, from the object management settings for the object whose action layout you want to see, go to Buttons, Links, and Actions.

Fields in the Action Layout Editor

The upper part of the action layout editor is the palette, and below the palette is the action layout. The palette contains fields from the action's target object that you can add to the action layout, except for these unsupported field types.

- Record type fields
- Read-only field types such as roll-up summary, formula, and auto-number fields
- Read-only system fields such as Created By or Last Modified By
- When you create a custom action and the action type is Update a Record, you can't add the Owner field to the action layout for most objects. The exception is Case records. You can add the Case Owner field to your action layout when you create a custom action to update a Case record.

The first time that you view the layout for an action you've created, certain fields are prepopulated.

- Target object default fields
- Standard required fields
- Any custom universally required fields

Default actions (available in organizations created after Winter '14) have [predefined sets of fields](#).

Inactive Fields

Fields that are already on the action layout still appear on the palette but are inactive. When you select an inactive field on the palette, Salesforce highlights the field on the action layout.

Field Type Conversion

If you convert a field's type from one that is supported for actions to a type that isn't supported, Salesforce removes the field from the action layout. If you convert the field back to a supported type without changing the action layout, Salesforce automatically adds the field back to the layout. If you edit the layout, and then convert the field back to a supported type, add the field back to the layout manually.

Layouts Used for Log a Call Actions

A Log a Call action takes the active task page layout except under these conditions.

- If your organization has a custom Log a Call action for a specific object, the custom action takes the custom action layout defined for it.
- If your organization has a custom Log a Call global action, the action takes the custom layout defined for it, unless you also have a custom Log a Call action for an object. A custom object-specific action overrides a custom global action.

To show the simpler New Task form to Salesforce mobile app users, enable the form in Activity Settings and ensure that the layout used includes a subject field.

Layout Auditing

Salesforce tracks action layout customization in the setup audit trail history.

Action Layout Editor Considerations

When you create an action, Salesforce populates its layout with a default set of fields. You can add, remove, or reorder fields on the action layout to present only the essential items your users need when they use the action.

See Also

[Quick Actions](#)

Action Layout Editor Considerations

When you create an action, Salesforce populates its layout with a default set of fields. You can add, remove, or reorder fields on the action layout to present only the essential items your users need when they use the action.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

- There's no hard limit to the number of fields you can add to an action layout. However, for optimum usability, we recommend a maximum of 8 fields. Adding more than 20 fields can severely impact user efficiency. To reduce the number of fields in your layout, you can [create predefined values](#) for the required fields, and then remove those fields from your layout. You can set predefined field values from the action detail page.
- By default, the Call productivity action in the Salesforce mobile app action bar uses the global Log A Call action layout. If you create one Log a Call action on an object, users see that custom Log a Call action's layout when they tap **Call** from the action bar for that object. If you create more than one Log a Call action for the same object, the Call action on that object doesn't know which layout to use, so it uses the global Log a Call action layout.
- Mobile smart actions are populated with all your org's required fields on the relevant object, regardless of how many fields there are. For example, the New Case action in the mobile smart action bundle includes all required case fields. You can't edit the fields on mobile smart actions. The fields that appear change only if you change which fields on an object are required.
- You can remove a required field from the action layout, but make sure that the field has a predefined value. Otherwise, users can't create records.
- Rich text area fields are supported only when you add them to one-column layouts, or as fields that span both columns in two-column layouts. If you add a rich text area field to only one column in a two-column layout, it appears as a plain text area, because there's not enough space to display the full rich text editor.
- When you create a custom action and the action type is Update a Record, you can't add the Owner field to the action layout for most objects. The exception is Case records. You can add the Case Owner field to your action layout when you create a custom action to update a Case record.
- When an Activities component is placed in a sidebar region on a Lightning page, global action fields always appear in a single column, even if the global action has a two-column layout.

Custom Success Messages for Quick Actions

For Create a Record, Update a Record, and Log a Call action types, you can create a custom message that displays when the action executes successfully.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

The screenshot shows the 'Enter Action Information' dialog box. At the top are 'Save' and 'Cancel' buttons. Below are fields: 'Object Name' set to 'Account', 'Action Type' set to 'Create a Record', 'Target Object' set to '--None--', 'Standard Label Type' set to '--None--', 'Label' (empty), 'Name' (empty), 'Description' (empty), 'Create Feed Item' (checkbox checked), 'Success Message' (text input field containing '[REDACTED]' and highlighted with a red border), and 'Icon' (link to 'Change Icon').

In the Salesforce mobile app and Lightning Experience, when a create, update, or Log a Call action is completed, a default success message displays, regardless of whether the action created a feed item. If you add a custom success message to one of these actions, your custom success message displays instead of the default message.

In Salesforce Classic, custom success messages have slightly different behavior. If you select **Create Feed Item** for a Create a Record or Log a Call action, no success message displays in Salesforce Classic. The feed item itself is the confirmation that the action executed successfully.

You can configure translations for custom success messages through the Translation Workbench. From Setup, enter *Translate* in the Quick Find box, and then select **Translate**. Choose **Action** for the Setup Component, and choose **Informational Message** for the Aspect.

-  **Note** If you have **All Related Objects** selected under feed tracking for a particular object, when a quick action creates related objects, a feed item is always created, regardless of the status of Create Feed Item.

See Also

- [Create Object-Specific Quick Actions](#)
- [Create Global Quick Actions](#)

Customize Actions with the Enhanced Page Layout Editor

Use the page layout editor to customize which actions show up in Salesforce and in the Salesforce mobile app.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

USER PERMISSIONS NEEDED

To create actions:	Customize Application
To customize action layouts and page layouts:	Customize Application
To view page layouts:	View Setup

 **Tip** To manage the actions for global pages, such as Home, Chatter Home, and Chatter groups, see [Global Publisher Layouts](#).

From the management settings for the object whose actions you want to manage, go to Page Layouts.

You can add actions to two sections on a page layout:

Quick Actions in the Salesforce Classic Publisher	This section can contain actions only from the Quick Actions category in the palette. Actions in this section appear in the Chatter publisher in Salesforce Classic.
Salesforce Mobile and Lightning Experience Actions	This section can contain actions only from the Mobile & Lightning Actions category in the palette. On object page layouts, the Mobile & Lightning Actions category contains all available types of actions for the object, including quick actions, productivity actions, Lightning component actions, and standard and custom buttons. Actions in this section appear in the action bar and action menu in the Salesforce mobile app and in various areas of Lightning Experience.

 **Tip** Hover over an action in the palette to see its label, API name, and action type.

- To override the action defaults for an action section that you haven't customized, either click the override text or hover over the section and click .
- If you haven't customized the Quick Actions in the Salesforce Classic Publisher section of a page layout, the actions that appear in the publisher for that object default to the actions that are assigned to the global publisher layout. Upon overriding, the actions default to the standard actions—Post, File, Link, Poll, Question, and Thanks—regardless of what actions were assigned to the global publisher layout.
- If you haven't customized the Salesforce Mobile and Lightning Experience Actions section of a page layout, the actions for that object default to a set of predefined actions. If you have customized actions in the Quick Actions in the Salesforce Classic Publisher section, and have saved the layout, the Salesforce Mobile and Lightning Experience Actions section inherits the actions from the Quick

Actions in the Salesforce Classic Publisher section, plus any standard or custom buttons present on the layout, when you click to override.

- To revert the actions in either section to the defaults for that section, hover over the section and click .

See Also

- [Set Up Actions with Chatter Enabled](#)
- [Actions in Lightning Experience](#)
- [Send Email Action Considerations for Cases](#)
- [Mobile Smart Actions](#)
- [The Enhanced Page Layout Editor](#)
- [Find Object Management Settings](#)

Set Predefined Field Values for Quick Action Fields

When you create actions, use predefined field values to set a value for a field. Predefined values can help ensure consistency and make it faster and easier for users to create records.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

USER PERMISSIONS NEEDED

To set predefined field values: Customize Applications

When you configure action layouts, it's better to use fewer fields. Most users, especially mobile users, don't like to fill in too many required fields. They want to get things done and move on to their next task. A good way to use fewer fields in action layouts is to set predefined values for as many fields as possible. The more fields you can set predefined values for, the more you can remove from the layout and make the action easier and quicker to use. Balance ease of use with the need for required information. However, don't remove required fields from an action layout without setting a predefined value for those fields, or when a user applies that action. The record doesn't save properly.

If you set predefined values for fields on object records created through an action, you don't need to add those fields to the action layout. For example, when you configure an action that lets users create opportunities, set Prospecting as the predefined value for the Stage field. All new opportunities users create through that action are automatically assigned to the prospecting stage. You can remove the Stage field from the action's layout, because the field is going to be assigned a value automatically.

 **Tip** Predefined values for fields on actions are different from default values that you can set for fields on records. If a field is included in an action, it can have both a predefined value set for the action *and* a default value set.

1. Click the name of an action in the Buttons, Links, and Actions list or the Global Actions list.

2. On the action detail page, click **New** in the Predefined Field Values list.

3. Select the field you want to predefine a value for.

4. Specify the value for the field.

For single-select picklists, you can specify both a specific value and a formula value. If you set both, the formula value takes precedence over the specific value.

5. Click **Save**.

 **Tip** On object-specific actions, the predefined value can include references to the source object and its related objects.

Notes on Predefined Field Values for Quick Actions

Setting predefined field values for quick actions is especially important if you remove required or Always on Layout fields from the action layout.

See Also

[Notes on Predefined Field Values for Quick Actions](#)

Notes on Predefined Field Values for Quick Actions

Setting predefined field values for quick actions is especially important if you remove required or Always on Layout fields from the action layout.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

- You can set predefined values for any field available in the action layout editor, with these exceptions.
 - Dependent picklists
 - Multi-select picklists
 - Read-only field types like auto-number, formula, and roll-up summary fields
- If you set a predefined value for the lookup field that's used as the action's relationship field, the action ignores the predefined value and uses the parent record ID instead.
- You can't use a dependent picklist to set a predefined value.
- If a field on an action has both a predefined value and a default value set, the action uses the predefined value, not the default value.
- If a picklist field on a quick action has both a predefined specific value and a predefined formula value, the formula value takes precedence over the specific value.
- If you set a predefined value for a field and leave it on the action layout, that value displays as the default value for the field.
- If you use a rich text field as a predefined field value, any text formatting that was applied in the rich text field is removed in the new field.

- When you have a required field with a predefined value and remove the field from the action layout, you must add the required field back to the action layout if you later delete the field's predefined value. Otherwise, users can't save the record.
- Formulas can't reference fields on external objects, so you can't reference an external object field to set a predefined field value for a quick action.
- You can use a `TEXT (picklist)` value in a predefined value field.
- A predefined field value for a lookup field must be a reference to a record Id field or another lookup field.
- To set the value for an email action's recipient, predefine the To, CC, or BCC fields. You can use ID fields, such as **Contact.Id**, or string fields, such as **Contact.custom_email_field** to predefine the value.
 - Contact, lead, and person account IDs are supported.
 - Use an ID field if you want to log the email on the recipient's record.
 - Use a string field to predefine an email recipient for a custom object or custom field.
 - If you use a string field, the email isn't logged on the recipient's record.

See Also

[Set Predefined Field Values for Quick Action Fields](#)

Global Publisher Layouts

Global publisher layouts determine the global actions that appear in the various Salesforce interfaces. In Salesforce Classic and Lightning Experience, these layouts customize the actions on global pages (like the Home page) and on the Chatter page. Lightning Experience also uses these layouts to populate the Global Actions menu. And in the Salesforce mobile app, these layouts drive the actions that appear in the action bar on the Feed and People pages. Global publisher layouts can include global actions only.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

USER PERMISSIONS NEEDED

To create actions:	Customize Application
To customize action layouts and page layouts:	Customize Application
To view page layouts:	View Setup

In Salesforce for Outlook, global publisher layouts drive the actions that Group, Contact Manager, and Professional Edition users see when they click the Salesforce Side Panel Publisher. Salesforce for Outlook users working in all other editions can [set up their side panel publishers using Outlook Side Panel Publisher Layouts](#).

 **Note** Chatter groups without customers display the global publisher layout by default, unless you

override it with a customized group publisher layout. In Chatter groups that allow customers, the publisher displays standard actions only, such as Post, File, Link, and Poll.

You can assign different global publisher layouts to different user profiles to customize which actions users see by default on global pages.

-  **Note** Changes to user layouts override the global publisher layout on user profile pages and the Chatter home page.

Follow these steps to work with global publisher layouts.

1. Create Global Publisher Layouts

Global publisher layouts determine the global actions that appear in the various Salesforce interfaces. In Salesforce Classic and Lightning Experience, these layouts customize the actions on global pages (like the Home page) and on the Chatter page. Lightning Experience also uses these layouts to populate the Global Actions menu. And in the Salesforce mobile app, these layouts drive the actions that appear in the action bar on the Feed and People pages. Global publisher layouts can include global actions only.

2. Add Actions to Global Publisher Layouts

Actions you add to the global publisher layouts appear on pages such as the Home and Chatter pages. They also appear on the action bar and action menu on the Feed and People pages in the Salesforce mobile app.

3. Assign Global Publisher Layouts to User Profiles

After you finish creating a global publisher layout, you can assign it to different user profiles. For example, a Marketing User and a Standard User need different actions in the Chatter publisher or in the Salesforce mobile app. Create multiple global publisher layouts and assign them to different user profiles to customize the actions for each profile.

See Also

[Create Global Quick Actions](#)

Create Global Publisher Layouts

Global publisher layouts determine the global actions that appear in the various Salesforce interfaces. In Salesforce Classic and Lightning Experience, these layouts customize the actions on global pages (like the Home page) and on the Chatter page. Lightning Experience also uses these layouts to populate the Global Actions menu. And in the Salesforce mobile app, these layouts drive the actions that appear in the action bar on the Feed and People pages. Global publisher layouts can include global actions only.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, Database.com, and Developer Editions**

USER PERMISSIONS NEEDED

To create actions:	Customize Application
To customize action layouts and page layouts:	Customize Application
To view page layouts:	View Setup

1. From Setup, in the Quick Find box, enter *Publisher Layouts*, and then select **Publisher Layouts**.
2. To create a new global publisher layout, click **New**.
3. To clone a publisher layout, select one from the Existing Global Publisher Layout dropdown list.
4. Enter a name for the new global publisher layout.
5. Save your changes.

See Also

[Assign Global Publisher Layouts to User Profiles](#)
[Global Publisher Layouts](#)

Add Actions to Global Publisher Layouts

Actions you add to the global publisher layouts appear on pages such as the Home and Chatter pages. They also appear on the action bar and action menu on the Feed and People pages in the Salesforce mobile app.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

USER PERMISSIONS NEEDED

To create actions:	Customize Application
To customize action layouts and page layouts:	Customize Application
To view page layouts:	View Setup

Arrange the actions so that the frequently used actions appear first in each list.

You can add actions to two sections on a page layout:

Quick Actions in the Salesforce Classic Publisher	This section can contain actions only from the Quick Actions category in the palette. Actions in this section appear in the Chatter publisher in
---	---

	Salesforce Classic.
Salesforce Mobile and Lightning Experience Actions	<p>This section can contain actions only from the Mobile & Lightning Actions category in the palette.</p> <p>On object page layouts, the Mobile & Lightning Actions category contains all available types of actions for the object, including quick actions, productivity actions, Lightning component actions, and standard and custom buttons.</p> <p>Actions in this section appear in the action bar and action menu in the Salesforce mobile app and in various areas of Lightning Experience.</p>

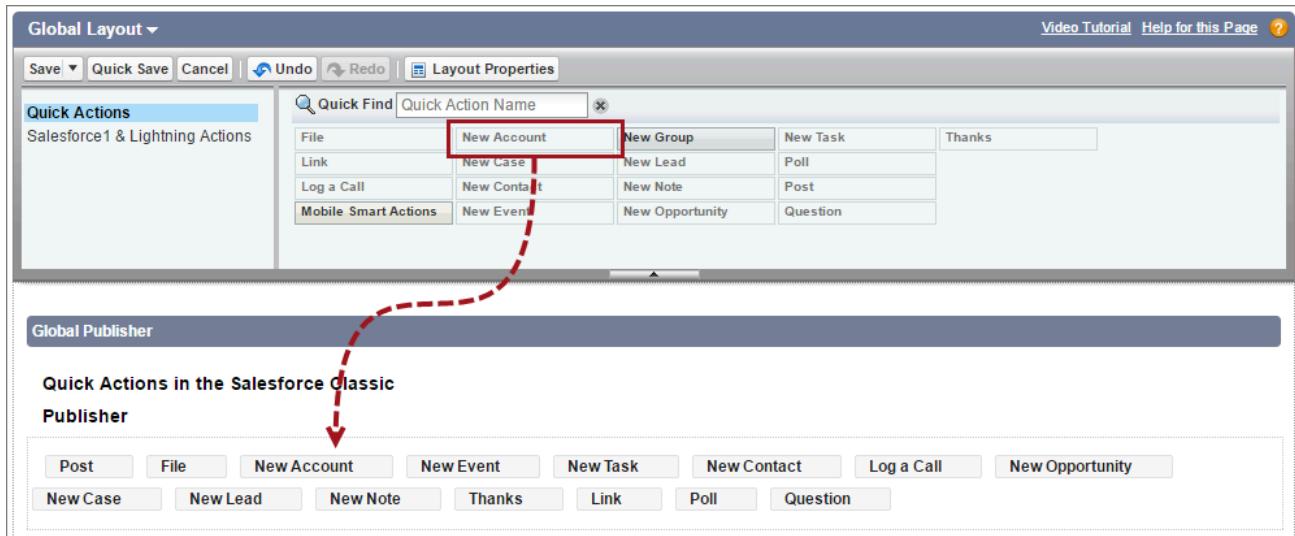
 **Note** Changes to user layouts override the global publisher layout on user profile pages and the Chatter home page. Actions on the user profile page come from the Quick Actions in the Salesforce Classic Publisher section of the global publisher layout. Only standard Chatter actions appear on the user profile page, regardless of which actions are present in the User Page Layout or the global publisher layout.

1. From Setup, in the Quick Find box, enter *Publisher Layouts*, and then select **Publisher Layouts**.
2. To add or remove actions, drag them to and from the palette.
3. To reorder actions, select an action and drag it to a new position.
4. Click **Save** when you're done, or click **Quick Save** to save your changes and continue working on the layout.

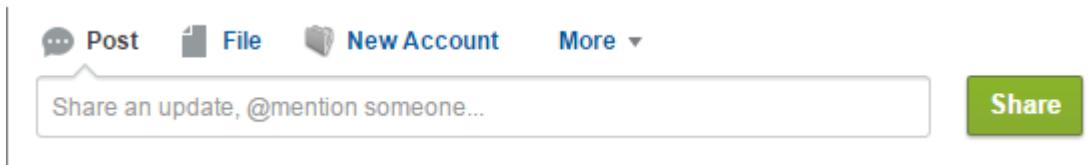
If you navigate away without saving, you lose your changes.

 **Note** Before using the personal email setting When you click an email address to compose an email, which email editor do you want to use?, confirm that the Global Publisher Layout has the email action in the **Salesforce Mobile and Lightning Experience Action** section.

 **Example** Let's add the New Account action to the publisher on the Home and Chatter pages in Salesforce Classic. The New Account action lets users create an account directly from the publisher. Drag the **New Account** action to the Quick Actions in the Salesforce Classic Publisher section and save your changes.



Go to the Chatter tab in Salesforce Classic. Now the New Account action shows up in the publisher.



Note The Chatter page in Lightning Experience supports only the standard Chatter actions Post, Poll, and Question, and if you have Groups, the Announcement action.

See Also

- [Actions in Lightning Experience](#)
- [Send Email Action Considerations for Cases](#)
- [Assign Global Publisher Layouts to User Profiles](#)
- [Email Application Publisher Layouts](#)

Assign Global Publisher Layouts to User Profiles

After you finish creating a global publisher layout, you can assign it to different user profiles. For example, a Marketing User and a Standard User need different actions in the Chatter publisher or in the Salesforce mobile app. Create multiple global publisher layouts and assign them to different user profiles to customize the actions for each profile.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

USER PERMISSIONS NEEDED

To create actions:

Customize Application

USER PERMISSIONS NEEDED

To customize action layouts and page layouts: Customize Application

To view page layouts: View Setup

1. From Setup, enter *Publisher Layouts* in the **Quick Find** box, then select **Publisher Layouts**.
2. Click **Publisher Layout Assignment**.
3. Click **Edit Assignment**.
4. Select a user profile by clicking anywhere on its row in the table.
5. From the Publisher Layout, select the global publisher layout that you want to assign to the highlighted profile.
6. Save the layout.

See Also

[Add Actions to Global Publisher Layouts](#)

[Global Publisher Layouts](#)

Quick Actions and Record Types

Using record types in your organization can affect the availability of quick actions for your users.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Quick actions available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

Custom canvas actions available in: **Professional** (with Canvas enabled), **Enterprise, Performance, Unlimited**, and **Developer** Editions

If users don't have access to a particular record type, actions that are assigned to that record type aren't available to them. For example, let's say that you have a page layout that contains a mix of actions—some have no record type assigned and some are assigned to Record Type A. Users without access to Record Type A see only the nonassigned actions when they visit the page.

! **Important** Don't assign actions to the Master record type. The Master record type is a placeholder record type that's assigned when your organization is created.

Default Global Actions: A Special Case

If you have default global actions in your organization and you're using record types, your users might not be able to see all the default actions that are assigned to a page layout.

Default global actions are assigned to the Master record type, which isn't accessible to most profiles. As a result, default global actions with the Master record type that are associated with target objects that have record types configured aren't available for most users. To fix this issue, edit the default global actions associated with those objects and reassign them to a different record type.

For example, the New Contact default global action has Contact as its target object. Let's say you have record types set up for the Contact object, and you add the New Contact default global object to a page layout. Users who visit records based on that page layout don't see the New Contact action because the action is assigned to the Master record type by default. Editing the New Contact default global action and assigning it to a record type other than Master makes it available for all users who have access to its assigned record type.

Actions Best Practices

Use these tips as you set up actions.

REQUIRED EDITIONS

Actions available in: both Salesforce Classic and Lightning Experience

Publisher available in: Salesforce Classic

Available in: **Essentials, Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

Tips for Creating Actions

- Action labels longer than approximately 12–14 characters are shortened when they're displayed in the Chatter publisher. Keep names short and descriptive.
- Give your actions task-oriented names that tell your users what they do. Use terms such as New, Create, Share, Update, or Import.
- Use the **Description** field to create notes for yourself about each action. Notes are especially useful if you're creating several similar actions for different record types, for example. The description appears in the list of buttons, links, and actions for object-specific actions, or in the list of global actions, as well as on the detail page for the action. Your notes aren't visible to users.
- When creating custom actions that are going to be displayed in the Chatter publisher, limit their height to 400 pixels so that they're displayed correctly.

Tips for Laying Out Actions

- When customizing action layouts, consider what your users do with them. Minimalism is key. Include only the fields that are necessary for them and for whomever handles the cases, calls, or records that result from those actions.
- To create a single-column layout, such as for display on mobile devices, add fields only in the left column.

- Use predefined field values to set standard values for common fields. For example, when you configure an action that lets users create opportunities, set Prospecting as the predefined value for the Stage field. All new opportunities users create through that action are automatically assigned to the prospecting stage. You can remove the Stage field from the action's layout, because the field is going to be assigned a value automatically. If you set predefined values for fields on object records created through an action, you don't need to add those fields to the action layout.
- To see how an action layout appears to different user profiles, use the **Preview As** button on the Action Layout Editor.

Tips for Adding Actions to Publishers

- Because adding too many actions can cause the page to load slowly, we recommend including no more than nine actions total in each publisher, including any standard actions.
- In Salesforce Classic, if you include five or more actions in the actions section of a page layout, three are shown and the rest are added to the publisher's **More** menu. If you include four or fewer actions, they're all shown.
- In the Salesforce mobile app, the first three actions show up in the action bar, and the full set of actions are accessed by tapping .

See Also

- [Quick Action Considerations](#)
- [Troubleshooting Actions](#)
- [Set Up Actions with Chatter Enabled](#)
- [Create Dynamic Actions in Lightning App Builder](#)

Quick Action Considerations

Keep these considerations in mind when working with quick actions.

REQUIRED EDITIONS

 **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Actions available in: both Salesforce Classic and Lightning Experience

Available in: **Essentials, Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

- If you're using record types in your org, sometimes quick actions aren't visible to your users. For more information, see [Quick Actions and Record Types](#).
- The Printable View action is supported on Lightning Experience for desktop only. It isn't supported on mobile devices.
- To see an object-specific action on a page layout, a user must have both Read and Edit permissions on the action's relationship field. The relationship field is the field that's automatically populated on the

target object when a user creates a record using an action. For example, for an action on a case that lets users create child cases, the default relationship field is **Parent Case**. To be sure that users can see the Create Child Case action, check that they have both Read and Edit permissions on the **Parent Case** field.

Field-level security sets the Account parent field on the Case object to read-only by default, which means it's not visible to non-system administrator users. Therefore, actions on accounts that have Case as the target object are also not visible to users without the System Administrator profile. To let users see actions on accounts that have Case as the target object, remove the Read Only setting.

- In quick action layouts:
 - A blank space inserted on the same line as another field in the quick action layout causes that field to stretch to the full page width. The field is left justified on the page, regardless of its position in the quick action layout editor. If you add a field only to the right column in a two-column layout, the field label stretches to the full width
 - Two blank spaces inserted on the same row in the quick action layout editor don't create a blank line on the page. There's extra spacing between the rows as a result of the blank spaces.
 - If there's an uneven number of fields on the quick action layout, the final field stretches to the full page width and is left justified. The behavior is the same if a blank space is inserted on the same row as the final field.
- Custom actions can be defined for person accounts but with these exceptions.
 - Person account-specific fields, including **Email** and **Mobile**, aren't available in action layouts when using object-specific custom actions to update accounts.
 - To set up object-specific custom quick actions that create person account records, define a custom lookup field for **Account** on the Account or Contact object. Global custom quick actions can be used without defining this field.
 - Actions that create business account records from a person account detail page must be global, not object-specific.
- If you delete an action, the action is removed from all layouts that it's assigned to.
- Automatic triggers that change record ownership can affect what object details are visible to users. In some cases, a user can enter information that is then immediately hidden from them if an automatic trigger changes the object's owner. For example, if you use a global create quick action to create an object and an automatic trigger changes the object ownership, some information about the object isn't visible to you.
- Chatter groups without customers display the global publisher layout by default, unless you override it with a customized group publisher layout. In Chatter groups that allow customers, the publisher displays standard actions only, such as Post, File, Link, and Poll.
- When you create a custom quick action, use a unique label and API name. If the custom quick action has the same API name as a legacy standard action, Metadata API and change set deployment errors can occur. The incorrect action can also appear on page layouts.

Actions and Master-Detail Object Relationships

- Actions to create records for an object that is the detail object in a master-detail relationship must be object-specific, not global.
- If you create an action on a detail object in a master-detail relationship, the action's target object can't be a different detail object of the same master.

For example, master object A has two detail objects, B and C. You can create actions on object A with B or C as the target object. However, when creating an action on object B, the target object can't be object C because B and C have the same master. To create an action on B with C as the target, you can do one of the following:

- Change the relationship between B and C to master-detail so that A is the master of B, and B is the master of C.
- Change the relationship between C and A to a lookup relationship. A is still the master of B with this option, so you can't create a quick action on B with a target object of A.

Troubleshooting Actions

Use these tips to help you solve problems that arise with actions.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

I don't see feeds on record detail pages for a certain object.

Feeds appear only for objects for which you've enabled feed tracking.

I see the feed on a record detail page, but I don't see a publisher.

If there are no actions in the Quick Actions in the Salesforce Classic Publisher section on a page layout, the publisher in Salesforce Classic doesn't appear. Add at least one action to the page layout for the publisher to appear.

I can create actions, but I can't add them to publishers.

[Enable actions in the publisher](#) to add nonstandard actions to publishers in Salesforce Classic.

I'm using Internet Explorer 10 and all the actions I've created appear in the publisher with the same icon, even though the actions are for different types of objects.

Internet Explorer version 10 doesn't support the techniques Salesforce uses to show icons that correspond to the type of object an action is associated with. Consider using Chrome, Firefox, Safari, or an earlier version of Internet Explorer.

I've added an action to a page layout, but a user assigned to the profile that uses that page layout can't see the action.

Be sure that the user has both Read and Edit permissions on the action's relationship field. The relationship field is the field that's automatically populated on the target object when a user creates a record using an action. For example, for an action on a case that lets users create child cases, the default relationship field is Parent Case. To be sure that users can see the Create Child Case action, check that they have both Read and Edit permissions on the Parent Case field.

I don't see a relationship field on my global create actions.

Relationship fields apply only to object-specific create actions, not to global actions.

I don't see some of the actions in my Chatter groups.

Which actions you see depends on your role in the group, the type of group, and how your administrator has configured the publisher layout for groups. Chatter groups without customers display the global publisher layout by default, unless you override it with a customized group publisher layout. In Chatter groups that allow customers, the publisher displays standard actions only, such as Post, File, Link, and Poll.

I see an error when I select a custom create account action from a person account.

Although your administrator can add the custom create account action to the page layout, this action isn't supported for person accounts.

See Also

[Actions Best Practices](#)

[Set Up Actions with Chatter Enabled](#)

Action Limits and Limitations

Actions can work differently than you expect in certain situations and for different objects. Keep these limits and limitations in mind when working with actions.

REQUIRED EDITIONS

Available in: **Essentials, Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

- Quick actions aren't available in Aura Experience Builder sites.
- You can display up to 10 actions in Lightning view.

- Custom images used for action icons must be less than 1 MB in size.
- Custom actions can be defined for person accounts but with these exceptions.
 - Person account-specific fields, including **Email** and **Mobile**, aren't available in action layouts when using object-specific custom actions to update accounts.
 - To set up object-specific custom quick actions that create person account records, define a custom lookup field for **Account** on the Account or Contact object. Global custom quick actions can be used without defining this field.
 - Actions that create business account records from a person account detail page must be global, not object-specific.
- Actions associated with objects that aren't supported in Lightning Experience don't appear in the Global Actions menu. Also, the Global Actions menu doesn't support standard Chatter actions.
- Mobile smart actions don't appear in the full Salesforce site, regardless of which page layouts you add them to. They appear only in the Salesforce mobile app.
- The Chatter page in Lightning Experience supports only the standard Chatter actions Post, Poll, and Question, and if you have Groups, the Announcement action.
- Quick actions don't inherit the app's branding on mobile.
- The palette in the action layout editor doesn't support:
 - Record type fields
 - Read-only field types such as roll-up summary, formula, and auto-number fields
 - Read-only system fields such as **Created By** or **Last Modified By**
 - When you create a custom action and the action type is Update a Record, you can't add the Owner field to the action layout for most objects. The exception is Case records. You can add the Case Owner field to your action layout when you create a custom action to update a Case record.
- Actions on the user profile page come from the Quick Actions in the Salesforce Classic Publisher section of the global publisher layout. Only standard Chatter actions appear on the user profile page, regardless of which actions are assigned to the User Page Layout or the global publisher layout.
- Actions on reports come from the Quick Actions in the Salesforce Classic Publisher section of the global publisher layout. Only standard Chatter actions appear on reports, regardless of which other actions are assigned to the global publisher layout.
- Chatter groups with customers don't support global create, log a call, or custom actions and display only standard Chatter actions, such as Post, File, Link, and Poll.
- External objects support quick actions, except when the actions involve features or functionality that are incompatible with external objects. For example:
 - Formulas can't reference fields on external objects, so you can't reference an external object field to set a predefined field value for a quick action.
 - Log a Call actions create tasks, which aren't available for external objects.
- When you create an object-specific action, you can choose as a target object an event, a task, or any object that has a parent-child or lookup relationship to the host object. You can't choose Quote as a target object from Opportunity. But to create quotes from an opportunity you can go to the opportunity's Quotes related list and click **New**.
- The Account Contact Relationship object supports custom actions, but with limitations.
 - Custom actions must be specific to the Account Contact Relationship object. Global actions aren't supported.
 - You can create actions that update records or invoke Lightning components or Visualforce pages. But you can't create actions that create records.

- You can't create actions that send emails or log calls because you can't associate activities to the Account Contact Relationship object.
- You can't override the default actions on the Account Contact Relationship object.
- Only custom actions that are created on the Contact object can use the Account Contact Relationship object as a target object.
- Chatter standard actions aren't supported because the Account Contact Relationship object doesn't have a Chatter feed.

Mass Quick Actions

With mass quick actions, users can create or update up to 100 records from a list view or related list. Users can select one record in the list to update only that record, or multiple records to perform bulk updates. To help users understand the changes they're making, the action window includes a Fields to update section when users update more than one record.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Quick actions available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer** Editions

Custom canvas actions available in: **Professional** (with Canvas enabled), **Enterprise, Performance, Unlimited, and Developer** Editions

Only these quick action types are supported for mass quick actions.

- Create a Record
- Update a Record

[Set Up a Mass Quick Action for List Views](#)

With mass quick actions on a list view, users can create or update records in bulk rather than one at a time. You can configure a mass quick action for cases, leads, accounts, campaigns, contacts, opportunities, work orders, and custom objects that support quick actions and have a list view button layout in Lightning Experience.

[Set Up a Mass Quick Action for Related Lists](#)

With mass quick actions on a related list, users can create or update related records in bulk without leaving the page. You can configure a mass quick action on related lists for any related object that supports quick actions.

[Mass Quick Action Considerations](#)

Review these guidelines and considerations before setting up and using mass quick actions in list views.

Set Up a Mass Quick Action for List Views

With mass quick actions on a list view, users can create or update records in bulk rather than one at a time. You can configure a mass quick action for cases, leads, accounts, campaigns, contacts, opportunities, work orders, and custom objects that support quick actions and have a list view button layout in Lightning Experience.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Essentials, Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To set up mass quick actions for Lightning Experience: Customize Application

Before setting up mass quick actions on a list view:

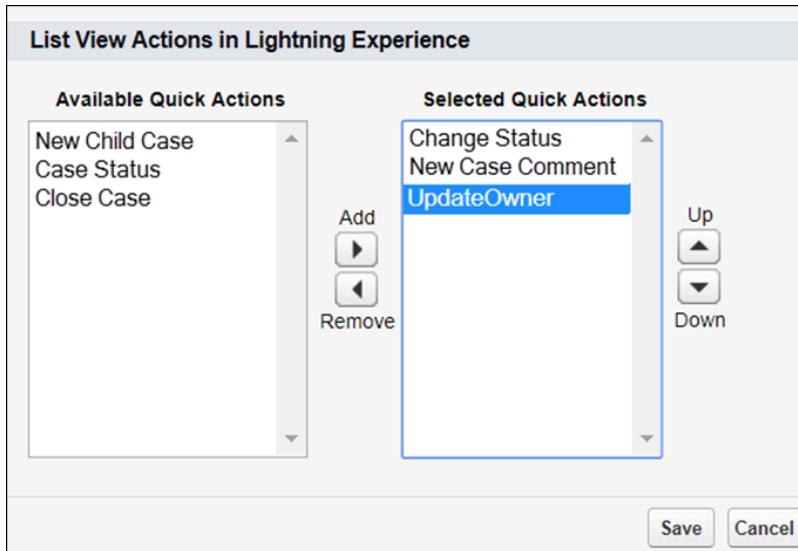
- Review [Mass Quick Action Considerations](#).
- Set up Create a Record or Update a Record quick actions for your objects. These actions are the ones that you want users to perform on multiple records in a list view.

To add mass quick actions on a list view, customize an object's list view button layout.

1. From the object management settings for the object that you want to allow mass quick actions on, go to List View Button Layout.
2. Edit the List View layout.
3. In the List View Actions in Lightning Experience section, add the actions that you want your users to be able to perform on list views for multiple records.
4. Save your changes.

The list view for the object that you updated now includes buttons for the actions that you added.

 **Example** Here's an example of the list view actions configured for the case object.



See Also

[Mass Quick Action Considerations](#)

Set Up a Mass Quick Action for Related Lists

With mass quick actions on a related list, users can create or update related records in bulk without leaving the page. You can configure a mass quick action on related lists for any related object that supports quick actions.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Essentials, Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To set up mass quick actions for Lightning Experience:

Customize Application

Before setting up mass quick actions on a related list:

- Review [Mass Quick Action Considerations](#).
- Set up Create a Record or Update a Record quick actions for the object contained in the related list, not the parent object. For example, to add a mass quick action to the Opportunities related list on the Account record detail page, create a quick action on the Opportunity object.

There are two ways to add mass quick actions on a related list. To add actions directly from the Lightning App Builder, select the Dynamic Related List - Single component in the record page. In the properties pane, click **Add Action**, and then select your quick action.

Or, you can set up mass quick actions on a related list from the page layout editor.

1. From the object management settings for the parent object, go to the Page Layouts section and choose the page layout to edit.
2. Edit the related list's properties and expand the Buttons section.
3. Under Quick Actions in Lightning Experience (Beta), select the quick actions to show on the related list.
4. Save your changes to the page layout.

If the actions that you added don't appear on your related list, update the list's properties. In the Lightning App Builder, on the parent record page, select the Related Lists component or the Related List - Single component. Then, in the properties pane:

- Set the **Show list view action bar** checkbox.
- Update the Related List Type to **Enhanced List**.

See Also

[Customize Related Lists](#)

[Standard Lightning Page Components](#)

Mass Quick Action Considerations

Review these guidelines and considerations before setting up and using mass quick actions in list views.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Essentials, Personal, Group, Enterprise, Performance, Unlimited, Developer, and Professional Editions**

Setting Up a Mass Quick Action

- Mass quick actions are available only in Lightning Experience apps, including apps with standard and console navigation.
- You can configure mass quick actions for related lists. Custom quick actions for related lists aren't supported.
- You can't perform mass quick actions in Experience Cloud sites, or on notes or users.
- You can't use Tooling API, AppExchange, and Changesets to add mass quick actions to an object's list view button layout.
- Set up **Create a Record or Update a Record** quick actions for the object referenced in a related list, not the parent object. For example, to add a mass quick action to the Opportunities related list on the Account record detail page, create a quick action on the Opportunity object.
- If you want to set up predefined field values for a quick action, we recommend not including those fields on the quick action's layout.

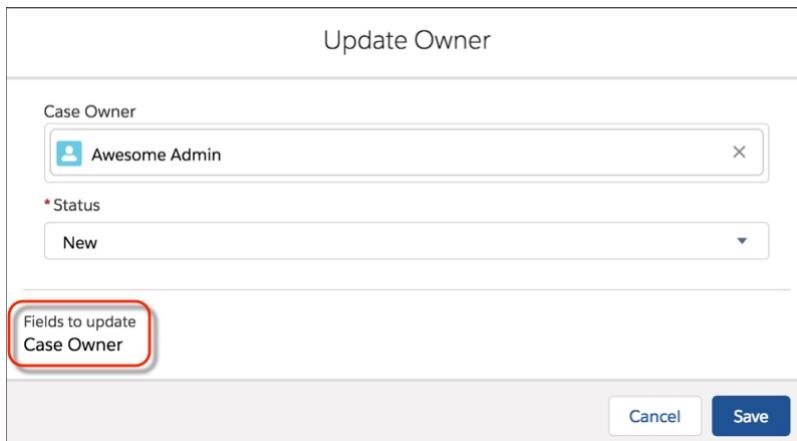
If a predefined field value is included in the quick action's layout, it's only updated if the user manually

selects it. If a predefined field value isn't included in the quick action's layout, the predefined value is updated, but the user isn't notified of the change. The Fields to update section includes only predefined field values that are specified in the layout.

Using a Mass Quick Action

- When using a mass quick action, only the fields you manually modify are changed. Some fields show a default value, but changes aren't made unless you manually select them. The Fields to update section shows the changes you're making.
- You can't undo changes performed by a mass quick action—so be careful when making your changes.
- In list views, you can use mass actions in both table view and split view. Mass actions in split view use the same logic as mass actions in table view.
- Mass quick actions on the related list don't work on the parent record. They work directly on the selected records within the related list. A user must select one or more target records to be used on the quick action.
- If you use a mass quick action to update only one record, the Fields to update section isn't displayed.
- List views in split view and related lists have checkboxes to select records only if a mass action is available.
- Mass quick actions aren't available on the Salesforce mobile app. The action button might be visible on the mobile app, but it doesn't make any changes.

 **Example** This mass quick action updates the owner on cases. The Case Owner field displays Awesome Admin because the user modified this field. The Status field displays New because that was the value of all the selected records. The changes to make are listed in the Fields to update section.



The screenshot shows a modal dialog titled "Update Owner". Inside, there are two fields: "Case Owner" (set to "Awesome Admin") and "* Status" (set to "New"). Below these, a section labeled "Fields to update" contains the "Case Owner" field. At the bottom of the dialog are "Cancel" and "Save" buttons.

If the selected records have different statuses, such as one is set to New and one is set to Escalated, the Status field shows None.

See Also

[Set Up a Mass Quick Action for List Views](#)

Actions in Lightning Experience

In Lightning Experience, actions appear in the Global Actions menu in the header, on related lists, and

on list view items. Actions also appear on a record page, in one of several places depending on the action's type.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

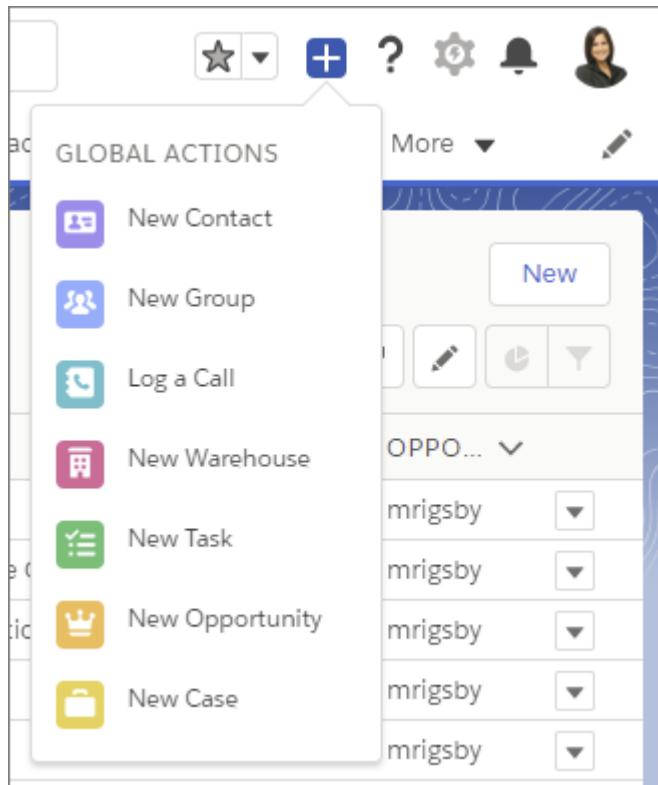
Quick actions available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

Custom canvas actions available in: **Professional (with Canvas enabled), Enterprise, Performance, Unlimited, and Developer Editions**

 **Note** Quick actions aren't the only action type covered here. Standard and custom buttons are also considered actions.

Actions in the Global Actions Menu

The Global Actions menu displays a subset of global actions from the Salesforce Mobile and Lightning Experience Actions section of the global publisher layout.



The items in the menu appear in the order that they're listed in the Salesforce Mobile and Lightning Experience Actions section of the global publisher layout.

Actions associated with objects that aren't supported in Lightning Experience don't appear in the Global

Actions menu. Also, the Global Actions menu doesn't support standard Chatter actions.

Actions on List Views and List View Items

Custom list buttons, list view actions, and certain standard buttons are supported on all list views, except Recently Viewed. To have a custom list button appear on a list view, add the button to the object's List View Button Layout.

In the Kanban view, only the standard New action is supported.

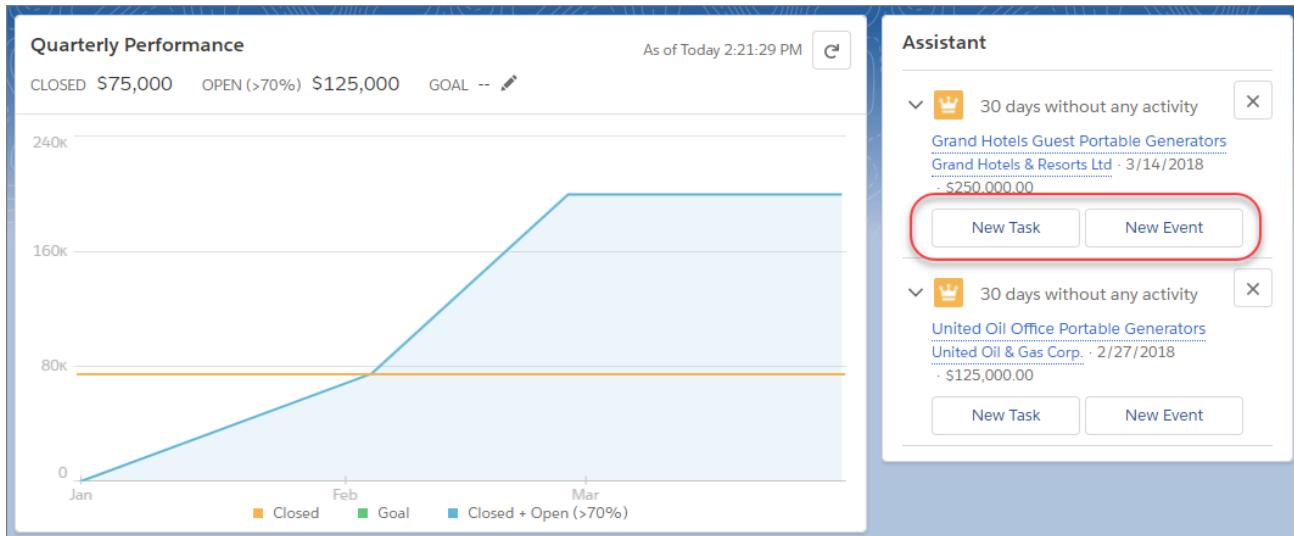
List view items support only specific standard actions, like Edit, Delete, or Change Owner.

1	ACCOUNT NAME	ACC...	BILLING ...	PHONE	TYPE	ACCO...	AUser
1	Barbary Coast Wireless	CA	415-555-3284	Customer - Direct			
2	Burlington Textiles Corp of America	NC	(336) 222-7000	Customer - Direct			
3	Dickenson plc	KS	(785) 241-6200	Customer - Channel			
4	Edge Communications	TX	(512) 757-6000	Customer - Direct			
5	Express Logistics and Transport	OR	(503) 421-7800	Customer - Channel			
6	GenePoint	CA	(650) 867-3450	Customer - Channel			
7	Grand Hotels & Resorts Ltd	IL	(312) 596-1000	Customer - Direct	AUser		

For Tasks, table format list views and the Kanban view support only standard buttons. Tasks list view items in table view, and the task item detail pane in split view contain the complete list of available actions for tasks.

Actions on the Home Page

On the Home page, you can find actions on recommendations in the Assistant. For example, imagine that a sales rep receives an update that an opportunity doesn't have any open activity. The rep can create a task or event directly from the recommendation.



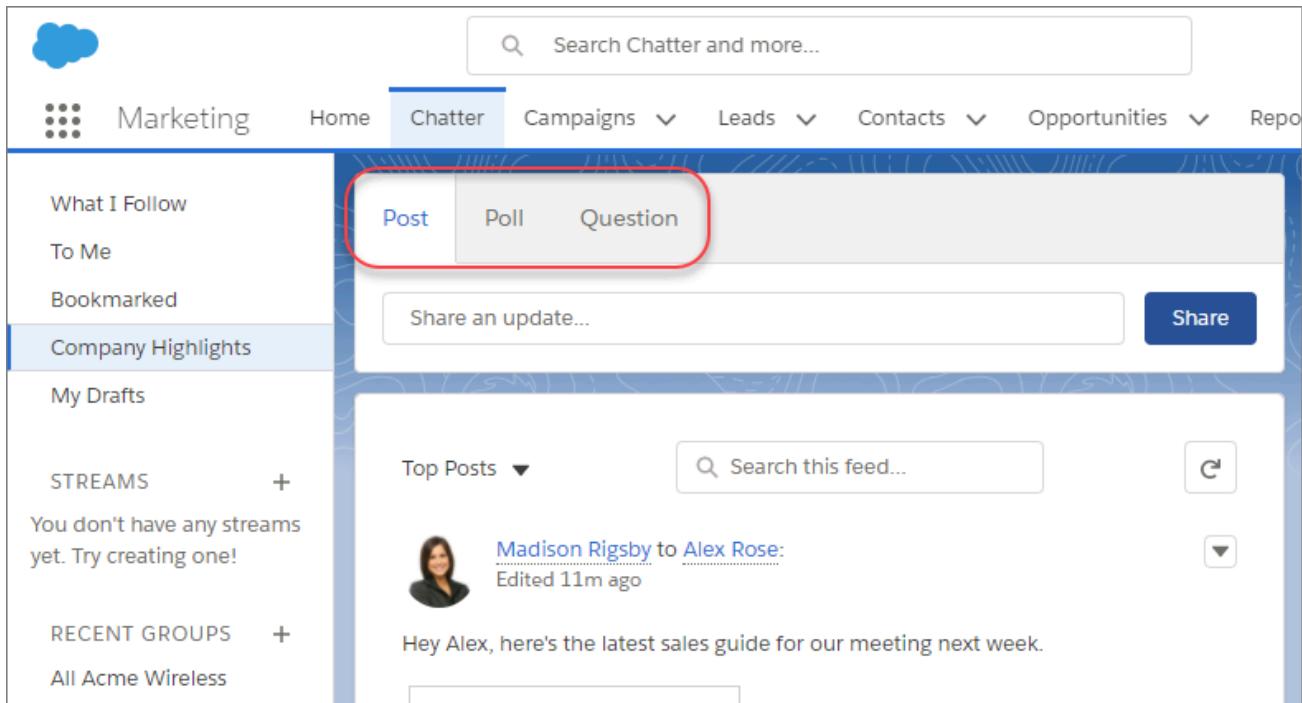
The actions that appear depend on the type of recommendation. To appear in the Assistant, actions must be added to the Salesforce Mobile and Lightning Experience Actions section of the global publisher layout. Supported actions include:

- New Task
- New Event
- Edit
- Email

After you complete an action, the related recommendation disappears from the Assistant.

Actions on the Chatter Page

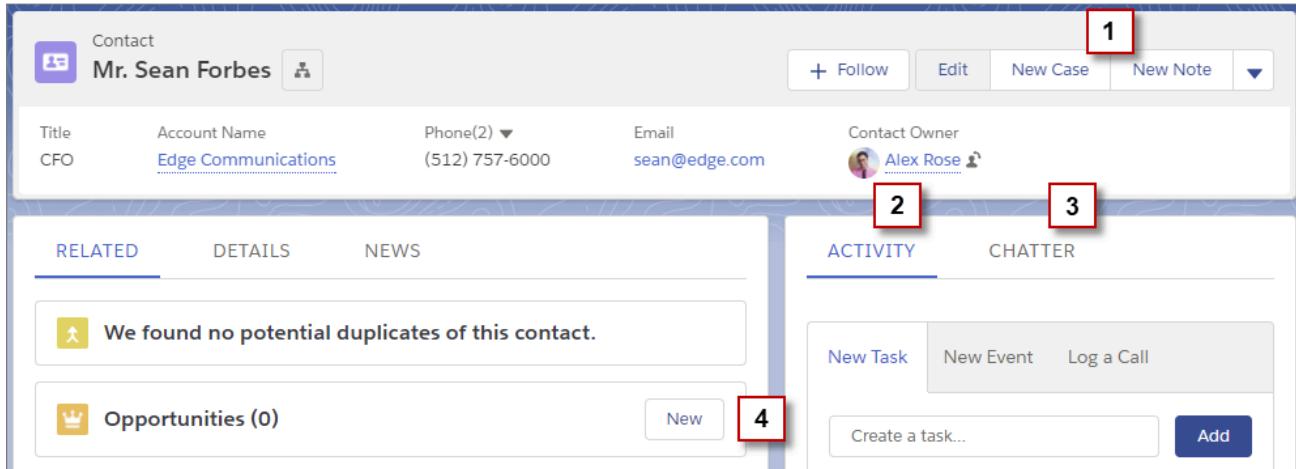
The Chatter page, like the Chatter tab on record pages, contains only standard Chatter actions. By default, only the Post, Poll, and Question actions are supported, and if you have Groups, the Announcement action. You can add, remove, or reorder the actions on the Chatter page from the Salesforce Mobile and Lightning Experience Actions section of the global publisher layout.



Actions on Record Pages

Here's a sample contact page in Lightning Experience.

- Note** The opportunity and leads workspaces have different structures, but actions appear in the same way on those pages.



The page-level action menu in the record's highlights panel (1) contains:

- Productivity actions
- Global and object-specific quick actions, except for those actions related to creating tasks, creating events, and logging calls
- Standard buttons
- Custom object-specific Lightning component and Lightning web component quick actions

- Custom flow actions
- Custom Visualforce quick actions
- Custom Visualforce buttons
- Canvas actions

The actions that appear in the page-level action menu are listed in the order that they appear in the Salesforce Mobile and Lightning Experience Actions section of the page layout.

 **Note** You can enable dynamic actions in the highlights panel on an object's record page using the Lightning App Builder. Dynamic actions are supported for custom objects on desktop and mobile and for standard objects on desktop. Add, remove, and reorder actions directly in the Lightning App Builder and control action visibility based on filters that you apply. When you enable dynamic actions in the Lightning App Builder, highlights panel actions for the record page no longer come from the object's page layout. Custom Lightning page templates with dynamic actions aren't supported in Salesforce Mobile.

The Activity tab (2) contains Create a Record quick actions that point to the Event and Task objects. It also contains Log A Call and Send Email actions.

The Chatter tab (3) contains standard Chatter actions. By default, only the Post, Poll, and Question actions are supported, and if you have Groups, the Announcement action. Some objects support other standard Chatter actions predefined by Salesforce.

 **Note** Actions on user profiles, cases, and work orders can appear in a different way than on other records.

- Actions on the user profile page come from the Quick Actions in the Salesforce Classic Publisher section of the global publisher layout. Only standard Chatter actions appear on the user profile page, regardless of which actions are assigned to the User Page Layout or the global publisher layout.
- When feed tracking is enabled for cases or work orders, the page-level action menu on those records contains only custom buttons and supported standard buttons. Quick actions appear on the Chatter tab. On Experience Builder sites, quick actions for work orders appear on the page-level action menu.

Actions on Related Lists

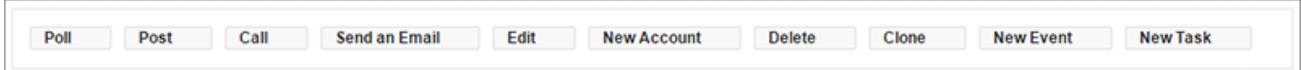
Related lists in Lightning Experience (4) show custom list buttons, supported quick actions, and supported standard buttons that are assigned to the related list. Not all related list standard buttons are supported in Lightning Experience.

Actions on Reports

Actions on reports come from the Quick Actions in the Salesforce Classic Publisher section of the global

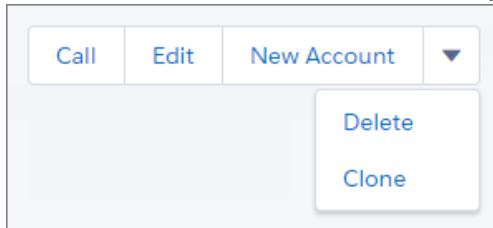
publisher layout. Only standard Chatter actions appear on reports, regardless of which other actions are assigned to the global publisher layout.

-  **Example** Let's say you have these actions on your Contact page layout in the Salesforce Mobile and Lightning Experience Actions section.

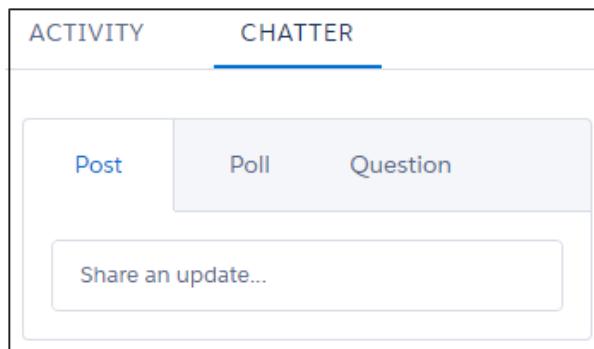


You have quick actions (New Account, New Event, New Task), a productivity action (Call), standard buttons (Edit, Delete, Clone, Send an Email), and Chatter actions (Poll, Post). Here's how those actions appear on a contact record page in Lightning Experience.

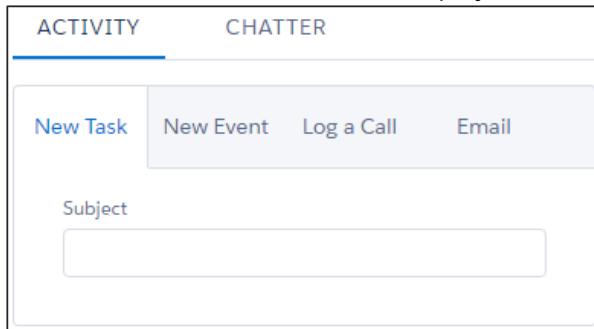
- The actions in the page-level action menu are a combination of the quick actions, productivity actions, and standard buttons. These actions appear in the order that they're listed on the page layout. Although they're quick actions, New Event and New Task don't show up here.



- The Chatter actions from the front of the action list are on the Chatter tab.



- The Activities-related actions—Email, New Event, New Task—display on the Activity tab.



How Actions Are Ordered in Lightning Experience

In Lightning Experience, the actions on record pages are derived from the list of actions in the Salesforce Mobile and Lightning Experience Actions section of the page layout for that object. The same section on global publisher layouts determines the global actions that appear in the Global

Actions menu.

See Also

[Quick Actions](#)

[Quick Action Considerations](#)

[Set Up Cases for Lightning Experience](#)

[Create Dynamic Actions in Lightning App Builder](#)

How Actions Are Ordered in Lightning Experience

In Lightning Experience, the actions on record pages are derived from the list of actions in the Salesforce Mobile and Lightning Experience Actions section of the page layout for that object. The same section on global publisher layouts determines the global actions that appear in the Global Actions menu.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Quick actions available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer** Editions

Custom canvas actions available in: **Professional** (with Canvas enabled), **Enterprise, Performance, Unlimited, and Developer** Editions

If you haven't customized the Salesforce Mobile and Lightning Experience Actions section of an object's page layout, the quick actions that appear on the object's record pages are derived from:

- The actions on the global publisher layout
- Standard and custom buttons in the buttons section of the object page layout

However, the order of the actions from the global page layout aren't respected on those record pages.

When you click to override the predefined actions in the Salesforce Mobile and Lightning Experience Actions section, the custom buttons in the buttons section of the page layout aren't automatically included in the action list. You must add the custom buttons as actions from the Mobile & Lightning Actions category in the palette.

After you customize the Salesforce Mobile and Lightning Experience Actions section of an object's page layout, the actions in each section of the record page respect the ordering of its types of actions on the page layout. For example, actions in the page-level actions menu appear in the order that you configure them in the Salesforce Mobile and Lightning Experience Actions section on the page layout. And actions in the Chatter and Activity tabs reflect the order of the actions supported for those tabs on the page layout.

The Global Actions menu (+) in the Lightning Experience header displays all global quick actions from the Salesforce Mobile and Lightning Experience Actions section of the global publisher layout, except the

standard Chatter actions Post, File, Poll, Link, Question, and Thanks.

See Also

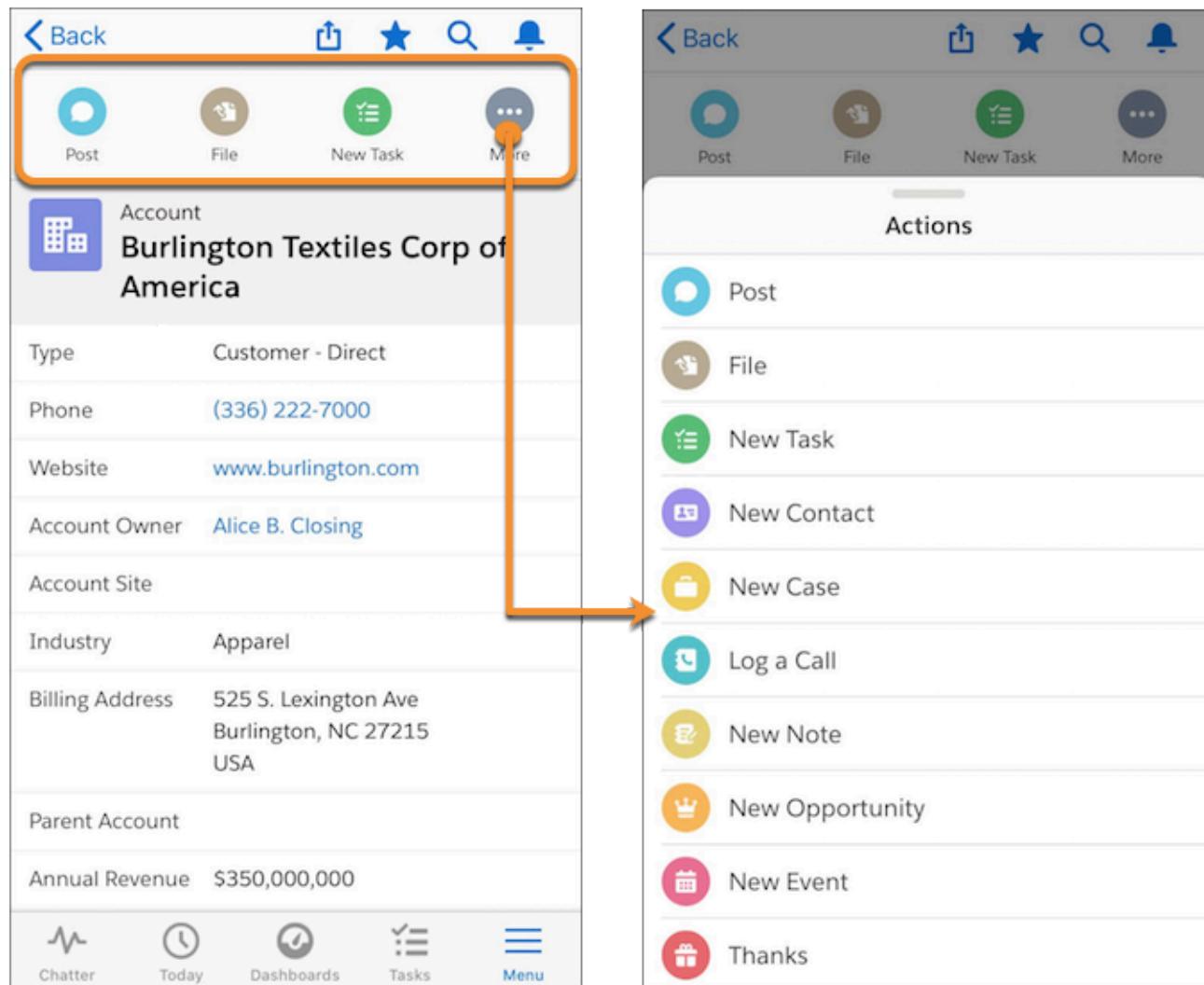
- [Actions in Lightning Experience](#)
- [Send Email Action Considerations for Cases](#)
- [Quick Actions](#)
- [Create Dynamic Actions in Lightning App Builder](#)

Salesforce Mobile App Action Bar

The productivity actions, standard and custom record buttons, and quick actions appear in the action bar and the action menu.

The action bar and action menu show all the available actions for a given page.

Salesforce Mobile App Actions in the Action Bar and Action Menu



The action bar appears in most places in the mobile app, including the feed, groups, user profiles,

dashboards and reports, standard and custom object record views, related lists, and search results. The actions that are available depend on where a user is in the app and on how you've configured page layouts and publisher layouts for your organization.

Users see some or all of these kinds of actions in the action bar, including the action menu.

- Productivity actions—The Send Email, Call, Map, View Website, and Read News actions are available on accounts, contacts, leads, and person accounts. The Quick Message, Join Conference Call, and Map actions are available on mobile calendar events in Salesforce Today.



Tip In most cases, a productivity action displays only if a record includes the information that the action is keyed to. For example, the Send Email action depends on the record including an email address. The View Website action requires the record to include a website URL.

- Custom and standard buttons—Buttons (such as Edit, Delete, or Clone) that are included in the Buttons section on an object's page layout are available in the mobile app as actions in the action bar on record pages. If you haven't customized the action order, the button order in the button section of the page layout is used. However, the Edit button is in a fixed position.



Note Custom links, custom buttons that are added to list views, and custom buttons that define the content source as *OnClick JavaScript* aren't supported and don't appear in the Salesforce mobile app.

- Quick actions—if you add, remove, or reorder actions in the action bar in the global publisher layout or an object's page layout, the changes are reflected in the Salesforce mobile app.
- Standard Chatter actions—Actions unique to Chatter, such as Post, File, Link, or Poll.

What are the exceptions?

- On accounts, the Call action appears even when there isn't a phone number on the record.
- On record feeds, the **Follow/Unfollow** button remains in the highlights area on the page.
- On public and private group feeds, the **Join Group** and **Ask to Join** buttons remain in the highlights area, but the **Leave Group** button is in the action bar.

What does this mean for custom branding?

There can be only one version of a custom icon. Custom action icons that you created before Winter '15 are still supported, but they're truncated in the action bar. To optimize your custom action icons for display in the action bar, use these guidelines.

- The icon should be 72 x 72 pixels. Use the full pixel area for the image—don't leave spacing around the image like before.
- Make the image a PNG with a transparent background, with a file size that is less than 10k.
- Have a resolution of 72 dpi.
- Make the icon graphic white or lighter than the background color.
- Avoid heavy inner or outer shadows.
- Use simple and flat styling resembling the Salesforce mobile app icon family.

How Actions Are Ordered in the Salesforce Mobile App Action Bar

The Salesforce Mobile and Lightning Experience Actions section of a page layout and global publisher layout drives which actions appear in the Salesforce mobile app action bar. You can also customize the order of quick actions, productivity actions, and standard and custom buttons that are available as actions.

List Item Actions in the Salesforce Mobile App

List item actions give you access to your actions in list views, task lists, and related record lists. You can use list item actions to update records directly from lists.

How Predefined Actions Are Ordered in the Salesforce Mobile App Action Bar and List Item Actions

Your org's page layouts and publisher layouts control the order in which actions appear in the Salesforce mobile app action bar and list item actions. If you don't customize the actions in the action bar on a page layout or global publisher layout, Salesforce predefines the location of key actions.

Considerations for Actions in the Salesforce Mobile App

Keep these considerations in mind when you configure actions for use in the Salesforce mobile app.

How Actions Are Ordered in the Salesforce Mobile App Action Bar

The Salesforce Mobile and Lightning Experience Actions section of a page layout and global publisher layout drives which actions appear in the Salesforce mobile app action bar. You can also customize the order of quick actions, productivity actions, and standard and custom buttons that are available as actions.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

If you customize the Salesforce Mobile and Lightning Experience Actions section of a layout, the mobile app reflects your customizations.

If you customize the Quick Actions in the Salesforce Classic Publisher section but not the Salesforce mobile app section, the actions in the mobile app action bar are a combination of the ones in the Quick Actions in the Salesforce Classic Publisher section plus any standard or custom buttons present on the page layout.

When you click to override the predefined actions in the Salesforce Mobile and Lightning Experience Actions section, the custom buttons in the buttons section of the page layout aren't automatically included in the action list. Add the custom buttons as actions from the Mobile & Lightning Actions category in the palette.

If neither section is customized, the action bar inherits a default set of actions predefined by Salesforce. The sets of actions differ between objects based on the most common or typical activities required for

each object.

-  **Note** You can enable dynamic actions for custom object record pages for the Salesforce mobile app. When you enable dynamic actions, you assign actions in the Lightning App Builder instead of the page layout and apply filters to control when and where actions appear for users. You can assign the same dynamic actions for desktop and mobile, or assign a different set of dynamic actions for mobile.

See Also

- [How Predefined Actions Are Ordered in the Salesforce Mobile App Action Bar and List Item Actions](#)
- [Salesforce Mobile App Action Bar](#)
- [Considerations for Actions in the Salesforce Mobile App](#)

List Item Actions in the Salesforce Mobile App

List item actions give you access to your actions in list views, task lists, and related record lists. You can use list item actions to update records directly from lists.

To access list item actions, navigate to a list view or task list or open a related list from an object's related information page. Then swipe left on the desired record.

Hide list item actions by swiping the list item back to the right or by tapping another item in the list.

List Views

List item actions in list views don't have the same actions that are available in the action bar when viewing an object's record. For example, when a user visits an opportunity record in the Salesforce mobile app, the actions in its action bar reflect the actions in the Salesforce Mobile and Lightning Experience Actions section of the Opportunity page layout.

However, when the user swipes left on an opportunity from a list view, as you can see below, the actions that display come from the list of predefined actions for opportunities. See [How Predefined Actions Are Ordered in the Salesforce Mobile App Action Bar and List Item Actions](#) for the breakdown of predefined actions for each object.

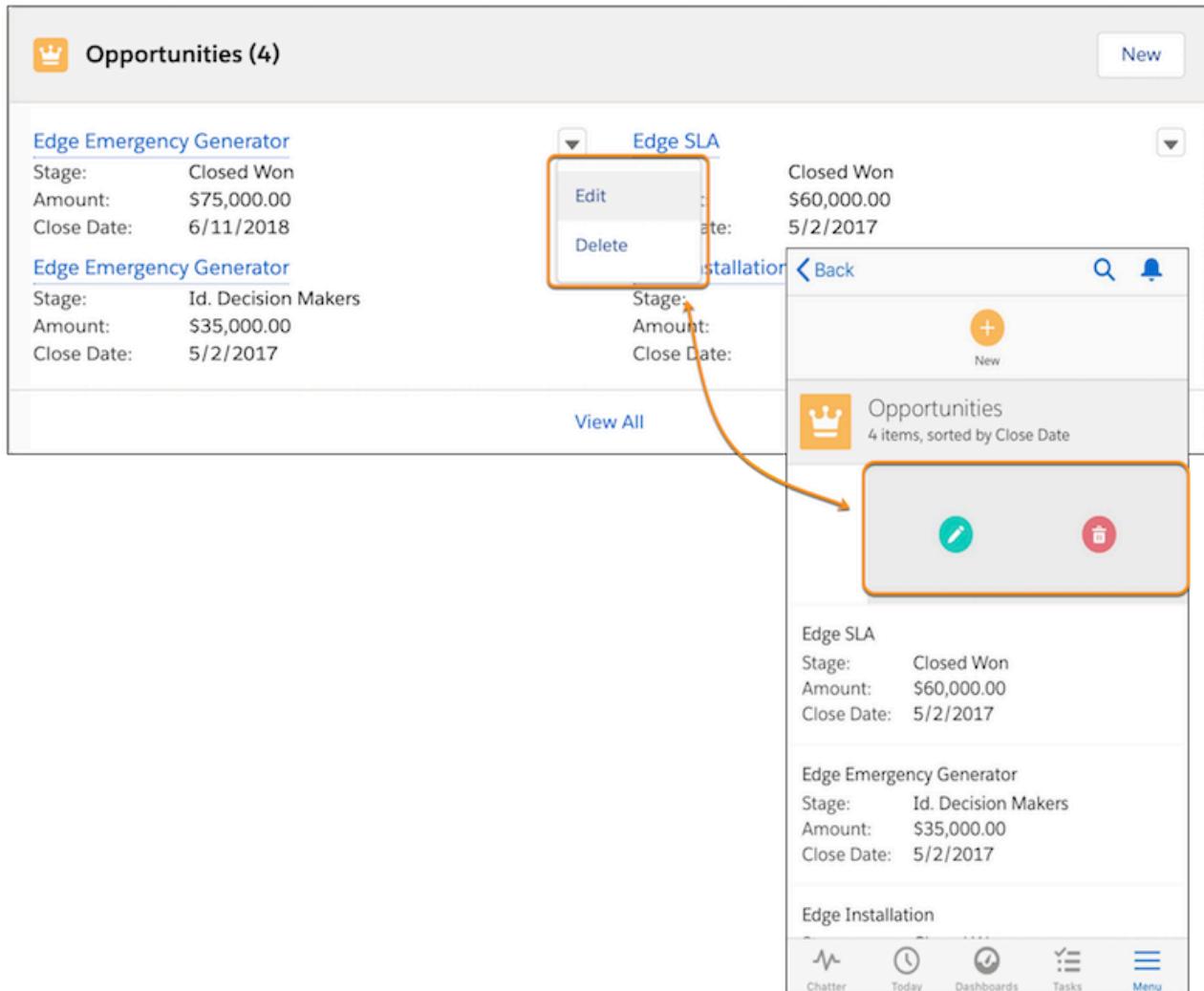
Here's the All Opportunities list view.	Swipe a list item to the left to reveal list item actions.	Tap  to show the action menu, with the full list of actions available for the list item.
		

Related Lists

List item actions on related lists in the app reflect the same actions as the related list in Lightning Experience. Usually, these are the standard actions Edit and Delete, but can vary by object. For example,

in Lightning Experience, the actions available in the dropdown menu on an Opportunity related list item are the same set of actions that Salesforce mobile app users see when swiping left on the same record in the related list.

- Note** Even if your org doesn't have Lightning Experience enabled, the actions on related lists that you see in the Salesforce mobile app still match the actions that would appear on the related list items in Lightning Experience if it was enabled.



- Note** Task and event list items are different than other objects when they display in related lists. In the mobile app, when you tap the Tasks item from the menu, you can swipe left on an item in the My Tasks list view and see the predefined actions for tasks. However, tasks and events in the Open Activities or Activity History related lists in the mobile app have no actions and aren't swipeable.

How Predefined Actions Are Ordered in the Salesforce Mobile App Action Bar and List Item Actions

Your org's page layouts and publisher layouts control the order in which actions appear in the Salesforce mobile app action bar and list item actions. If you don't customize the actions in the action bar on a page layout or global publisher layout, Salesforce predefines the location of key actions.

! **Important** Predefined actions apply to the Salesforce mobile app action bar only if you *haven't* customized the Salesforce Mobile and Lightning Experience Actions section of an object's page layout or a global publisher layout.

Predefined actions are derived from the Quick Actions in the Salesforce Classic Publisher section of the object page layout or global publisher layout.

Layout	Behavior
On object page layouts, when the Salesforce mobile app section isn't customized.	If you customized the Quick Actions in the Salesforce Classic Publisher section, the quick actions in the action bar reflect those customizations. If you didn't customize either section, the quick actions in the action bar come from the Quick Actions in the Salesforce Classic Publisher section of the global publisher layout.
On global publisher layouts, when the Salesforce mobile app section isn't customized.	If you customized the Quick Actions in the Salesforce Classic Publisher section, the quick actions in the action bar reflect those customizations. If you didn't customize either section, the quick actions in the action bar for global pages default to a Salesforce predefined set.

Here's the breakdown of which actions are contained in each group for each object or page. Keep in mind these considerations.

- The predefined actions in the action bar, list item actions, and associated action menus are divided into groups. The arrangement of these groups is fixed.
- Unless they're in an ordered list, the order of actions within the groups can vary based on the object and the actions present on the global publisher layout or on an object's page layout.
- Some actions are in fixed positions. Actions in a numbered list appear in this fixed position in the action bar, list item actions, and in the respective action menus.
 - For example, for the Account object, the standard Chatter Post action is in the fourth position. This

position is fixed. Regardless of where you put the Post action in the account page layout, Post always displays in the fourth position.

- However, deletion of actions is respected. So in our example, if you delete the Post action from the account page layout, the remaining actions move up and you see Edit in the fourth position.
- Some action groups aren't supported for every object or page.

 **Note** Actions on list view items reflect only the predefined set of actions for that object. For example, let's say you're viewing the All Accounts list in the Salesforce mobile app. If you swipe left on an account item in the list, you see a set of actions. Those actions come from the predefined list of actions for accounts in this chart. You always see Call, Edit, and Delete. The other actions on the list view item follow the order and rules defined for the action groups in the chart.

Account Object

Action Group	Predefined Actions
1	1. Call, 2. New Task, 3. New Event, 4. Post
2	Edit
3	The remaining quick actions from the Quick Actions in the Salesforce Classic Publisher section of the Account page layout. If that section isn't customized, the remaining quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of the <i>global publisher layout</i> .
4	Custom buttons that are supported in the Salesforce mobile app, in the order defined on the page layout.*
5	The remaining standard buttons that are supported in the Salesforce mobile app, in the order defined on the page layout.
6	Send Text (if the Phone field is populated), View Website (if the Website field is populated)

Case Object

Action Group	Predefined Actions
1	Actions from the Quick Actions in the Salesforce Classic Publisher section of the Case page layout. If that section isn't customized, quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of the <i>global publisher layout</i>
2	Edit
3	Action group 3 isn't supported for the Case object.
4	Custom buttons that are supported in the Salesforce mobile app, in the order defined on the page layout.*

Action Group	Predefined Actions
5	The remaining standard buttons that are supported in the Salesforce mobile app, in the order defined on the page layout.
6	Productivity actions aren't supported for this object.

Contact Object

Action Group	Predefined Actions
1	1. Call, 2. Send Email, 3. New Task, 4. New Event
2	Edit
3	Remaining quick actions from the Quick Actions in the Salesforce Classic Publisher section of the Contact page layout. If that section isn't customized, the remaining quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of the <i>global publisher layout</i> .
4	Custom buttons that are supported in the Salesforce mobile app, in the order defined on the page layout.*
5	Remaining standard buttons that are supported in the Salesforce mobile app, in the order defined on the page layout.
6	Send Text

Custom Objects

Action Group	Predefined Actions
1	The first four actions in the order defined on the page layout. If the Quick Actions in the Salesforce Classic Publisher section of the page layout isn't customized, then the first four actions in the order defined on the <i>global publisher layout</i> .
2	Edit
3	The remaining quick actions from the Quick Actions in the Salesforce Classic Publisher section of the page layout. If that section isn't customized, remaining quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of the <i>global publisher layout</i> .
4	Custom buttons that are supported in the Salesforce mobile app, in the order defined on the page layout.*
5	Remaining standard buttons that are supported in the Salesforce mobile app, in the order defined on the page layout.

Action Group	Predefined Actions
6	Action group 6 isn't supported for custom objects.

Event Object

Action Group	Predefined Actions
1	Quick actions in the order defined on the layout. Standard Chatter actions aren't supported.
2	Edit, Delete
3-6	Action groups 3-6 aren't supported for the Event object.

Feed Object

Action Group	Predefined Actions
1	Quick actions in the order defined on the global publisher layout
2-6	Action groups 2-6 aren't supported for the Feed object.

Group Object

Action Group	Predefined Actions
1	Actions from the Quick Actions in the Salesforce Classic Publisher section of the Group page layout. If that section isn't customized, quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of the <i>global publisher layout</i> .
2-4	Action groups 2-4 aren't supported for the Group object.
5	The remaining standard buttons that are supported in the Salesforce mobile app, in the order defined on the page layout.
6	Action group 6 isn't supported for the Group object.

Lead Object

Action Group	Predefined Actions
1	1. Log a Call, 2. New Task, 3. Convert (if enabled), 4. Post
2	Edit
3	The remaining quick actions from the Quick Actions in the Salesforce Classic Publisher section of the Lead page layout. If that section isn't customized, the remaining quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of

Action Group	Predefined Actions
	the <i>global publisher layout</i> .
4	Custom buttons that are supported in the Salesforce mobile app, in the order defined on the Lead page layout.*
5	The remaining standard buttons that are supported in the Salesforce mobile app, in the order defined on the Lead page layout.
6	In order: Call, Send Text, Send Email

“App Page” Lightning Page

Action Group	Predefined Actions
1	Global actions in the order defined in the Lightning page.
2–6	Action groups 2–6 aren’t supported for “App Page” Lightning pages.

List View

Action Group	Predefined Actions
1	New
2–6	Action groups 2–6 aren’t supported for list views.

Object Home Page (Tablet Only)

Action Group	Predefined Actions
1	1. New, 2. Sort
2–6	Action groups 2–6 aren’t supported for object home pages.

Opportunity Object

Action Group	Predefined Actions
1	1. Log a Call, 2. New Task, 3. New Event, 4. Post
2	Edit
3	The remaining quick actions from the Quick Actions in the Salesforce Classic Publisher section of the Opportunity page layout. If that section isn’t customized, the remaining quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of the <i>global publisher layout</i> .

Action Group	Predefined Actions
4	Custom buttons that are supported in the Salesforce mobile app, in the order defined on the Opportunity page layout.*
5	The remaining standard buttons that are supported in the Salesforce mobile app, in the order defined on the Opportunity page layout.
6	Action group 6 isn't supported for the Opportunity object.

Person Object

Action Group	Predefined Actions
1	1. Call, 2. Send Email, 3. Post
2	Action group 2 isn't supported for the Person object.
3	The remaining actions in the order defined on the global publisher layout
4–6	Action groups 4–6 aren't supported for the Person object.

Person Account Object

Action Group	Predefined Actions
1	1. Call, 2. Send Email, 3. New Task, 4. New Event
2	Edit
3	The remaining quick actions from the Quick Actions in the Salesforce Classic Publisher section of the Person Account page layout. If that section isn't customized, the remaining quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of the <i>global publisher layout</i> .
4	Custom buttons that are supported in the Salesforce mobile app, in the order defined on the Person Account page layout.*
5	The remaining standard buttons that are supported in the Salesforce mobile app, in the order defined on the page layout.
6	Read News, Send Text, View Website

Related List (for Standard Objects)

Action Group	Predefined Actions
1	New

Action Group	Predefined Actions
2–6	Action groups 2–6 aren't supported for related lists.

Salesforce Today–Main Page

Action Group	Predefined Actions
1	Quick actions in the order defined on the global publisher layout
2–6	Action groups 2–6 aren't supported for the Salesforce Today main page.

Salesforce Today–Mobile Calendar Event

Action Group	Predefined Actions
1	1. Quick Message, 2. Join Conference Call, 3. Map
2	Action group 2 isn't supported for Salesforce Today mobile calendar events.
3	The remaining quick actions from the Quick Actions in the Salesforce Classic Publisher section of the Event page layout. If that section isn't customized, the remaining quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of the <i>global publisher layout</i> .
4–6	Action groups 4–6 aren't supported for Salesforce Today mobile calendar events.

Task Object

Action Group	Predefined Actions
1	1. Edit Comments, 2. Change Date, 3. Change Status, 4. Change Priority
2	Edit
3	The remaining quick actions from the Quick Actions in the Salesforce Classic Publisher section of the Task page layout. If that section isn't customized, the remaining quick actions are inherited from the Quick Actions in the Salesforce Classic Publisher section of the <i>global publisher layout</i> . Standard Chatter actions aren't supported.
4	Custom buttons that are supported in the Salesforce mobile app, in the order defined on the Task page layout.*
5	The remaining standard buttons that are supported in the Salesforce mobile app, in the order defined on the Task page layout.
6	Action group 6 isn't supported for the Task object.

* Custom buttons that are added to the Button section of a page layout and that define the content

source as *URL* or *Visualforce* are supported in the Salesforce mobile app. Remember that Visualforce pages must be enabled for use in the Salesforce mobile app. Custom links, custom buttons that are added to list views, and custom buttons that define the content source as *OnClick JavaScript* aren't available in the Salesforce mobile app.

See Also

- [How Actions Are Ordered in the Salesforce Mobile App Action Bar](#)
- [Salesforce Mobile App Action Bar](#)
- [Considerations for Actions in the Salesforce Mobile App](#)

Considerations for Actions in the Salesforce Mobile App

Keep these considerations in mind when you configure actions for use in the Salesforce mobile app.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Contact Manager, Database.com, and Developer Editions**

- Most actions, including quick actions, productivity actions, and standard and custom buttons, are displayed in the action bar in the Salesforce mobile app.
- The Save & New action button isn't available in the Salesforce mobile app.
- The Delete action is available on record pages only, not record detail pages.
- In most cases, a productivity action appears only if a record includes the information that the action is keyed to. For example, the Send Email action depends on the record including an email address. The View Website action requires the record to include a website URL. On accounts, the Call action appears even when there isn't a phone number on the record.
- By default, the Call productivity action in the mobile app action bar uses the global Log A Call action layout. If you create one Log a Call action on an object, users see that custom Log a Call action's layout when they tap **Call** from the object's action bar. If you create more than one Log a Call action for the same object, the Call action on that object uses the global Log a Call action layout.
- There are a few differences between the Send Email quick action in Salesforce and the standard Email action in Case Feed:
 - Users can't switch between the rich text editor and the plain text editor in a Send Email action.
 - Templates aren't supported in the Send Email action.
 - Quick Text isn't available in the Send Email action.
 - The Send Email action doesn't support attachments.
 - Users can't save messages as drafts when using the Send Email action.
 - Users can't edit or view the From field in the Send Email action.
- If feed tracking isn't enabled on an object, only nonstandard actions appear in the Salesforce mobile app action bar and in third-party apps that use action lists. Nonstandard actions include Create, Update, Log a Call, custom actions, and Mobile Smart Actions.
- If you're using record types in your org, sometimes quick actions aren't visible to your users. For more

information, see [Quick Actions and Record Types](#).

- The Mobile Smart Actions element appears as a single action element in the page layout editor, but it expands to several actions when it appears in the Salesforce mobile app. If the Mobile Smart Actions element is in Quick Actions in the Salesforce Classic Publisher section when you customize the action bar section, the actions in the app use your action bar customizations. In this case, the Mobile Smart Actions element in the Quick Actions in the Salesforce Classic Publisher section becomes irrelevant.
- To customize the actions in the Salesforce mobile app action bar for standard and custom objects, first override the predefined actions. You can then add or remove actions from the Salesforce Mobile and Lightning Experience Actions section. For instance, to move the Join Group, Edit Group, or Leave Group actions on groups in the Salesforce mobile app, override the predefined actions in the Groups page layout.
- Using URL custom buttons to pass parameters to standard pages in Salesforce Classic, such as prepopulating fields when creating a record, doesn't work in the Salesforce mobile app.
- If you add a custom button to the Salesforce Mobile and Lightning Experience Actions section of a page layout, it has a lightning bolt icon in the Salesforce mobile app action bar. If the Salesforce Mobile and Lightning Experience Actions section isn't customized, the icon in the app action bar is a wrench.

See Also

[How Actions Are Ordered in the Salesforce Mobile App Action Bar](#)

[How Predefined Actions Are Ordered in the Salesforce Mobile App Action Bar and List Item Actions](#)

[Create Dynamic Actions in Lightning App Builder](#)

Custom Buttons and Links

Custom buttons and links help you integrate Salesforce data with external URLs, applications, your company's intranet, or other internal office systems.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **All** Editions except **Database.com**

USER PERMISSIONS NEEDED

To create or change custom buttons or links: **Customize Application**

Salesforce supports these custom links and custom buttons.

Type of Custom Link or Custom Button	Setup Page Where It's Configured
Bookmark-style links defined in the standard custom links home page component	Home Page Components
Full-featured custom links included in custom	Custom Links

Type of Custom Link or Custom Button	Setup Page Where It's Configured
home page components	
Full-featured custom links or custom buttons on objects	Buttons, Links, and Actions (in the object's management settings)

Custom links can link to an external URL, such as www.google.com, a Visualforce page, or your company's intranet. Custom links can also link to a custom s-control in the custom s-control library, such as a Java applet or Active-X control.

Custom buttons can:

- Connect users to external applications, such as a web page that displays a map to a contact's address.
- Run an s-control from the s-control library, such as an s-control that escalates a case from the case detail page.
- Launch custom links.

You can choose the display window properties that determine how the target of a link or button is displayed to your users. Custom links and s-controls can include Salesforce fields as tokens within the URL or custom s-control. For example, you can include an account name in a URL that searches Yahoo: http://search.yahoo.com/bin/search?p={!Account_Name}.

You can override the default action of some standard buttons and customize the behavior of tab home pages to suit your org's needs.

Define Custom Buttons and Links

Define the action that occurs when a user clicks a custom button or link. Custom buttons and links can streamline actions within Salesforce or integrate Salesforce data with external URLs, applications, or systems.

Override Standard Buttons and Tab Home Pages

You can override the behavior of standard buttons—like New, View, or Edit—in Salesforce Classic, Lightning Experience, and mobile independently. You can also override the tab home page that displays when a user clicks a standard, custom, or external object tab.

Custom Button and Link Samples

Use samples of custom Salesforce buttons and links to determine whether they can work for you.

Custom Button and Link Considerations

Keep these considerations in mind when working with custom buttons and links.

Custom Button and Link Limitations

Keep these limitations in mind when working with custom buttons and links.

Viewing References to Salesforce Components

View a list of all the areas in Salesforce that reference a component. For example, view the custom links, custom buttons, or page layouts that reference another component, such as a Visualforce page or static resource.

See Also

- [Define Custom Buttons and Links](#)
- [Add Default Custom Links](#)
- [Salesforce Classic Home Tab Page Layouts](#)

Define Custom Buttons and Links

Define the action that occurs when a user clicks a custom button or link. Custom buttons and links can streamline actions within Salesforce or integrate Salesforce data with external URLs, applications, or systems.

REQUIRED EDITIONS

Available in: Salesforce Classic

Custom buttons and links are available in: **All Editions**

Visualforce pages and s-controls are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create or change custom buttons or links: Customize Application



If you want the button or link to launch a custom page or other code, consider a Visualforce page.

1. From the management settings for the object that you want to edit, go to Buttons, Links, and Actions.
 - Note** Custom buttons aren't available on the User object or custom home pages. Custom buttons and links are available for activities under the individual object management settings for tasks and events. To override a standard button that applies to both tasks and events, go to the object management settings for activities.
2. Click **New Button or Link**. Or, to add a predefined custom link, click **Default Custom Links**.
3. For **Display Type**, select **Detail Page Link**, **Detail Page Button**, or **List Button**.

If you select **List Button**, and your list button requires users to select individual records in a list, then select **Display Checkboxes (for Multi-Record Selection)**. When you select this option, a checkbox appears next to each list item to let users select records, and the list button action is applied to those records. Don't select **Display Checkboxes (for Multi-Record Selection)** if your list button doesn't require users to select individual records in the list. For example, your list button links to a URL that doesn't support POST operations, such as a URL that links to a Lightning component.

In Lightning Experience, when you select **Display Checkboxes (for Multi-Record Selection)**, the related list type must be set to **Enhanced List**. You can set the related list type from the Related List–Single component or Related Lists component on a record page in the Lightning App Builder.
4. Enter the [button or link attributes](#).

Here's an example of the attributes for a button that performs a web search for an account's name.

5. To validate all Salesforce merge fields and functions, click **Check Syntax**.
6. Click **Save** when you're finished, or click **Quick Save** to save and continue editing. If you set the content source to URL, saving validates the URL you defined.
7. To open a button or link using settings other than the user's default browser settings, click **Window Open Properties** on the button or link's detail page.
8. To view all references to the new button or link, click **Where is this used?** on its detail page.

Custom links for users are automatically added to the Custom Links section of the user detail page. You can add page buttons only to the Button section of a page layout.

- Note** A link URL can be up to 2,048 bytes. When data is substituted for the tokens in the URL, the link can exceed 3,000 bytes. Some browsers enforce limits on the maximum URL length.

Before you can use your custom buttons and links, add them to an object's page layout. You can then see and use the button or link on a record detail page.

Account Detail		Edit	Delete	Include Offline	Web Search
Account Name	Textiles Corp of America [View Hierarchy]				
Parent Account					
Account Site	San Francisco				
Shipping Address					
Phone	(415) 555-8973				
		Edit	Delete	Include Offline	Web Search

[Custom Button and Link Fields](#)

This table describes the fields that are available when you create a custom button or link.

[Construct Effective Custom URL Buttons and Links](#)

Use these methods to construct effective custom URL buttons and links.

[Edit Window Open Properties](#)

Custom buttons and links can open in different types of windows. If you configure a custom button or link to open in a window, set the window properties. If you leave the window properties blank, the custom button or link uses the default settings of the user's browser.

[Merge Fields for Custom Buttons and Links](#)

A merge field is a field that incorporates values from an object record. You can put merge fields in an email template, mail merge template, custom link, or formula.

[Add Default Custom Links](#)

Update predefined custom links for account, contact, lead, campaign, and solution detail pages.

[Custom Link Best Practices](#)

Custom buttons and links can streamline actions within Salesforce, or integrate Salesforce data with external URLs, applications, or systems. Use these tips to get the most out of your custom links.

See Also

[Find Object Management Settings](#)

[Custom Button and Link Considerations](#)

[Custom Button and Link Limitations](#)

[Custom Button and Link Samples](#)

Custom Button and Link Fields

This table describes the fields that are available when you create a custom button or link.

REQUIRED EDITIONS

Available in: [Salesforce Classic](#)

Custom buttons and links are available in: [All Editions](#)

Visualforce pages and s-controls are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Attribute Name	Description
Label	Enter the text that displays in the user interface for the custom button or link.
Name	Accept or enter the unique name to use for the button or link when it's referenced from a merge field. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
Namespace Prefix	Enter the prefix that uniquely identifies your package. In a packaging context, a namespace prefix is a one to 15-character alphanumeric identifier that distinguishes your package and its contents from packages of other developers on AppExchange. Namespace prefixes are case-insensitive. For example, ABC and abc aren't recognized as unique. Your namespace prefix must be globally unique across all Salesforce organizations. It keeps your managed package under your control exclusively.
Protected Component	Optionally limit use in a subscriber org. Protected components can't be linked to or referenced by components created in a subscriber org. A developer can delete a protected component in a future release without worrying about failing installations. However, after a component is marked as unprotected and is released globally, the developer can't delete it.
Description	Enter text that distinguishes the button or link from others. This text displays only to administrators when setting up buttons and links.
Display Type	<p>Determine where the button or link appears on page layouts.</p> <p>Detail Page Link Adds the link to the Custom Links section of your page layouts.</p> <p>Detail Page Button Adds the custom button to a record's detail page. You can add detail page buttons to the Button section of a page layout.</p> <p>List Button Adds the custom button to a list view, search result layout, or related list. You can add list buttons to the Related List section of a page layout or the List View and Search Result layouts.</p> <p>For list buttons, Salesforce selects the Display Checkboxes (for Multi-</p>

Attribute Name	Description
	<p>Record Selection) option to include a checkbox next to each record in the list. Users can select the records they want applied to the action on the list button. If your custom button doesn't require the user to select records, deselect this option.</p>
Behavior	<p>Choose the outcome of clicking the button or link.</p> <p>When applicable, some settings have default values. For example, if you choose Display in new window, the default height of a new window is 600 pixels. See Edit Window Open Properties.</p> <p>Some custom button and link behavior can't be changed.</p> <ul style="list-style-type: none"> Custom buttons that use a URL to link to a Visualforce page open the page in the same tab, even if the Display in new window option is selected. Custom button links open in a new tab in Experience Builder sites, even if any of the Display in existing window options is selected.
Content Source	<p>Choose whether to use a URL, s-control, JavaScript action, or Visualforce page as the content of the button or link.</p> <p>Salesforce checks the correctness of URLs in custom links and custom buttons. When you create or edit custom links or buttons that contain invalid URL markup, such as scripts, Salesforce blocks the links from rendering. Instead, an error message is shown on hover. Reconfigure the URLs to be valid and well formed. The URL can be a relative URL or an absolute <code>http://</code>, <code>https://</code>, <code>file://</code>, <code>ftp://</code>, or <code>mailto://</code> address. Invalid URLs in custom links or custom buttons created before Spring '13 aren't checked for correctness until you edit them.</p>
Content	<p>Enter the content of the button or link for buttons and links of type URL or OnClick JavaScript.</p> <ul style="list-style-type: none"> To insert a field, choose the field type from Select Field Type and choose a field from Insert Field. To insert activity merge fields, select Event or Task from Select Field Type. To insert an operator, choose the appropriate operator icon from the Insert Operator dropdown list. To insert a function, double-click its name in the list, or select it and click Insert Selected Function. To filter the list of functions, choose a category

Attribute Name	Description
	<p>from the Functions dropdown list. Select a function and click Help on this function to view a description and examples of formulas using that function.</p> <p>Internet standards require special encoding for URLs. Salesforce encodes the text from any merge field you insert into a link. Encode extra text in your link manually. For example, to generate this URL:</p> <pre data-bbox="502 572 1432 601"><code>http://www.google.com/search?q={!user.name} Steve Mark 50%</code></pre> <p>Use this content:</p> <pre data-bbox="502 756 1393 819"><code>http://www.google.com/search?q={!user.name}+Steve+Mark+50%25</code></pre> <p>Salesforce strips double quotes from URLs when the content source is a URL. If you must use double quotes, encode them manually. For example, to generate the URL <code>http://www.google.com/search?q="salesforce+foundation"</code>, use this content:</p> <pre data-bbox="479 1051 1405 1081"><code>http://www.google.com/search?q=%22salesforce+foundation%22</code></pre>
Link Encoding	<p>Choose the encoding setting. Encoding defaults to Unicode (UTF-8). Change the default encoding setting if the target of a link requires data in a different format. Encoding is available if your Content Source is URL.</p>

Construct Effective Custom URL Buttons and Links

Use these methods to construct effective custom URL buttons and links.

REQUIRED EDITIONS

Available in: Salesforce Classic

Custom buttons and links are available in: **All Editions**

Visualforce pages and s-controls are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Use Merge Fields to Pass Salesforce Data

For example, you can create custom links that search Google for an account name or look up an address

on a map.

```
http://google.com/search?q={!Account.Name}
```

```
http://maps.google.com?q={!Account.ShippingStreet} {!Account.ShippingCity} {!Account.ShippingState} {!Account.ShippingPostalCode}
```

-  **Note** Custom links don't support data type conversion. When creating custom links to pass data from one Salesforce field to another, the data must match the data type of the fields in which you're placing it. For example, if you have numeric data, you must pass it to a numeric field.

Substitute Symbols for Certain Characters

Substitute symbols for certain characters or use `urlencode()`.

Due to the World Wide Web Consortium's URL encoding standards ([W3C](#)), certain unsafe characters, such as spaces and punctuation marks, can't be passed through a URL. Custom buttons and links escape these characters, so you don't have to URL-encode them.

If you must encode your URL, use the `urlencode()` function in the merge field like so:

```
{!urlencode(text)}
```

 and replace *text* with the merge field or text string that you want to encode.

For example: If the merge field `foo_c` contains `Mark's page`, `{!urlencode(foo_c)}` results in: `%3CB%3EMark%27s%20page%3C%2Fb%3E`.

Link to Visualforce Pages

To link to a Visualforce page, use `urlfor()` with the relative path to the page, which is `/apex/PageName`. For example, to link to a Visualforce page called `MissionList` that isn't associated with a record, use the following syntax.

```
{! urlfor( "/apex/MissionList" ) }
```

When you use `urlfor()` with a Visualforce page, and you want to pass a record ID into the page, you must pass the ID in as a parameter.

```
{! urlfor( "/apex/Mission", null, [id=Mission__c.Id] ) }
```

Point to Salesforce Pages

Use the `$Action` global variable and `urlfor()` to point to Salesforce pages.

When creating a custom button or link that points to a page in Salesforce, use the `$Action` global variable to construct the link, instead of pasting in the path to the page. Then, if your organization is moved to another server or the URL to the page changes, the link still points to the right place.

To construct the link, use the `URLFOR()` formula function with the `$Action` variable.

```
{ !URLFOR( $Action.Case.NewCase, Account.Id ) }
```

This custom link on the Account object opens the New Case form, creating the case as a child of the account record. You can use this process for any object that has a lookup to the Account object. To create a record that isn't a child of another record, or if two objects have no relationship, use `$ObjectType.ObjectName` as the second argument. For example:

```
{ !URLFOR( $Action.Case.NewCase, $ObjectType.Case ) }
```

`$Action` global variables expect either a record ID or the `$ObjectType`. For example, these formulas create links to the tab and detail page for an account, respectively.

```
{ !URLFOR( $Action.Account.Tab, $ObjectType.Account ) }
```

```
{ !URLFOR( $Action.Account.View, Some_Account_Lookup__c.Id ) }
```

The `URLFOR()` function takes additional optional arguments that get passed into the destination as query string parameters. You can use these arguments when overriding a standard action with a Visualforce page to pass in the additional parameters needed by the Visualforce page or its controller. For example, if when closing a case you want to change the value of a custom field on the case called Actual Delivery Date to today, you could use:

```
{ !URLFOR($Action.Case.CloseCase, Case.Id, [ actualDeliveryDate=TODAY() ] ) }
```

You can then override the Close Case action with a Visualforce page and handle setting the value of the Actual Delivery Date field either in that Visualforce page or its controller. See [Using Query String Parameters in a Visualforce Page](#) for more information.

Point to Other Pages in Lightning Experience

Use these Lightning-friendly URL button and link methods to point to other pages in Lightning Experience.

Custom URL Button or Link	Lightning Experience Behavior
External URL www.google.com	URL opens in new tab
Relative Salesforce URL, View /{!Account.Id}	Record home page opens in existing tab
Relative Salesforce URL, Edit /{!Account.Id}/e	Edit overlay pops up on the existing page
Relative Salesforce URL, List /001/o	Object home page opens in existing tab
\$Action URL, View { !URLFOR(\$Action.Account.View, Account.Id) }	Record home page opens in existing tab
\$Action URL, Edit { !URLFOR(\$Action.Account.Edit, Account.Id) }	Edit overlay pops up on the existing page

 **Note** Using URL custom buttons to pass parameters to standard pages in Salesforce Classic, such as prepopulating fields when creating a record, doesn't work in the Salesforce mobile app.

See Also

- [Custom Link Example: Link to Documents](#)
- [Custom Link Example: Link to Files in Chatter](#)
- [Custom Link Example: Link to Reports](#)

Edit Window Open Properties

Custom buttons and links can open in different types of windows. If you configure a custom button or link to open in a window, set the window properties. If you leave the window properties blank, the custom button or link uses the default settings of the user's browser.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **All** Editions except **Database.com**

S-controls are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To edit custom button or link properties: Customize Application

To edit the window open properties.

1. From the management settings for the object whose button or link you want to edit, go to Buttons, Links, and Actions, and then click the button or link name. Custom buttons aren't available on the user object.
2. Click **Window Open Properties**.
3. Edit the window properties.

Window Property	Description
Width	The window width (in pixels).
Height	The window height (in pixels).
Window Position	The location on the screen where you want the window to open.
Resizeable	Allow users to resize the window.
Show Address Bar	Show the browser's address bar that contains the URL.
Show Scrollbars	Show browser scrollbars in the window.
Show Toolbars	Show the browser toolbars. Toolbars normally contain navigation buttons like Back, Forward, and Print.
Show Menu Bar	Show the browser menus. The menus typically contain option like File and Edit.

Window Property	Description
Show Status Bar	Show the status bar at the bottom of the browser.

 **Note** Some properties aren't available depending on the behavior of the custom button or link.

For example, if you chose *Execute JavaScript*, no window open properties are available.

4. Save your changes.

See Also

[Define Custom Buttons and Links](#)

Merge Fields for Custom Buttons and Links

A merge field is a field that incorporates values from an object record. You can put merge fields in an email template, mail merge template, custom link, or formula.

REQUIRED EDITIONS

Available in: Salesforce Classic

Custom buttons and links are available in: **All Editions**

Visualforce pages and s-controls are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Syntax and Formatting

When you insert a merge field in a custom button or link, the syntax consists of an open curly brace and exclamation point, followed by the object name, a period, the field name, and a closing curly brace.

```
{ !Object_Name.Field_Name }
```

To ensure that you're using the correct syntax, select merge fields from the dropdown list in the editor for custom buttons and links.

 **Note** Custom objects named Org can't be used in a merge field. Objects named org are explicitly filtered from the Select Field Type list for a custom button.

Tips

- To insert activity merge fields, select **Event** or **Task** from the Select Field Type dropdown list.
- You can add links quickly to the sidebar by using the standard home page's Custom Links component.



- Warning** The standard home page's Custom Links component doesn't support
- Merge fields
 - Functions, such as `URLFOR`
 - JavaScript execution
 - Customizable window opening properties

See Also

[Generate Emails From Records](#)
[Custom Button and Link Considerations](#)

Add Default Custom Links

Update predefined custom links for account, contact, lead, campaign, and solution detail pages.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **All** Editions except **Database.com**

USER PERMISSIONS NEEDED

To create or change custom links: [Customize Application](#)

1. From the management settings for the appropriate object, go to Buttons, Links, and Actions or to Buttons and Links.
2. Click **Default Custom Links**.
3. Next to a sample link you want to add, click **Add Now!**.
4. Change the default data for the link, as necessary.
5. Save your changes.
6. To display the new link, [edit the page layout](#) for the appropriate tab.

See Also

[Define Custom Buttons and Links](#)

Custom Link Best Practices

Custom buttons and links can streamline actions within Salesforce, or integrate Salesforce data with external URLs, applications, or systems. Use these tips to get the most out of your custom links.

REQUIRED EDITIONS

Available in: Salesforce Classic

Custom buttons and links are available in: **All Editions**

Visualforce pages and s-controls are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

-  **Note** Salesforce checks the correctness of URLs in custom links and custom buttons. When you create or edit custom links or buttons that contain invalid URL markup, such as scripts, Salesforce blocks the links from rendering. Instead, an error message is shown on hover. Reconfigure the URLs to be valid and well formed. The URL can be a relative URL or an absolute `http://`, `https://`, `file://`, `ftp://`, or `mailto://` address. Invalid URLs in custom links or custom buttons created before Spring '13 aren't checked for correctness until you edit them.

Field Names

Salesforce has no plans to change field names; however, that doesn't guarantee that field names won't change in the future. Therefore, custom links that include Salesforce fields could change how they're mapped.

Pass Session IDs with SSL

Never pass session IDs to an `http` URL. Instead, pass session IDs with a secure sockets layer (SSL) `https` URL. Always use SSL for any data that is passed to other applications hosted on the Internet since the URL could contain sensitive customer information.

Single Sign-On

Use custom links to pass a session ID to support Single Sign-On (SSO), so users can avoid multiple logins to web applications that your organization hosts to manage Salesforce data. Construct your custom link to pass the `{!User_Session_ID}` merge field, which allows users to access all authorized resources during a single authentication. External systems can access Salesforce resources using a web service, which allows organizations to communicate data without intimate knowledge of each other's IT systems behind a firewall.

Send the Server Date in the URL

Some integration projects need custom links to include a server date to know the Salesforce initiation date. The Salesforce server date can be passed to external systems using custom links. For example, `http://someurl.com/somepath?current_date={!Today}`. Dates use the Pacific time zone.

Avoid Double Sets of Tabs in a Browser

Unlimited Edition and Enterprise Edition users can build a custom link to perform an action that keeps

users in the same browser window and doesn't display a double set of tabs. Create a Visualforce page that contains the following code, replacing <REGULAR_WIL> with your regular or existing custom link.

```
<script language="JavaScript">
function redirect() {
parent.frames.location.replace("<REGULAR_WIL>")
}
redirect();
</script>
```

Override Standard Buttons and Tab Home Pages

You can override the behavior of standard buttons—like New, View, or Edit—in Salesforce Classic, Lightning Experience, and mobile independently. You can also override the tab home page that displays when a user clicks a standard, custom, or external object tab.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

Visualforce overrides also available in: **Contact Manager, Group, and Professional** Editions

Record types available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To override standard buttons and tab home pages: Customize Application

To reset button and tab home page overrides: Customize Application

Button overrides are global. For example, if you override the New button on opportunities, your replacement action takes effect wherever that action is available, including:

- Opportunities home page
 - Any opportunities related lists on other objects, such as accounts
 - Create New dropdown list in the Salesforce Classic sidebar
 - Any browser bookmarks for this Salesforce page
1. From the object management settings for the object you want to set an override for, go to Buttons, Links, and Actions.
 2. Click **Edit** next to the button or tab home page you want to override.
 3. For each experience—Salesforce Classic, Lightning Experience, or mobile—click the type of override you want associated with the action.

You have a few options for overrides.

! **Important** Before you override a standard button with a Lightning component or Visualforce page, review implementation details in the respective developer guides.

- **No override (use default)**—Use a custom override provided by an installed package. If there isn't one installed, the standard Salesforce behavior is used.
 - **Standard page**—This option is available only for subscribers who are overriding the actions on an installed custom object. If selected, the standard Salesforce behavior is used.
 - **Custom s-control**—Use the behavior from an s-control. This option isn't supported for mobile.
- !** **Important** Visualforce pages supersede s-controls. Organizations that haven't previously used s-controls can't create them. Existing s-controls are unaffected and can still be edited.
- **Lightning component**—Use the behavior from a Lightning component. Supported only for the Edit, New, New Event, Tab, and View actions. This option isn't supported for Salesforce Classic.
 - **Lightning page**—Use the behavior from the Lightning record page assigned as the org default for the object. This option is available only for the View action in Lightning Experience.
 - **Visualforce page**—Use the behavior from a Visualforce page.
 - **Use the Salesforce Classic override**—Inherits the behavior from the Salesforce Classic Override setting.
4. Select the name of the s-control, Lightning component, Lightning page, or Visualforce page you want to run when users click the button or tab.

When overriding the New button with a Visualforce page, you can choose to skip the record type selection page. If you do, new records you create aren't forwarded to the record type selection page. Salesforce assumes that your Visualforce page is already handling record types.

! **Important** When a Salesforce mobile app user clicks **New** to create a product, the user must select a record type even if the **Skip record type selection page** option is selected in Setup.

5. Save your changes

! **Note** A standard button—New, Edit, View, Delete, and Clone—that is overridden with a Visualforce page doesn't show up in the Salesforce mobile app unless the Visualforce page is enabled for Salesforce mobile apps. Overriding standard list and tab controls isn't supported in mobile.

Standard Action Overrides

For standard actions, such as Delete, Edit, List, New, Tab, and View, you can provide a custom user interface for the action, called an action override. Use action overrides when your business model requires a more customized user experience than the Salesforce standard page provides.

Considerations for Overriding Standard Buttons

Before you override a standard button, review these considerations.

Remove Overrides for Standard Buttons and Tab Home Pages

Remove applied overrides for standard buttons and links in Salesforce Classic, Lightning Experience, and the Salesforce mobile app.

See Also

[Considerations for Overriding Standard Buttons](#)[Remove Overrides for Standard Buttons and Tab Home Pages](#)[Visualforce Developer Guide: Overriding Buttons, Links, and Tabs with Visualforce](#)[Lightning Aura Components Developer Guide: Standard Actions and Overrides Basics](#)

Standard Action Overrides

For standard actions, such as Delete, Edit, List, New, Tab, and View, you can provide a custom user interface for the action, called an action override. Use action overrides when your business model requires a more customized user experience than the Salesforce standard page provides.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

Visualforce overrides also available in: **Contact Manager, Group, and Professional Editions**

Record types available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To override standard buttons:	Customize Application
-------------------------------	-----------------------

You can specify a different override for each user experience, with the following guidelines.

- Salesforce Classic—Use a Visualforce page as the action override.
- Lightning Experience and the Salesforce mobile app –You can override the Edit, New, Tab, and View standard actions on most standard and all custom objects.
- Lightning Experience—Use a Lightning component as the action override for the Edit, New, and Tab actions. Use a Lightning page or a Lightning component as the action override for the View action.
- Aura Experience Builder sites—Use a custom Lightning component to replace standard forms when users click the New or Edit button. The action that you choose in Lightning Experience is the same action that you use to override actions in Aura Experience Builder sites.
- Salesforce app—Use a Lightning component as the action override.
- Managed packages—Developers can include action overrides as package defaults, and managed package subscribers can specify local overrides as alternatives to the default overrides.

 **Note** For Salesforce Classic, usually a Visualforce page is the only supported override option. Under certain conditions, you can use existing s-controls as overrides for Salesforce Classic. However, s-controls have been deprecated since the Spring '09 release. We recommend using Visualforce pages instead.

[Assign Action Overrides](#)

To assign a Visualforce page or Lightning component as an override, first create the page or

component (or use one created by a Salesforce developer in your org). It's important to understand the action override options for each user experience and how your selections for each one can affect the others.

[Assign a Lightning Record Page Override for the View Action](#)

When you create an override for the View action for Lightning Experience, you can use either a Lightning record page or a Lightning component. To use a Lightning record page as an override, you must activate the page in the Lightning App Builder and choose the type of override you want.

[Action Overrides in Managed Packages](#)

In a managed package, overrides specified by the package developer are included as default overrides. The package subscriber can specify local overrides for each user experience to use instead of the package default overrides. Package default overrides impact the options and behaviors of local overrides.

[Action Overrides in Aura Experience Builder Sites](#)

Personalize your site users' experience in Aura Experience Builder sites by adding a custom Lightning component to replace standard forms when users click the New or Edit button. Use action overrides when your site and portal users require a more customized user experience than the Salesforce standard page provides.

See Also

[How Do Visualforce Pages Compare to S-Controls?](#)

[Trailhead: Introduction to Creating Visualforce Pages](#)

[Lightning Aura Components Developer Guide: Override Standard Actions with Aura Components](#)

Assign Action Overrides

To assign a Visualforce page or Lightning component as an override, first create the page or component (or use one created by a Salesforce developer in your org). It's important to understand the action override options for each user experience and how your selections for each one can affect the others.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

Visualforce overrides also available in: **Contact Manager, Group, and Professional** Editions

Record types available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To override standard buttons:

Customize Application

When assigning overrides, keep in mind the following requirements.

- Salesforce Classic—To use a Visualforce page to override an action, the Visualforce page must use the standard controller for the object on which the action appears. To use a Visualforce page to override a tab, the Visualforce page must use the standard list controller for that tab’s associated object.
 - Lightning Experience and the Salesforce mobile app—To make a Visualforce page available for use as an override, select **Available for Lightning Experience, Experience Builder sites, and the mobile app** in the Visualforce Pages setup panel.
1. From Setup, select **Object Manager**, and select the object that you want to modify.
 2. In the Object Manager settings list, select **Buttons, Links, and Actions**.
 3. In the row for the action you want to modify, select **Edit** from the dropdown menu.
 4. Specify the override option for each user experience.



Example This Override Properties panel specifies a Visualforce page for Salesforce Classic and a Lightning component for Lightning Experience. The mobile override specifies the Salesforce Classic override, so mobile users see the Visualforce page.



See Also

[Trailhead: Introduction to Creating Visualforce Pages](#)

[Lightning Aura Components Developer Guide: Override Standard Actions with Aura Components](#)

[Visualforce Developer Guide: Overriding Buttons, Links, and Tabs with Visualforce](#)

[Visualforce Developer Guide: Overriding Tabs Using a Standard List Controller](#)

Assign a Lightning Record Page Override for the View Action

When you create an override for the View action for Lightning Experience, you can use either a Lightning record page or a Lightning component. To use a Lightning record page as an override, you must activate the page in the Lightning App Builder and choose the type of override you want.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

Visualforce overrides also available in: **Contact Manager, Group, and Professional Editions**

Record types available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To override standard buttons:

Customize Application

1. From Setup, enter *Lightning App* in the Quick Find box, and select **Lightning App Builder**.
2. Click **Edit** next to the Lightning page that you want to use as an override.
3. In the Lightning App Builder, click **Activation**.

You have a few options for activating a Lightning record page as an override.

- Make the page the org default for the object.

After you activate the Lightning page as the org default, the page is selected as the Lightning Experience override for the View action in the Override Properties panel.

- Make the page the default object record page for specific Lightning apps.

If you activate the Lightning record page for specific Lightning apps only, the page takes precedence over the Lightning Experience Override setting for the View action on the object in those apps.

- Assign the page to a combination of Lightning apps, record types, and profiles.

4. To remove the Lightning page as an override, return to the Activation page in the Lightning App Builder, select the type of override you want to remove, and walk through the activation wizard.



Example In this Override Properties panel, the Lightning page `Account_Lightning_Page_Test` is the specified Lightning Experience override for the View action.



See Also

[Create and Configure Lightning Experience Record Pages](#)

Action Overrides in Managed Packages

In a managed package, overrides specified by the package developer are included as default overrides.

The package subscriber can specify local overrides for each user experience to use instead of the package default overrides. Package default overrides impact the options and behaviors of local overrides.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

Visualforce overrides also available in: **Contact Manager, Group, and Professional Editions**

Record types available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To override standard buttons:

Customize Application

Several rules determine local override behaviors in a managed package.

- For each user experience—When a package subscriber specifies a local override, the local override takes precedence over the package default override. If no local override is specified for a user experience, the package default override for that user experience is used.
- For Lightning Experience and the Salesforce mobile app—if no local override is specified and no package default override is specified, the local override inherits from Salesforce Classic. Either the Salesforce Classic local override (if it's specified) is used, or the Salesforce Classic package default

override (if no Salesforce Classic local override is specified) is used.

- A package developer can remove or replace the default overrides when updating a managed package. The new default overrides are available for package subscribers when they install the updated package.



Example The Override Properties panel for a managed package identifies the override defaults and the local override options for each user experience. The available local override options vary based on the override defaults.

- **No override (use default)**—Uses either the override default Visualforce page or the Salesforce standard page if no Visualforce page is specified.
- **Standard page**—Available for package subscribers only if the package default override for the given user experience is a Visualforce page or Lightning component.
- **Lightning component**—The specified component takes precedence over the package default override for the given user experience.
- **Use the package default override**—Available for package subscribers only if the package default override for the given user experience is a Lightning component.
- **Use the Salesforce Classic override**—Available for package subscribers only if the package default override for the given user experience is the Salesforce standard page. Option inherits from the local Salesforce Classic override.

Override Properties		Save	Cancel
Label	Edit		
Name	Edit		
Salesforce Classic Override Default	Standard page		
Lightning Experience Override Default	mm210de1:ForEditRecord (Lightning component, Package: mainde1)		
Mobile Override Default	Standard page		
Salesforce Classic Override	<input checked="" type="radio"/> No override (use default) <small>(i)</small> <input type="radio"/> Visualforce page <small>--None--</small>		
Lightning Experience Override	<input type="radio"/> Standard page <input type="radio"/> Lightning component <small>--None--</small> <input checked="" type="radio"/> Use the package default override		
Mobile Override	<input type="radio"/> Lightning component <small>--None--</small> <input checked="" type="radio"/> Use the Salesforce Classic override		

Action Overrides in Aura Experience Builder Sites

Personalize your site users' experience in Aura Experience Builder sites by adding a custom Lightning component to replace standard forms when users click the New or Edit button. Use action overrides when your site and portal users require a more customized user experience than the Salesforce standard page provides.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

Applies to: [Aura sites](#)

USER PERMISSIONS NEEDED

To override standard buttons: [Customize Application](#)

To override actions in an Aura Experience Builder site:

1. From Setup, in the Quick Find box, enter *Digital Experiences*, and then select **All Sites**.
2. Next to the site that you want to override, click **Workspaces**.
3. Click **Administration**.
4. In the Preferences section, under Experience Management, select **Override standard actions with the Lightning component**.
The action that you choose in Lightning Experience is the same action that you use to override actions in Experience Builder sites.
5. Save your changes.

 **Note** Action overrides aren't available in Visualforce sites or Lightning Web Runtime (LWR) sites.

Considerations for Overriding Standard Buttons

Before you override a standard button, review these considerations.

REQUIRED EDITIONS

Available in: **Salesforce Classic and Lightning Experience**

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

Visualforce overrides also available in: **Contact Manager, Group, and Professional Editions**

Record types available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To override standard buttons: [Customize Application](#)

Implementation Tips

 **Important** Before you override a standard button with a Lightning component or Visualforce page, review implementation details in the respective developer guides.

- If you override a standard button in Salesforce, that button is still available in Connect Offline, but it retains its original behavior.
- If a button isn't available for overrides, you can still hide it on the page layout.

- Button overrides affect everywhere that action or behavior is available. For example, overriding the New button on an account also overrides the account option in the Create New dropdown list in the Salesforce Classic sidebar.
- Person Account records use any standard button overrides that you make for accounts. Person Account records also use any overrides for the View Self-Service and Enable Self-Service buttons that you make for contacts.
- If your organization uses the Console tab, overrides for the Edit and View buttons for an object don't affect the Edit and View buttons in the mini page layouts. Pages that display due to overrides display in the console without the header or sidebar.
- To replace a standard button with a custom button, first define the custom button, then customize the page layout to hide the standard button and display the custom one in its place.
- When you override the Edit button with a Visualforce page, you're also overriding default logic that checks whether the record has been locked for approval. However, you can perform the same check by implementing the `Approval.lock` Apex method.

Implementation Tips Specific to the View Action

- The View standard button refers to all links in Salesforce that display the detail page for a record. Overriding the View standard button reroutes all of these links.
- The View action is the only one that supports overrides with a Lightning record page. To make a Lightning record page available for a View action override, activate it in the Lightning App Builder.
- Activating the Lightning record page assigns it as the org default for its associated object. If you activate the Lightning record page for specific Lightning apps only, the page takes precedence over the Lightning Experience Override setting for the View action on the object in those apps.
- If you have the View action overridden with a Visualforce page in Salesforce Classic, the Setup menu on that object record page in Lightning Experience displays the Edit Page option. Selecting **Edit Page** in Lightning Experience on an object page that's overridden with a Visualforce page in Salesforce Classic lets you create a custom Lightning Experience record page for that object in the Lightning App Builder.

Limitations

- You can override buttons on the detail page but not the edit page of a record.
- You can only override these standard buttons: New, View, Edit, and Delete.
- For tasks in Salesforce Classic, you can only override standard buttons and links.
- You can't change buttons on lookup dialogs, reports, or tabs. However, you can change the buttons on list view and search result layouts under search layouts.
- Action overrides on the New standard button don't work on New *Object* links in lookup searches.
- Action overrides on the Edit standard button disable inline editing on the corresponding record detail component in Lightning experience.
- For objects with multiple record types, the Outlook and Gmail integrations don't support action overrides unless you choose to skip the record type selection page.
- You can't relabel or relocate standard buttons on a record detail page.
- When overriding tabs or buttons with a Lightning component, you can select only Lightning

components that implement the `lightning:actionOverride` interface.

- Visualforce overrides to standard actions such as View, New, Edit, and Delete aren't supported in Experience Builder sites. When Visualforce overrides are applied to these actions, the buttons don't appear on record detail pages in sites.
- A standard button (New, Edit, View, Delete, and Clone) overridden with a Visualforce page doesn't show up in the Salesforce mobile app unless the Visualforce page is enabled for Salesforce mobile apps. Overriding standard list and tab controls isn't supported in mobile.
- When overriding a standard action with an Aura component on an object with more than one active record type, the record type selection screen uses Salesforce Classic styling on mobile devices.
- When overriding tabs with a Visualforce page, you can select only Visualforce pages that use the standard list controller for that tab's associated object, pages with a custom controller, or pages with no controller.
- When overriding lists with a Visualforce page, you can select only Visualforce pages that use a standard list controller.
- When overriding buttons with a Visualforce page, you can select only Visualforce pages that use the standard controller for the object on which the button appears. For example, if you want to use a page to override the **Edit** button on accounts, the page markup must include the `standardController="Account"` attribute on the `<apex:page>` tag:

```
<apex:page standardController="Account">  
  
    ... page content here ...  
  
</apex:page>
```

See Also

[Override Standard Buttons and Tab Home Pages](#)

[Visualforce Developer Guide: Overriding Buttons, Links, and Tabs with Visualforce](#)

[Lightning Aura Components Developer Guide: Standard Actions and Overrides Basics](#)

Remove Overrides for Standard Buttons and Tab Home Pages

Remove applied overrides for standard buttons and links in Salesforce Classic, Lightning Experience, and the Salesforce mobile app.

REQUIRED EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

Visualforce overrides also available in: **Contact Manager, Group, and Professional** Editions

Record types available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To override standard buttons and tab home pages: Customize Application

To reset button and tab home page overrides: Customize Application

1. From the management settings for the object whose button or link you want to edit, go to Buttons, Links, and Actions.

2. Next to the standard button or link that has overrides applied, click **Edit**.

3. For the Salesforce Classic override, select **No override (use default)**.

For actions that are supported only in Salesforce Classic, you see these options.



4. To also remove overrides for Lightning Experience or the Salesforce mobile app, select **Use the Salesforce Classic override**.

For actions that are also supported in Lightning Experience and the Salesforce mobile app, you see these options.



5. Save your changes.

Custom Button and Link Samples

Use samples of custom Salesforce buttons and links to determine whether they can work for you.

REQUIRED EDITIONS

Available in: Salesforce Classic

Custom buttons and links are available in: **All Editions**

Visualforce pages and s-controls are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create or change custom buttons or links: Customize Application

[Custom Link Example: Link to Documents](#)

Use custom links to reference documents from a Salesforce record detail page.

[Custom Link Example: Link to Files in Chatter](#)

Use custom links to reference files from Chatter.

[Custom Link Example: Link to Reports](#)

Use custom links to run reports with filtered results from a Salesforce record detail page. For example, let's say you frequently run a mailing list report for the contacts related to an account. You can create a custom link for accounts that links directly to a report that is automatically filtered to the account that you're viewing. In this case, your custom link must pass the account's unique record ID to the report.

Custom Button Example: Mass Delete

This example creates a JavaScript custom button for Salesforce Classic that can be added to activity-related lists and list views and allows users to delete selected records at the same time.

Custom Button Example: Display Alerts

This example creates a button that opens a window with a welcome message containing the user's first name.

Custom Button Example: Clone Records

The clone button is available for most standard objects. When a clone button is not available by default for a standard or custom object, you can create a custom clone button or link. This example creates a button that passes the field values of the record you're on to a record create page.

Custom Button Example: Get Record IDs

This example creates a button that opens a window listing record IDs for user selected records. Getting record IDs is useful when testing to ensure that you have the correct records before processing them further.

Custom Button Example: Pass Record IDs to an External System

You can use Salesforce record IDs as unique identifiers for integrating with an external system. This example creates a button that calls a Visualforce page to determine the record IDs of selected records and passes them in a URL query parameter to an external Web page called www.yourwebsitehere.com.

Custom Button Example: Record Create Page with Default Field Values

Construct custom buttons and links that pass default field values to a record create page. This feature applies to Lightning Experience in all editions. This feature doesn't apply to Lightning Out, Experience Builder sites, or the Salesforce mobile app.

Custom Button Example: Reopen Cases

This example creates a button that can be added to cases related lists so that users can reopen several cases on an opportunity at once.

Custom Link Example: International Maps

This example creates a link that displays a country-specific Google map.

See Also

[Define Custom Buttons and Links](#)

Custom Link Example: Link to Documents

Use custom links to reference documents from a Salesforce record detail page.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Custom buttons and links are available in: **All Editions**

Visualforce pages and s-controls are available in: **Contact Manager, Group, Professional, Enterprise**

Performance, Unlimited, and Developer Editions

USER PERMISSIONS NEEDED

To create or change custom buttons or links: [Customize Application](#)

1. Create a folder on the Documents tab to which all users have access.
2. Upload the document to that folder.
3. From the Documents tab, choose the folder and click **Go**.
4. Click **View** next to the document.
5. Copy the document's URL from the browser. For example,
`https://MyDomainName.my.salesforce.com/servlet/
servlet.FileDownload?file=015300000000xvU.`
6. Use everything after the domain portion of the URL to create your custom link. Using the example in the previous step, your link would point to `/servlet/
servlet.FileDownload?file=015300000000xvU.`

See Also

[Construct Effective Custom URL Buttons and Links](#)

[Custom Link Example: Link to Files in Chatter](#)

[Custom Link Example: Link to Reports](#)

Custom Link Example: Link to Files in Chatter

Use custom links to reference files from Chatter.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Custom buttons and links are available in: **All Editions**

Visualforce pages and s-controls are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create or change custom buttons or links: [Customize Application](#)

1. Upload a file to the Files tab.
2. When the upload is finished, from the Upload dialog box, click **Share settings**.
3. Click **Anyone with link**.
4. Copy the document's URL from the Share via Link dialog box. For example,
`https://MyDomainName.my.salesforce.com/sfc/p/D0000000JsES/a/D00000001dd/
aiq8UPJ5q5i6Fs4Sz.IQLKUErswYdbAm320cjqWnkfk=.`

5. Use everything after the domain portion of the URL to create your custom link. Using the example in the previous step, your link would point to `/sfc/p/D0000000JsES/a/D00000001dd/aiq8UPJ5q5i6Fs4Sz.IQLKUERsWYdbAm320cjqWnkfk=.`

See Also

[Construct Effective Custom URL Buttons and Links](#)

[Custom Link Example: Link to Documents](#)

[Custom Link Example: Link to Reports](#)

Custom Link Example: Link to Reports

Use custom links to run reports with filtered results from a Salesforce record detail page. For example, let's say you frequently run a mailing list report for the contacts related to an account. You can create a custom link for accounts that links directly to a report that is automatically filtered to the account that you're viewing. In this case, your custom link must pass the account's unique record ID to the report.

REQUIRED EDITIONS

Available in: Salesforce Classic

Custom buttons and links are available in: **All Editions**

Visualforce pages and s-controls are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create or change custom buttons or links: [Customize Application](#)

1. Copy the ID for the type of record by which you want to filter your report. This example uses an account record. To do so, view the record and copy the 15-character ID from the last part of the URL. For example, from `https://MyDomainName.my.salesforce.com/001200030012j3J`, copy **001200030012j3J**.
2. From the Reports tab, create the report you want by either customizing a standard report or creating a custom report.
3. Filter the report by the record ID you copied. For example, Account ID equals 001200030012j3J.
4. Run the report and verify that it contains the data you expect.
5. Click **Customize**.
6. To save the report to a public folder where it's accessible by the appropriate users, click **Save** or **Save As**. **Save** doesn't create a custom report, whereas **Save As** does.
7. Run the report and copy the report's URL from the browser.
8. Begin creating your custom link. Set the **Content Source** field to URL. In the large formula text area, paste the report URL that you copied. Remember to omit the domain portion `https://MyDomainName.my.salesforce.com`.
9. Add the custom link to the appropriate page layouts.

10. Verify that the new custom link works correctly.

 **Tip** When creating a report for use in a custom link, set date ranges and report options generically so that report results include data that can be useful for multiple users. For example, if you set a date range using a record's Created Date, set the Start Date far enough in the past to not exclude any relevant records and leave the End Date blank. If you scope the report to just My records, the report doesn't always include all records that a user can see. Try setting the report options to All visible records.

See Also

[Construct Effective Custom URL Buttons and Links](#)

[Custom Link Example: Link to Documents](#)

[Custom Link Example: Link to Files in Chatter](#)

Custom Button Example: Mass Delete

This example creates a JavaScript custom button for Salesforce Classic that can be added to activity-related lists and list views and allows users to delete selected records at the same time.

REQUIRED EDITIONS

Available in: Salesforce Classic

Custom buttons and links are available in: **All Editions**

Visualforce pages and s-controls are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create or change custom buttons or links: [Customize Application](#)

 **Note** JavaScript custom buttons are supported in all editions, in Salesforce Classic only. The mass delete function described here doesn't work in editions where the API isn't enabled.

1. Define a button for cases with these attributes.

Display Type	List Button Select Display Checkboxes (for Multi-Record Selection) so users can select multiple records in the list before clicking the button.
Behavior	Execute JavaScript
Content Source	OnClick JavaScript

Use this sample code:

```
{!REQUIRESCRIPT("/soap/ajax/9.0/connection.js")}

var records = {!GETRECORDIDS( $ObjectType.Event )};
var taskRecords = {!GETRECORDIDS( $ObjectType.Task )};
records = records.concat(taskRecords);

if (records[0] == null) {
    alert("Please select at least one record.") }
else {

    var errors = [];
    var result = sforce.connection.deleteIds(records);
    if (result && result.length){
        var numFailed = 0;
        var numSucceeded = 0;
        for (var i = 0; i < result.length; i++){
            var res = result[i];
            if (res && res.success == 'true'){
                numSucceeded++;
            } else {
                var es = res.getArray("errors");
                if (es.length > 0) {
                    errors.push(es[0].message);
                }
                numFailed++;
            }
        }
        if (numFailed > 0){
            alert("Failed: " + numFailed + "\nSucceeded: " + numSucceeded + "\n Due to:
" + errors.join("\n"));
        } else {
            alert("Number of records deleted: " + numSucceeded);
        }
    }
    window.location.reload();
}
```

2. Add the button to your activity list views.
3. Add the button to any page layout that contains an activity-related list. The button deletes any selected task or event in the list.

You can install custom buttons from the Mass Delete app at <https://appexchange.salesforce.com/>.

See Also

[Custom Button and Link Samples](#)

Custom Button Example: Display Alerts

This example creates a button that opens a window with a welcome message containing the user's first name.

REQUIRED EDITIONS

Available in: Salesforce Classic

Custom buttons and links are available in: **All Editions**

Visualforce pages and s-controls are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create or change custom buttons or links: [Customize Application](#)

1. Define a button with these attributes.

Display Type	Detail Page Button
Behavior	Execute JavaScript
Content Source	OnClick JavaScript

Use this sample code.

```
alert ("Hello {!$User.FirstName}");
```

2. Add the button to the appropriate page layout.

See Also

[Custom Button and Link Samples](#)

Custom Button Example: Clone Records

The clone button is available for most standard objects. When a clone button is not available by default for a standard or custom object, you can create a custom clone button or link. This example creates a button that passes the field values of the record you're on to a record create page.

REQUIRED EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **All Editions**

USER PERMISSIONS NEEDED

To create or change custom buttons or links: **Customize Application**

1. Define a button with these attributes.

Display Type	Detail Page Button
Behavior	Display in new window
Content Source	URL

Use this sample formula to create a clone button for the Task object.

```
/ {!Task.Id} /e?clone=1
```

To create a clone button for other objects, replace **Task** in the formula with the API name of the object.

2. Add the button to the appropriate page layout.

See Also

[Custom Button and Link Samples](#)

Custom Button Example: Get Record IDs

This example creates a button that opens a window listing record IDs for user selected records. Getting record IDs is useful when testing to ensure that you have the correct records before processing them further.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Custom buttons and links are available in: **All Editions**

Visualforce pages and s-controls are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

1. Define a button with these attributes.

Display Type	List Button
	Select Display Checkboxes (for Multi-Record Selection) so users can select multiple records in

	the list before clicking the button.
Behavior	Execute JavaScript
Content Source	OnClick JavaScript

Use this sample code. This example is for contacts. Change the object type for a different type of record.

```
idArray = {!GETRECORDIDS($ObjectType.Contact)};  
alert("The Ids you have selected are: "+idArray);
```

2. Add the button to the appropriate related list on a page layout or list view layout.

See Also

[Custom Button and Link Samples](#)

Custom Button Example: Pass Record IDs to an External System

You can use Salesforce record IDs as unique identifiers for integrating with an external system. This example creates a button that calls a Visualforce page to determine the record IDs of selected records and passes them in a URL query parameter to an external Web page called www.yourwebsitehere.com.

REQUIRED EDITIONS

Available in: Salesforce Classic

Custom buttons and links are available in: **All Editions**

Visualforce pages and s-controls are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create or change custom buttons or links: [Customize Application](#)

1. Create a Visualforce page that uses the GETRECORDIDS function to retrieve a list of selected records:

```
<script type="text/javascript">  
  
idArray = {!GETRECORDIDS($ObjectType.Account)};  
  
window.location.href="http://www.yourwebsitehere.com?array="+idArray;
```

```
</script>
```

 **Note** Replace `www.yourwebsitehere.com` with your own URL.

- Define a button for accounts with these attributes.

Display Type	List Button Select Display Checkboxes (for Multi-Record Selection) so users can select multiple records in the list before clicking the button.
Behavior	Display in existing window with sidebar.
Content Source	Visualforce page.

- Select the Visualforce page you created in the first step.
- Add the button to the appropriate page layout or list view layout.

See Also

[Custom Button and Link Samples](#)

Custom Button Example: Record Create Page with Default Field Values

Construct custom buttons and links that pass default field values to a record create page. This feature applies to Lightning Experience in all editions. This feature doesn't apply to Lightning Out, Experience Builder sites, or the Salesforce mobile app.

- Define a custom button or link, and encode any field that is read from a record and contains special characters, like commas.

Use this sample formula.

```
/lightning/o/Account/new?defaultFieldValues=
    Name={!URLENCODE(Account.Name)},
    OwnerId={!Account.OwnerId},
    AccountNumber={!URLENCODE(Account.AccountNumber)},
    NumberOfEmployees=35000,
    CustomCheckbox__c={!IF(Account.SomeCheckbox__c, true, false)}
```

 **Important** The `URLENCODE` function works only when creating custom buttons and links. You can't use it for custom fields. The `TEXT` function is supported for custom number fields, but it returns only the numbers without any separators, like commas.

- Add the button or link to the appropriate page layout.

 **Note** Passing the `RecordTypeId` to `defaultFieldValues` isn't yet supported. The

`recordTypeId` influences routing behavior, layout assignment, and page assignment, so you can see unexpected results if you try to use it.

Custom Button Example: Reopen Cases

This example creates a button that can be added to cases related lists so that users can reopen several cases on an opportunity at once.

REQUIRED EDITIONS

Available in: Salesforce Classic

Custom buttons and links are available in: **All** Editions

Visualforce pages and s-controls are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create or change custom buttons or links: Customize Application

1. Define a button for cases with these attributes.

Display Type	List Button
Behavior	Execute JavaScript
Content Source	OnClick JavaScript

Use this sample code.

```
{!REQUIRESCRIPT (" /soap/ajax/13.0/connection.js") }
var records = {!GETRECORDIDS($ObjectType.Sample)};
var newRecords = [];
if (records[0] == null) {
    alert("Please select at least one row")
} else {
    for (var n=0; n<records.length; n++) {
        var c = new sforce.SObject("Case");
        c.id = records[n];
        c.Status = "New";
        newRecords.push(c);
    }
    result = sforce.connection.update(newRecords);
    window.location.reload();
}
```

This example references the AJAX Toolkit, which is available if API access is enabled. See <https://developer.salesforce.com/page/Integration>. Notice the check for `records[0] == null`, which displays a message to users when they don't select at least one record in the list.

2. Add the button to your opportunity page layouts by editing the Cases related list.

Use this button on any page layout that contains the cases related list, such as account or contact page layouts.

See Also

[Custom Button and Link Samples](#)

Custom Link Example: International Maps

This example creates a link that displays a country-specific Google map.

REQUIRED EDITIONS

Available in: Salesforce Classic

Custom buttons and links are available in: **All** Editions

Visualforce pages and s-controls are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create or change custom buttons or links: [Customize Application](#)

1. Define a link for accounts with these attributes.

Display Type	Detail Page Link
Behavior	Display in new window
Content Source	URL

Use this sample code.

```
{!  
IF(Sample.BillingCountry = "US",  
"http://maps.google.com/maps?q=&Sample.BillingStreet&  
+"&Sample.BillingCity+"&"&Sample.BillingState+"&"&Sample.BillingCountry,  
(IF(Sample.BillingCountry = "UK",  
"http://maps.google.co.uk/maps?q=&Sample.BillingStreet  
&"&Sample.BillingCity+"&"&Sample.BillingCountry,  
"http://maps.google.com")))  
}
```

2. Add the link to your account page layout.

See Also

[Custom Button and Link Samples](#)

Custom Button and Link Considerations

Keep these considerations in mind when working with custom buttons and links.

REQUIRED EDITIONS

Available in: Salesforce Classic

Custom buttons and links are available in: **All** Editions

Visualforce pages and s-controls are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create or change custom buttons or links: Customize Application

Implementation Tips

- Custom buttons display at the top and bottom of the detail page to the right of all standard buttons.
- Custom buttons aren't distinguished from standard buttons in any graphical way. However, you can recognize them by their location on the right of all standard buttons.
- If the button bar gets too wide on the detail page layout, the browser displays a horizontal scroll bar. If the button bar gets too wide on the list view, search result, tagging result, or related list layouts, the buttons wrap.
- Custom buttons are available for activities under the individual setup links for tasks and events. To add a custom button to an activity list view or search layout, first create a custom list button in tasks or events. Next, add it to your activity list view or search result layouts. You can override a button that applies to both tasks and events.
- Person Account records use the custom buttons and links you have made for accounts.
- If your organization uses the Console tab, list buttons are available in Mass Action. List buttons don't display in the mini page layouts. Pages that display due to custom buttons and links display in the console without the header or sidebar.
- If you get an error message when overriding a button that appears in a list, try calling the s-control using the URLFOR function.
- When creating custom buttons, be aware of any validation rules your organization has for records on that object. For example, some custom list buttons that change case status conflict with a case validation rule. In this scenario, Salesforce displays the error message for the validation rule when users click the custom button.
- To replace a standard button with a custom button, first define the custom button, then customize the

page layout to hide the standard button and display the custom one in its place.

- Visualforce pages used as custom buttons or links on detail pages must specify a standard controller of the same object.
- Visualforce pages used as custom list buttons must use a standard list controller of the same object.
- A web tab or custom link could display a blank page if the embedded site:
 - Has been set to deny the loading of its content in a frame.
 - Has been set to allow the loading of its content in a frame only if the same site is delivering the content.
 - Contains a mix of secure and unsecure content, and the user's browser has been configured to block mixed active content.

To resolve this issue, try these workarounds.

- Set your custom link to either open in a new window or display in the existing window without the sidebar or header.
- Move the URL from a web tab into a custom link instead. Set the URL to either open in a new window or display in the existing window without the sidebar or header.
- If the site you're embedding has an HTTP prefix and mixed active content, try changing the prefix to HTTPS. If the embedded site has a valid security certificate and it hasn't blocked itself from being displayed in frames, using HTTPS as the prefix allows the site to display.

Best Practices

- Use formula functions in custom buttons with caution. Because functions run on the server before your HTML or JavaScript is passed to the browser, they can only evaluate information that exists at that time. Don't use functions like IF to evaluate conditions that only exist when the code reaches the browser, such as the value of a JavaScript variable that your code returns.
- Use relative or absolute URLs as the content source for custom buttons or links to ensure that they're rendered correctly.
- To prevent a user from performing a particular action, such as creating or editing, change the user's permissions rather than hiding the standard button. Hiding a standard button removes it from a page layout, but the link is still available and users can navigate to the new or edit page manually.
- Use global variables to access special merge fields for components like custom buttons, links, and s-controls. For example, the `$Request` global variable allows you to access query parameters inside a snippet, s-control, or custom button.
- When you create a custom list button, select **Display Checkboxes (for Multi-Record Selection)** only if your list button requires users to select individual records in a list. If your list button doesn't require users to select individual records, don't select this option. Don't select **Display Checkboxes (for Multi-Record Selection)** if your list button links to a URL that doesn't support POST operations, such as a URL that links to a Lightning component.
- In Lightning Experience, when you select **Display Checkboxes (for Multi-Record Selection)**, the related list type must be set to **Enhanced List**. You can set the related list type from the Related List-Single component or Related Lists component on a record page in the Lightning App Builder.
- If you create multiple custom list buttons on a list and select **Display Checkboxes (for Multi-Record Selection)** for at least one of the list buttons, checkboxes appear next to records in the list. But those checkboxes aren't activated for custom list buttons without **Display Checkboxes (for Multi-Record Selection)** selected.

Considerations for the Salesforce Mobile App

- Custom buttons that are added to the Button section of a page layout and that define the content source as *URL* or *Visualforce* are supported in the Salesforce mobile app. Remember that Visualforce pages must be enabled for use in the Salesforce mobile app.
Custom links, custom buttons that are added to list views, and custom buttons that define the content source as *OnClick JavaScript* aren't available in the Salesforce mobile app.
- Using custom URL buttons to pass parameters to standard pages in Salesforce Classic—such as prepopulating fields when creating a record—doesn't work in the Salesforce mobile app.
- Custom images used for action icons must be less than 1 MB in size.

See Also

[Custom Button and Link Limitations](#)

[Define Custom Buttons and Links](#)

[Custom Button and Link Samples](#)

Custom Button and Link Limitations

Keep these limitations in mind when working with custom buttons and links.

REQUIRED EDITIONS

Available in: Salesforce Classic

Custom buttons and links are available in: **All Editions**

Visualforce pages and s-controls are available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

- A custom link's label can't exceed 1,024 characters.
- A link URL can be up to 2,048 bytes. When data is substituted for the tokens in the URL, the link can exceed 3,000 bytes. Some browsers enforce limits on the maximum URL length.
- Custom buttons that call JavaScript aren't supported in Lightning Experience.
- Custom buttons with a content source of OnClick JavaScript aren't supported in the Salesforce app.
- Using URL custom buttons to pass parameters to standard pages in Salesforce Classic—such as prepopulating fields when creating a record—doesn't work in the Salesforce mobile app.
- On record detail pages for external objects that are associated with high-data-volume external data sources, custom buttons, and links that call JavaScript aren't supported.
- Custom buttons aren't available for Web-to-Lead, Web-to-Case, the Case Teams related list, or the user object.
- Custom buttons on search results pages aren't supported in Lightning Experience.
- When you redirect to an external website with merge fields, use the Text Field data type field. If you use the URL data type field, the link breaks because symbols in the link are converted to hexadecimal code.

- Some URL custom buttons open links in the same tab, even if the button behavior is set to open in a new window. For example, URL links to Visualforce pages always open in the same tab.
- Visualforce pages used as custom links on the home page can't specify a controller.

See Also

[Custom Button and Link Considerations](#)

[Define Custom Buttons and Links](#)

Viewing References to Salesforce Components

View a list of all the areas in Salesforce that reference a component. For example, view the custom links, custom buttons, or page layouts that reference another component, such as a Visualforce page or static resource.

REQUIRED EDITIONS

Available in: Salesforce Classic

Custom buttons and links are available in: **All** Editions

Visualforce pages and custom components available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create or change custom buttons or links: Customize Application

To create, edit, and delete Visualforce pages and custom components: Customize Application

To clone, edit, or delete static resources: Customize Application

To view a list of areas in Salesforce that reference a component, click **Where is this used?** from the component's detail page. Salesforce lists the type of component that references the component and the label for that component. Click any item in the list to view it directly.

See Also

[Define Custom Buttons and Links](#)

Manage Your MuleSoft Anypoint Platform Connections

Enable cross-platform features between Salesforce and Anypoint Platform by establishing a tenant relationship.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To connect your Salesforce org and the Anypoint Platform organization:

Salesforce Admin permission set

AND

Anypoint Platform organization key from the organization's administrator

Establishing a tenant relationship between your Salesforce organization and an Anypoint Platform organization enables Anypoint Platform users to:

- Enable the capability for API Catalog for Salesforce to sync, view, and activate invocable and agent actions for MuleSoft APIs in the connected Salesforce org.
- Enable the capability for Salesforce admins to discover and deploy pre-built MuleSoft Direct integrations.
- Enhance Anypoint Code Builder and Intelligent Document Processing (IDP) developer experiences with Einstein.

After you set up the tenant relationship, you can enable and disable your organization's features for Anypoint Platform users. You can also remove connections to the Anypoint Platform organization. Here are the ways to establish a tenant relationship.

- A Salesforce org is provisioned with the tenant relationship with an Anypoint Platform organization already established. In this case, the Anypoint Platform organization administrator must enable the connection in Anypoint Platform.
- The Anypoint Platform organization administrator initiates the tenant relationship in Anypoint Platform and the Salesforce org administrator completes the setup in the Salesforce org.

As a Salesforce administrator, you provide the Salesforce org tenant key to the Anypoint Platform administrator so that they can start the connection configuration process. After they initiate the connection process, you complete the connection configuration between your Salesforce org and the Anypoint Platform organization. You must have the Anypoint Platform organization key to complete the connection.

1. From Setup, in the Quick Find box, enter *MuleSoft*, and then select **MuleSoft**.
2. Select **Anypoint Platform Setup**.
3. In **Your connected Anypoint Platform Organization**, click **View Connection Info**.
4. Give your Salesforce org tenant key to your Anypoint Platform administrator and get the Anypoint Platform organization key. Each of you complete the steps to enter these in the respective

organizations you're connecting. Complete these steps.

- a. To get your Salesforce org tenant key for the connection, in **Connection to MuleSoft Anypoint Platform**, copy the tenant key and give it to your Anypoint Platform administrator. For information on how Anypoint Platform administrators initiate the connection, see [MuleSoft Documentation: Establish a Tenant Relationship with a Trusted Salesforce Organization](#).
 - b. To complete the connection after the Anypoint Platform administrator initiates it in Anypoint Platform, in **Connection to MuleSoft Anypoint Platform**, enter the Anypoint Platform organization key that you receive from the Anypoint Platform administrator **Anypoint Platform Organization ID**.
5. Click **Connect**.

The **Connection to MuleSoft Anypoint Platform** page shows the connected Anypoint Platform organization details, including the enablement status of the capabilities associated with the connected Salesforce organization.

[Manage Capabilities for MuleSoft in Salesforce](#)

After you establish a trusted connection between your Anypoint Platform and Salesforce organizations, you can enable and disable MuleSoft capabilities supported in the Salesforce organization.

[Remove a Connection Between Salesforce and Anypoint Platform](#)

Removing a connection revokes Anypoint Platform capabilities associated with your Salesforce org.

Manage Capabilities for MuleSoft in Salesforce

After you establish a trusted connection between your Anypoint Platform and Salesforce organizations, you can enable and disable MuleSoft capabilities supported in the Salesforce organization.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Salesforce admins can enable and disable MuleSoft capabilities in Salesforce and Anypoint Platform admins can enable and disable them in Anypoint Platform. This lets either administrator control the capabilities available from the UI they're responsible for. If the capabilities are disabled in either UI, the capabilities are not available to Salesforce users.

[Enable Einstein Entitlement and Routing in Anypoint Platform](#)

Let Anypoint Platform users use your entitlement and route Einstein prompts through your Salesforce org.

[Enable API Catalog Support for MuleSoft APIs](#)

Let Salesforce admins sync MuleSoft APIs into API Catalog and activate invocable actions.

[Enable MuleSoft Direct](#)

Let Salesforce admins discover and deploy pre-built MuleSoft Direct integrations directly from Salesforce.

Enable Einstein Entitlement and Routing in Anypoint Platform

Let Anypoint Platform users use your entitlement and route Einstein prompts through your Salesforce org.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To enable Einstein entitlement and routing in MuleSoft Anypoint Platform: MuleSoft Anypoint Platform permission set

1. From Setup, in the Quick Find box, enter *MuleSoft*, and then select **MuleSoft**.
2. Select **Anypoint Platform Setup**.
3. In the Your connected Anypoint Platform Organization section, click **View Connection Info**.
4. In the Connection to MuleSoft Anypoint Platform window, enable Enable Einstein in MuleSoft Anypoint Platform.

Enable API Catalog Support for MuleSoft APIs

Let Salesforce admins sync MuleSoft APIs into API Catalog and activate invocable actions.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To enable Einstein in MuleSoft Anypoint Platform: MuleSoft Anypoint Platform permission set

1. From Setup, in the Quick Find box, enter *MuleSoft*, and then select **MuleSoft**.
2. Select **Anypoint Platform Setup**.
3. In Your connected Anypoint Platform Organization, click **View Connection Info**.
4. If you aren't the Anypoint Platform admin, contact your Anypoint Platform admin to request to turn on MuleSoft API support for API Catalog. If you are the Anypoint Platform admin, in Connection to MuleSoft Anypoint Platform, click **Go to Anypoint Platform** and then follow the steps in [MuleSoft Documentation: Enabling API Catalog](#).

See Also

[API Catalog](#)

[MuleSoft Documentation: Establish a Tenant Relationship With a Trusted Salesforce Organization](#)

Enable MuleSoft Direct

Let Salesforce admins discover and deploy pre-built MuleSoft Direct integrations directly from Salesforce.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To enable MuleSoft Direct in MuleSoft Anypoint Platform:	MuleSoft Anypoint Platform root Organization Administrator permission and role
--	--

A MuleSoft Direct integration is a prebuilt app or a configurable example. Integrations simplify the business workflow implementation by allowing external systems to easily integrate with the Salesforce Cloud apps. An integration can depend on several other apps that are necessary to complete a business use case.

Enable MuleSoft Direct in Anypoint Platform

The Anypoint Platform organization administrator enables MuleSoft Direct in Anypoint Platform

1. Make sure your Anypoint Platform and Salesforce orgs are connected through a trusted tenant relationship. See [Manage Your MuleSoft Anypoint Platform Connections](#).
2. Sign in to Anypoint Platform using an account that has the root Organization Administrator permission.
3. In the navigation bar or the main Anypoint Platform page, click **Access Management**.
4. Select **Salesforce**.
5. In the section for the connected Salesforce organization to enable MuleSoft Direct for, select **Enable**.
6. Select **Continue**.
7. Select the business groups to enable MuleSoft Direct integrations for, and then select **Save settings**.

Access MuleSoft Direct in Salesforce Setup

After MuleSoft Direct is enabled in Anypoint Platform, the Salesforce organization administrator can view and deploy available integration assets to the Salesforce organization.

1. From Setup, in the Quick Find box, enter *MuleSoft*, and then select **MuleSoft**.
2. Select **MuleSoft Direct**.
3. Select **Get Started**.
4. Accept the terms and conditions, and select **Save**.

Deploy Integration Assets

When you deploy an integration asset, an application instance is created and shown in App instances for the integration asset.

For information about enabling and deploying assets from the Salesforce org, see [Enabling Integrations](#).

Delete an App Instance

Remove an app instance that you no longer need from your Salesforce org and Anypoint Runtime Manager.

1. From Setup, in the Quick Find box, enter *MuleSoft*, and then select **MuleSoft**.
2. Select **MuleSoft Direct**.
3. Select the integration asset to delete.
4. Click the down arrow, and then select **Delete Instance**.
5. Select **Delete**.

Remove a Connection Between Salesforce and Anypoint Platform

Removing a connection revokes Anypoint Platform capabilities associated with your Salesforce org.

1. From Setup, in the Quick Find box, enter *MuleSoft*, and then select **MuleSoft**.
2. Select **Anypoint Platform Setup**.
3. Click **View Connection Info**.
4. In Connection to MuleSoft Anypoint Platform, click the action menu and select **Remove Connection**.
5. Click **OK**.

The connection to Anypoint Platform is removed.

MuleSoft API Catalog for Salesforce

Bring your APIs into MuleSoft API Catalog for Salesforce, a single place to view your APIs from different sources and make them available to use in Salesforce. With just a few clicks, activate actions to use in your automations, such as flows, Apex code, and topic and agent actions.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions



-  **Example** Using API actions in a service agent that manages theme park reservations and experiences. Using credit service API actions in Salesforce flows.

Bring APIs into API Catalog

You can bring APIs into your API catalog from several sources.

View APIs in API Catalog

After you bring APIs into your API catalog, view and search the list views. Use your catalog as a central place to find APIs and actions to use in Salesforce.

Create and Manage Connections for Your APIs

Create connections to authorize the use of actions you activate for MuleSoft APIs and HTTP callouts in Salesforce. You create connections for other external API sources from outside of API Catalog, and then you can view the connections from the catalog.

Activate API Actions to Use in Salesforce

You can activate actions to use in your automations, such as flows, topic and agent actions, and Apex code.

Remove APIs from API Catalog

You can remove entire APIs from API Catalog for all supported sources. You can also remove specific MuleSoft API versions. Keep your catalog current to use the latest APIs, including their associated actions.

Examples of Using API Actions in Salesforce

After you sync your APIs in API Catalog and activate operations as actions, you can use them in Salesforce in flows, Apex code, and topic and agent actions.

Deploy API Catalog Data Using Change Sets

After you create, test, and debug your API Catalog configuration in a sandbox, use a change set to send the catalog and its associated dependencies, such as external services entries and named credentials, to production when you're ready to deploy.

Bring APIs into API Catalog

You can bring APIs into your API catalog from several sources.

Sources integrate with API Catalog in different ways based on their relationship with Salesforce and the features the API sources provide.

- API Catalog automatically brings in external services that you register by using the External Services page. To register an API from an external source not yet supported in API Catalog, see [Register an External Service](#).

Bring MuleSoft APIs into API Catalog

Connect your Anypoint Platform and Salesforce orgs, and enable the API Catalog capability. You can then sync MuleSoft APIs into your API catalog. API Catalog makes it easy to create secure connections for your MuleSoft APIs. You can stay up to date on new API versions as they sync into the catalog, and request API access when you need it.

Bring Apex APIs into API Catalog

Bring Apex REST and AuraEnabled APIs into API Catalog to take advantage of the automatic agent action activation and make your APIs more discoverable.

Bring Named Query APIs into API Catalog (Beta)

Named Query APIs are automatically shown in API Catalog when you create them in Salesforce. This makes them more discoverable by others and makes it easy for admins to activate agent actions for these APIs.

Bring Heroku APIs into API Catalog

After you publish your Heroku APIs into Salesforce, the Heroku APIs are automatically added to your org's API catalog. Heroku APIs shown in API Catalog are available as a Heroku action type that can be used to create a custom agent action.

Bring MuleSoft APIs into API Catalog

Connect your Anypoint Platform and Salesforce orgs, and enable the API Catalog capability. You can then sync MuleSoft APIs into your API catalog. API Catalog makes it easy to create secure connections for your MuleSoft APIs. You can stay up to date on new API versions as they sync into the catalog, and request API access when you need it.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

API specification information brought into API Catalog must meet schema and governance requirements.

- Schema definitions
- Schema update requirements
- API governance ruleset definitions (applied to APIs annotated for agent actions)

APIs brought into API Catalog must meet the External Services system limits.

Sync MuleSoft APIs into API Catalog

Sync eligible new and updated versions of MuleSoft APIs into your catalog.

Sync MuleSoft APIs into API Catalog

Sync eligible new and updated versions of MuleSoft APIs into your catalog.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

MuleSoft APIs that meet these criteria are eligible to sync into API Catalog.

- Are a REST API type
- Have an `sf-api-catalog` tag in Anypoint Exchange
- Have an additional `sf-api-topic` tag in Anypoint Exchange for APIs with agent topic and action annotations
- Have a consumer endpoint for at least one instance, where the consumer endpoint is a valid HTTPS URL

Consumer URLs appear as destination URLs in API Catalog.

See [MuleSoft Documentation: Add and Remove Asset Tags](#) and [MuleSoft Documentation: Add API Instances](#).

API Catalog makes it easy to create secure connections for your MuleSoft APIs. Monitor API versions as they sync into the catalog, request API access, and select SLA tiers directly from the catalog.



1. Ask an Anypoint Platform admin to complete the prerequisite configuration in the Anypoint Platform organization you want to sync APIs from. For details, see [MuleSoft Documentation: Establish a Tenant Relationship With a Trusted Salesforce Organization](#).
2. From Setup, in the Quick Find box, enter `API Catalog`, and then select **API Catalog**.
3. Click **Enable Sync**.
The sync typically takes several minutes to complete, especially for the initial sync, and runs as a background operation.
4. Make sure the latest synced APIs show in their respective list by refreshing the page.
If you don't see the APIs that you expect, double check the prerequisites that are required to sync into the catalog.

If you reach system limits during the sync, remove APIs or API versions that you no longer need. See [Remove APIs from API Catalog](#).

Stop and Restart the API Sync

Stop and restart the sync if necessary. Stop and restart to sync APIs or MCP servers that you just added to Anypoint Platform to avoid waiting for the next scheduled sync. Also consider stopping the sync before

creating API Catalog change sets for promotion to other orgs and restarting the sync after you're done.

1. From Setup, in the Quick Find box, enter *API Catalog*, and then select **API Catalog**.
2. To start the sync the first time or after you stop it, click **Enable Sync** on the API Catalog home page.
3. To stop and restart the sync at any time, click **Manage Sync** and enable or disable the sync by using the toggle.

See Also

[MuleSoft Documentation: Establish a Tenant Relationship With a Trusted Salesforce Organization](#)

[MuleSoft Documentation: Manage API Instances](#)

[System Limits](#)

[Schema Definitions](#)

[Schema Update Support](#)

[Manage External Services](#)

Bring Apex APIs into API Catalog

Bring Apex REST and AuraEnabled APIs into API Catalog to take advantage of the automatic agent action activation and make your APIs more discoverable.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

API information brought into API Catalog must meet schema and governance requirements.

- Schema definitions
- Schema update requirements
- API governance ruleset definitions (applied to APIs annotated for agent actions)

APIs brought into API Catalog must meet the system limits.

- External Services system limits
- Other system limits for Apex APIs

If you reach system limits during the Apex deployment, remove APIs that you no longer need.

Bring Apex REST APIs into API Catalog

If you have Apex REST code with custom agent actions and an LLM to create the OAS document for the REST code, you can bring the API into your API catalog.

Deploy an Apex REST class, OAS document, and associated metadata to use as an agent action. See [Agentforce Developer Guide: Build Custom Agent Actions using Apex REST \(Beta\)](#).

The deployment automatically adds the OAS API to your org's API catalog.

Bring Apex AuraEnabled APIs into API Catalog

If you have Apex classes with methods annotated with @AuraEnabled, you can bring the API into your API catalog.

Deploy an AuraEnabled controller method to use as an agent action. See [Agentforce Developer Guide: Build Custom Agent Actions from Apex Controller Methods](#).

The deployment automatically adds the OAS API to your org's API catalog.

System Limits for Apex Documents in API Catalog

When you deploy Apex classes, there are maximum limits for registrations, schema size, operations, objects, and properties. These limits affect what is displayed in API Catalog.

Limit	Value
Max Apex API registrations	150 per org
Max schema size (JSON)	10,000,000 characters (10.0 MB)
Max schema size (YAML)	3,000,000 characters (3.0 MB)
Max active operations	1250 per org
Max active and inactive operations	10,000 total per org
Max active objects	1250 per org
Max active and inactive objects	10,000 total per org
Max object properties	400,000 total per org

See Also

[Agentforce Developer Guide: Build Custom Agent Actions using Apex REST](#)

[Agentforce Developer Guide: Build Custom Agent Actions from Apex Controller Methods](#)

[System Limits](#)

[Schema Definitions](#)

[Schema Update Support](#)

[Manage External Services](#)

Bring Named Query APIs into API Catalog (Beta)

Named Query APIs are automatically shown in API Catalog when you create them in Salesforce. This makes them more discoverable by others and makes it easy for admins to activate agent actions for these APIs.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

 **Note** Support for Named Query APIs in API Catalog is a pilot or beta service that is subject to the Beta Services Terms at [Agreements - Salesforce.com](#) or a written Unified Pilot Agreement if executed by Customer, and applicable terms in the [Product Terms Directory](#). Use of this pilot or beta service is at the Customer's sole discretion.

API information brought into API Catalog must meet schema and governance requirements.

- Schema definitions
- Schema update requirements

APIs brought into API Catalog must meet the External Services system limits. If you reach system limits, remove APIs that you no longer need.

Save a Named Query API in Setup or deploy Named Query API metadata to Salesforce.

The Named Query API is automatically added to your API catalog.

See Also

[Developer Documentation: Create Actions from Named Query API \(Beta\)](#)

[System Limits](#)

[Schema Definitions](#)

[Schema Update Support](#)

[Manage External Services](#)

Bring Heroku APIs into API Catalog

After you publish your Heroku APIs into Salesforce, the Heroku APIs are automatically added to your org's API catalog. Heroku APIs shown in API Catalog are available as a Heroku action type that can be used to create a custom agent action.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

Publish Heroku APIs into Salesforce. See [Heroku Documentation: Heroku AppLink](#).

The Heroku publish brings the APIs into API Catalog and generates the External Service registration, named credential, external credential, and permission set. All operations available with the Heroku API are activated, and no user selection is required.

If you reach system limits during the publish, remove APIs that you no longer need. See [Remove APIs from API Catalog](#).

See Also

- [System Limits](#)
- [Schema Definitions](#)
- [Schema Update Support](#)
- [Manage External Services](#)

View APIs in API Catalog

After you bring APIs into your API catalog, view and search the list views. Use your catalog as a central place to find APIs and actions to use in Salesforce.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

You can view APIs from these sources in API Catalog:

- MuleSoft
- Apex REST
- Apex AuraEnabled
- Named Query API (Beta)
- Heroku
- External Services

1. In Salesforce, from Setup, in the Quick Find box, enter *API Catalog*, and then select **API Catalog**.
2. Click the Refresh icon to make sure that you're viewing the latest information.
3. Select the **All Sources** list view or a list view for a specific source.

View MuleSoft API Specifications by Version

After you select a MuleSoft API from the list of synced APIs, select the specific version of the API specification that you want to view.

API Catalog shows the latest available synced version of the MuleSoft API specification by default.

In a selected MuleSoft API in API Catalog, select a version.

The API details page shows information about the selected version.

See Also

- [Agentforce Developer Guide: Build Custom Agent Actions using Apex REST](#)
- [Agentforce Developer Guide: Build Custom Agent Actions from Apex Controller Methods](#)
- [Developer Documentation: Create Actions from Named Query API \(Beta\)](#)
- [Heroku Documentation: Heroku AppLink](#)

[Manage External Services](#)

Create and Manage Connections for Your APIs

Create connections to authorize the use of actions you activate for MuleSoft APIs and HTTP callouts in Salesforce. You create connections for other external API sources from outside of API Catalog, and then you can view the connections from the catalog.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

Create connections before you activate an action for your APIs. If a MuleSoft API doesn't have a connection yet, when you activate actions for the API, you're led through the create connection process.

-  **Important** After you create connections for MuleSoft APIs that have custom authentication protocols, complete the configuration and verify external credentials in Salesforce Security. Examples of these protocols are Basic, JSON Web Token (JWT), OAuth 2.0, and client ID through a third-party identity provider.

1. In Salesforce, from Setup, in the Quick Find box, enter *API Catalog*, and then select API Catalog.
2. Select the MuleSoft API that you want to activate actions for.
3. To create the connection and activate actions, click **Activate Actions**.
4. Select the API version.
5. To customize the default connection label and name, overtype them.
6. Enter a description.
7. Select a destination URL (API instance) to use as the managed named credential for this connection to Salesforce. If you can't access a destination that you select, ask your Anypoint Platform admin to approve the access and check the status again. Select an SLA tier if needed. See [MuleSoft Documentation: Client Applications, Contracts, and Credentials](#).
8. Click **Create and Continue**.

For APIs that use client ID enforcement through MuleSoft, the connection configuration is created automatically, including external credentials, named credentials, external services, and permission sets.

For APIs using external or custom authentication protocols, API Catalog creates the named credential and sets the connection's **Authentication Status** to Custom. After you complete the connection, verify your connection credentials and finish configuring them as needed. See [Create or Edit an External Credential](#).

The first connection that you create is automatically set as the default connection. To use a different connection in your automations, set it as the default connection.

You can't go back to update the connection immediately, but you can after you complete the activate action process. After you create the connection, you can update only the connection label and description. To use a different destination URL, create another connection for the API version using the new destination URL.

Make sure to assign user permissions for the connections that you create to avoid permission errors when you run the API actions. See [Assign User Permissions for the Connections](#).

Continue with activating actions for the API. See [Activate Invocable Actions for APIs](#).

Create Multiple Connections for a MuleSoft API

If a MuleSoft API has multiple destination URLs (API instances), you can create multiple connections for the API. However, you can have only one active connection at a time for each destination URL.

1. In Salesforce, from Setup, in the Quick Find box, enter *API Catalog*, and then select **API Catalog**.
2. Select the API that you want to create a connection for.
3. Click **New Connection**.
4. Select the API version.
5. To customize the default connection label and name, which default to the API name and version, overtype the label and name.
6. Enter a description.
7. Select a destination URL (API instance) to use as the managed named credential for this connection to Salesforce. If you can't access a destination URL that you select, ask your Anypoint Platform admin to approve the access and check the status again. Select an SLA tier if needed. See [MuleSoft Documentation: Client Applications, Contracts, and Credentials](#).
8. Click **Done**.

For APIs using client ID enforcement from MuleSoft, the connection configuration is created automatically, including external credentials, named credentials, external services, and permission sets.

For APIs using external or custom authentication protocols, API Catalog creates the named credential and sets the connection's **Authentication Status** to Custom. After you complete the connection, verify your connection credentials and finish configuring them as needed. See [Create or Edit an External Credential](#).

The first connection that you create is automatically set as the default. To use a different connection in your automations, set it as the default connection.

Set Default Connections for MuleSoft APIs

Set a connection as the default connection that's used by automations that run the associated MuleSoft API's actions. The first connection you create is automatically set as the default.

1. Select the **Connections** tab for a version of a MuleSoft API.

2. For a connection you want to set as the default, select the **Set as Default** action.

Create a Connection for an HTTP Callout

To create a connection to support an HTTP callout, create a connection without going through the activate action process.

1. In Salesforce, from Setup, in the Quick Find box, enter *API Catalog*, and then select **API Catalog**.
2. Select the API that you want to create a connection for.
3. Click **New Connection**.
4. Select the API version.
5. To customize the default connection label and name, which default to the API name and version, overtype the label and name.
6. Enter a description.
7. Select a destination URL to use as the managed named credential for this connection to Salesforce. API Catalog lists only active destination URLs, which are managed in Anypoint API Manager.
8. Click **Done**.
Connection configuration is created automatically, including external credentials, named credentials, external services, and permission sets.

Assign User Permissions for the Connections

After you create the connection, give specific Salesforce users access to the activated actions so that they can use them in their Salesforce automations. You're prompted to assign permissions for API connections as part of the create connections process. If you skip that step, you can assign the permissions later. To avoid permission errors, assign permissions before you try to use the associated API actions.

1. In Salesforce, from Setup, in the Quick Find box, enter *API Catalog*, and then select **API Catalog**.
2. Select the API to set connection permissions for.
3. If Permission Status is Not Assigned for that API or you want to update the permission assignment, click **Assign**.
The Permission Set page for the API opens.
4. Manage and add assignments for users according to your organization's standards. For details, see [Manage Permission Set Assignments](#).

Edit Connections

You can update the label or description for connections.

1. In Salesforce, from Setup, in the Quick Find box, enter *API Catalog*, and then select **API Catalog**.
2. Select the API version to modify.
3. On the **Connections** tab, click **Edit**.
4. Update the connection label or description.

To use a different destination URL, create another connection for the API version using the new destination URL.

Delete Connections

Delete unused, non-default connections.

1. In Salesforce, from Setup, in the Quick Find box, enter *API Catalog*, and then select **API Catalog**.
2. Select the API that has the connection to delete.
3. On the **Connections** tab, select a connection that's not set as the default and click **Delete**.
To delete a connection that's set as the default, set a different connection as the default and then delete the non-default connection. See [Set Default Connections for MuleSoft APIs](#).

See Also

- [Manage Permission Set Assignments](#)
- [System Limits](#)
- [Manage External Services](#)
- [MuleSoft Documentation: Managing API Instances](#)

Activate API Actions to Use in Salesforce

You can activate actions to use in your automations, such as flows, topic and agent actions, and Apex code.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

[Activate Invocable Actions for APIs](#)

Activate invocable actions for APIs. You can then use activated actions in your automations, such as those in flows, agents, and Apex code. You can activate actions by using API Catalog for MuleSoft, Heroku, and Apex REST APIs.

[Activate Agent Topic and Agent Actions](#)

In API Catalog, you can activate agent actions for APIs. If you activate invocable actions for an API that has annotations for agent actions, the associated agent actions also activate.

[Deactivate Actions](#)

Deactivate actions for APIs if you no longer need the actions.

Activate Invocable Actions for APIs

Activate invocable actions for APIs. You can then use activated actions in your automations, such as those in flows, agents, and Apex code. You can activate actions by using API Catalog for MuleSoft,

Heroku, and Apex REST APIs.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

For some sources, such as Apex and AuraEnabled, you can activate invocable actions through code during deployment of the APIs. This sets the invocable action to activated status in the catalog.

If you don't have a connection for a MuleSoft API, the activate actions process begins the create connection process. See [Create and Manage Connections for Your APIs](#).

1. From Setup, in the Quick Find box, enter *API Catalog*, and then select **API Catalog**.
2. Select the API that you want to activate actions for. For MuleSoft APIs, select the API version from **Version**.
3. View resource names for operations. API Catalog generates the operations from the API schema.
4. Expand the rows to view operation names and descriptions.
5. Click **Activate Actions**.
6. Select operations to activate as actions, and create connections for them if prompted to do so.

Make sure to assign user permissions for the connections that you create. See [Assign User Permissions for the Connections](#).

See Also

[Heroku Documentation: Heroku AppLink](#)

[Agentforce Developer Guide: Build Custom Agent Actions using Apex REST](#)

[Agentforce Developer Guide: Build Custom Agent Actions from Apex Controller Methods](#)

[System Limits](#)

Activate Agent Topic and Agent Actions

In API Catalog, you can activate agent actions for APIs. If you activate invocable actions for an API that has annotations for agent actions, the associated agent actions also activate.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

API developers add annotations to API operations to define topics and actions for Salesforce agents through their Integrated Development Environments (IDEs). Use features in Anypoint Code Builder for MuleSoft APIs and Salesforce Code Builder for Apex APIs to run API governance rules against the APIs to make sure that the API specs meet security, documentation, and other requirements and standards

defined for autonomous API operations.

API permissions and policies and user permission sets control the access and options users have when they design, administer, and use the APIs.

! **Important** If APIs have topic and agent annotations, to be eligible to sync into API Catalog, the APIs have a tag of `sf-api-topic` and meet other sync criteria.

1. From Setup, in the Quick Find box, enter *API Catalog*, and then select **API Catalog**.
2. View the Agent Action column to see which APIs don't have Activated status.
3. Select an API to activate actions for, and then select the API version.
4. View resource names for operations that are generated from the API schema.
5. Expand the rows to view operation names and descriptions.
6. Click **Activate Actions**.
7. Select operations to activate as actions, and create connections for them if prompted to do so.
The invocable actions and their associated agent actions are activated.

After you create your action, assign it to a topic to add it to your agent. See [Add an Action to a Topic](#).

See Also

[Create a Custom Agent Action](#)

[Agentforce Developer Guide: Build Custom Agent Actions using Apex REST](#)

[MuleSoft Documentation: Enabling an API Project for Topics and Agent Actions](#)

Deactivate Actions

Deactivate actions for APIs if you no longer need the actions.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

1. To make invocable actions eligible to deactivate for Apex REST, Apex AuraEnabled, and Heroku APIs, remove the associated agent actions.
2. From Setup, in the Quick Find box, enter *API Catalog*, and then select API Catalog.
3. Select the API that you want to deactivate actions for, and then select the API version.
4. Click **Activate Actions**.
5. In the Select Operations to Activate as Actions page in the **Operations** tab, deselect the operations for actions that you want to deactivate.
6. Click **Update**.
The invocable action deactivates.

To deactivate an agent action for a MuleSoft API, remove the API or API version. See [Remove APIs from API Catalog](#).

Remove APIs from API Catalog

You can remove entire APIs from API Catalog for all supported sources. You can also remove specific MuleSoft API versions. Keep your catalog current to use the latest APIs, including their associated actions.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

1. To make sure that an API or API version is eligible to remove, follow these steps.
 - a. Remove its actions that are part of Salesforce automations.
 - b. Deactivate its actions in API Catalog.
2. Before you remove a MuleSoft API, remove the tags in Anypoint Exchange that make the APIs eligible to sync. The tags are typically `sf-api-catalog` and `sf-api-topic`. If you don't remove the tags, API Catalog adds the APIs or versions again in the next sync.
3. From Setup, in the Quick Find box, enter `API Catalog`, and then select **API Catalog**.
4. To remove an entire API, in the API list, for the API to remove, select the **Remove API** action. Removing an External Services API removes the API from API Catalog and from the External Services page.
5. To remove a MuleSoft API version, complete these steps.
 - a. In the API list, select the API to remove.
 - b. Select the version of the API.
 - c. In the API version details page, click **Remove API**.

Examples of Using API Actions in Salesforce

After you sync your APIs in API Catalog and activate operations as actions, you can use them in Salesforce in flows, Apex code, and topic and agent actions.

See the examples for ideas on how to use actions to customize your Salesforce org with clicks.

Example of Using MuleSoft for Agentforce

Consider a scenario where a fictitious company called Enchanted Entertainment wants to create a service agent to manage theme park reservations and experiences. They've developed APIs using MuleSoft Anypoint Platform to integrate with external systems. Now, they need to connect these APIs to their service agent. Here's how API Catalog can help with that.

Example of Using Actions in Flow

Consider a situation where you want to connect Salesforce to a credit service that has an API that's managed in MuleSoft Anypoint Platform. The credit service retrieves the customer's credit score, determines whether the customer qualifies for the credit service, and provides the payment terms. You want to extend the credit service to a Salesforce account so that you can view the payment terms in Salesforce.

Example of Using MuleSoft for Agentforce

Consider a scenario where a fictitious company called Enchanted Entertainment wants to create a service agent to manage theme park reservations and experiences. They've developed APIs using MuleSoft Anypoint Platform to integrate with external systems. Now, they need to connect these APIs to their service agent. Here's how API Catalog can help with that.

Enchanted Entertainment can bring the theme park reservation API into API Catalog, activate operations as actions, and then use these actions in topic and agent actions. The service agent can then seamlessly access and manage reservation data, ensuring a smooth and efficient customer experience.

Watch the video to see MuleSoft for Agentforce: API Catalog in action.

Watch the video: <https://play.vidyard.com/NpRpzLga4tM8R3vgrYRr3F>

Example of Using Actions in Flow

Consider a situation where you want to connect Salesforce to a credit service that has an API that's managed in MuleSoft Anypoint Platform. The credit service retrieves the customer's credit score, determines whether the customer qualifies for the credit service, and provides the payment terms. You want to extend the credit service to a Salesforce account so that you can view the payment terms in Salesforce.

You can bring the credit service API into API Catalog, activate operations as actions, and then use the actions in a Salesforce flow.

Here are more detailed steps for the example.

After you sync the credit service API to API Catalog in Salesforce:

- Find and select the credit service API in API Catalog in Salesforce.
- Select the operations that you want to make available in Salesforce as invocable actions, and create connections for them.
The create connection process automatically generates the credentials for invoking the API actions from within Salesforce automations.
- Assign user permissions to the permission set created for the API's connection.

After you activate the invocable actions from API Catalog:

- Create a flow by using the new credit service API category in Flow Builder.
After the flow runs, the output contains the credit decision and, if applicable, payment terms.

See Also

[End-to-end Example with Flow](#)

Deploy API Catalog Data Using Change Sets

After you create, test, and debug your API Catalog configuration in a sandbox, use a change set to send the catalog and its associated dependencies, such as external services entries and named credentials, to production when you're ready to deploy.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

User Permissions Needed	
To edit deployment connections:	Deploy Change Sets AND Modify Metadata Through Metadata API Functions
To use outbound change sets:	Create and Upload Change Sets
To use inbound change sets:	Deploy Change Sets AND Modify Metadata Through Metadata API Functions

! **Important** You can't use managed packages to promote API Catalog data from org to org. The authentication data necessary for API Catalog can't be retained and installed in a separate org, and you can't edit the authentication information installed through a package in the target org.

1. Create an outbound change set in the source org for the API Catalog entries you want to deploy to another org. See [Outbound Change Sets](#).
2. Add **Cataloged API** to the change set and then select the catalog entries to include in the change set. The dependencies are automatically selected for you.
3. Upload your outbound change set.
4. Deploy your inbound change set in your target org. See [Inbound Change Sets](#).
5. Make sure MuleSoft API connections that are in Custom status have valid external credentials, because API Catalog can't automatically verify or create those during the connection process. See [Create and Manage Connections for Your APIs](#).

Deploy Change Sets Using Salesforce CLI

You can alternatively deploy change sets using Salesforce CLI commands.

1. Create an empty Salesforce DX project. See [Salesforce DX Developer Guide: Create a Salesforce DX Project](#).
2. Create a package XML file that has these metadata API components as shown in the example.
 - CatalogedApi
 - CatalogedApiVersion
 - CatalogedApiArtifactVersionInfo
 - ExternalServiceRegistration
 - NamedCredential
 - ExternalCredential
 - PermissionSet
 - Flow

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>PermissionSet</name>
    </types>
    <types>
        <members>*</members>
        <name>ExternalCredential</name>
    </types>
    <types>
        <members>*</members>
        <name>NamedCredential</name>
    </types>
    <types>
        <members>*</members>
        <name>ExternalServiceRegistration</name>
    </types>
    <types>
        <members>*</members>
        <name>CatalogedApiArtifactVersionInfo</name>
    </types>
    <types>
        <members>*</members>
        <name>CatalogedApi</name>
    </types>
    <types>
        <members>*</members>
        <name>CatalogedApiVersion</name>
    </types>
```

```
</types>
<types>
  <members>*</members>
  <name>Flow</name>
</types>
<version>65.0</version>
</Package>
```

3. Pull the package into your local project to create a change set by using the `sf project retrieve start` command.

This example creates a change set in a source org with the org ID `mySandboxOrg`.

```
sf project retrieve start --manifest package.xml --api-version 65.0 --target-org mySandboxOrg
```

4. Deploy to the target org by using the `sf project deploy start` command.

This example deploys the change set to a target org with the org ID `myProdOrg`.

```
sf project deploy start -o myProdOrg
```

See Also

[Change Sets](#)

[Developer Documentation: Metadata API Developer Guide](#)

[Developer Documentation: Salesforce CLI Command Reference](#)

External Services

Connect your Salesforce org to an external API by using zero lines of code. Use declarative tools and OpenAPI specifications to describe the external API functionality, and External Services automatically creates invocable actions within Salesforce. Use External Services for outbound integrations from Salesforce by using low-code, process-based integrations or to turbocharge your Apex integrations. Call the invocable actions natively from Apex, or create a custom agent action, flow, or Einstein bot that interacts with the external API source.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

With External Services, you first register OpenAPI 2.0 or OpenAPI 3.0 schemas. The operations imported from your registered schema automatically become invocable in Apex, or as `External Services` action types within point-and-click automation tools such as Flow Builder, Orchestrator, Einstein bots, or OmniStudio Assets.

External Services is best used when the externally hosted service is a RESTful service and the API specification is available in OpenAPI 2.0 or OpenAPI 3.0 JSON or YAML schema format.

Introduction Video

Watch the Introduction video to see how you can declaratively transform API specifications into invocable actions.

Watch the video: <https://play.vidyard.com/UNmoeSMfkdFy7LnWKunNd5>

Example Use Cases

The examples in this section demonstrate typical External Services workflows.

External Services Considerations

Here are the semantic, service, and usage constraints to keep in mind when integrating your services into External Services.

Register an External Service

Provide an API spec that describes your endpoint's services and methods. The API spec's schema generates the external service operations with corresponding input and output parameters. You can also edit an existing registration, register an external service with a Mulesoft API, or register an external service using Flow Builder's HTTP Callout functionality.

Use Agentforce to Invoke External Service Actions

Add capabilities to agents with customizable actions configured via External Services. Use External Services to declaratively connect your Salesforce org to an external API. After you register an OpenAPI schema, the actions defined in that schema become automatically available as custom agent actions.

Use Flow to Invoke External Service Actions

In this Flow example, design and test the automation that sends a user's information from Salesforce to the external employee banking system. Create the variables for user phone numbers, and combine different phone numbers into one piece of data. Then use the external service action to create the user.

Invoke External Service Callouts Using Apex

You can call external service registrations natively from Apex. Make a callout to an external service like the Apex `Http Class` without the need to write boilerplate code. The registered services are strongly typed in Apex with the registration's schema as Apex types. These Apex types reflect your registered service's specification, making the Apex compiler do the heavy lifting for you.

Add External Service Actions to an Einstein Bot

Integrating your Einstein bot with a registered external service is now as easy as adding an action to a dialog. You can add an external service action to your bot from Einstein's Bot Builder.

Invoke External Services from Omnistudio Assets

You can invoke External Services' registered actions from Omnistudio Integration Procedures in OmniScripts and FlexCards.

Install or Create Packages

You can install or create packages that contain an external service. Read these tips before you begin.

Testing External Services

Test the example flow `MyAtmFlow` with a flow action calling the external service `MyAtmExternalService` by implementing the `HttpCalloutMock` interface. The interface provides a test response without performing the callout to the external service endpoint. The sample Apex unit test performs a mock test against the same HTTP callout mock implementation.

Schema Examples

Explore various scenarios with OpenAPI 2.0 and OpenAPI 3.0 compliant JSON or YAML schemas supported by External Services. The examples cover schema elements like HTTP header as input parameters, and include example usage in Apex. Understanding the examples helps with proper syntax and code placement.

Example Use Cases

The examples in this section demonstrate typical External Services workflows.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

Connect to a credit service.

To illustrate how External Services work, let's say you want to connect Salesforce to an external credit service. The credit service determines if credit can be extended to a Salesforce account. You also want to know the payment terms.

First, you create a Named Credential by supplying a URL and authentication settings for the credit service. Salesforce uses these items to make callouts to the external service. With External Services, you then select whether you get your API spec from Mulesoft Anypoint Platform, or from an API spec provided elsewhere. To register an API spec:

- Enter a URL (or endpoint) to the location of the hosted OpenAPI specification ("API spec") for the external service, or provide (paste in) the complete schema from the API spec. For Mulesoft, provide your credentials and select the API spec.
- Select the operations from the schema you want to make available in Salesforce as invocable actions, and complete the registration.

External Services walks you through it so that you don't directly touch the API. You can also skip writing Apex code to make the callout.

After you've registered your new external service, use a Salesforce point-and-click automation tool, such as Flow Builder or Einstein Bots. Create a flow using the External Service type for flow actions automatically generated from your External Services registration. When the flow runs, the output contains the credit decision and, if applicable, payment terms.

Automatically add a new org user to external applications.

Another use case involves saving time when you add new users to an org. For instance, you want new users to automatically become collaborators for all external org-related applications. Create a flow using inputs such as user ID, which you can define in your API spec's schema. Then you add triggers in your flow. When a user is created, the triggers engage and add the new user as a collaborator to your other applications.

Send alerts to an external application with Apex.

If your business logic requires more tailored processing needs, you can easily extend the reach of your Apex logic to external integrations conforming to OpenAPI specifications. You can extend Apex with synchronous or asynchronous calls to a registered external service. In one use case, a detailed alert is sent to a third-party application if an urgent case is opened.

External Services Considerations

Here are the semantic, service, and usage constraints to keep in mind when integrating your services into External Services.

[OpenAPI 2.0 and 3.0 Support](#)

Learn about valid API schema components that can be used to register a spec successfully.

[Authentication](#)

To authenticate, External Services uses defined authentication parameters in Named Credentials.

[Schema Definition Support](#)

Learn about External Services' schema limits and other considerations.

[Schema Update Support](#)

You can register an updated schema version for one currently in use in flow or Apex that includes *supported* components. This section provides details about whether changes are supported.

See Also

[External Services OpenAPI 2.0 Schema](#)

[External Services OpenAPI 3.0 Schema](#)

[Using the Schema Examples](#)

OpenAPI 2.0 and 3.0 Support

Learn about valid API schema components that can be used to register a spec successfully.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

Along with normal packaging support, you can use:

- Nested parameters—rendered as Dynamic Apex-defined data types in Apex and Flow Builder.
- Named arrays, anonymous arrays, or inline arrays, supported as generic `List` type in Apex, or as `collection` type in Flow Builder.
- OpenAPI `additionalProperties` as generic `Map` type in Apex.
- Header parameters in HTTP requests.
- The following media types are supported:
 - `application/json` -- Structured data in JSON format
 - `application/x-www-form-urlencoded` -- URL encoded form
 - `text/plain` -- Unstructured data as plain text
- Non-supported media types can be mapped to supported media types for request and response content serialization. See [Media Type Mapping in External Service Registrations](#).
- Form data parameters are supported.
- Supports all Java runtime supported character set encodings. UTF-8 is the default encoding.
- OpenAPI 2.0: Server and operation level `consumes` and `produces` directives.
- OpenAPI 3.0: Content media type with supported media types.

Supported Schema Format

- OpenAPI 2.0, JSON schema format
- OpenAPI 3.0, JSON schema format
- OpenAPI 2.0 schema, YAML format
- OpenAPI 3.0 schema, YAML format
- UTF-8, with full character set for names and identifiers

Unsupported Schema Formats

- Interagent hyper-schema format

Unsupported Schema Constructs

- OpenAPI 3.0: The composite schema keyword: `not`
- Media types such as xml, png, and pdf

Authentication

To authenticate, External Services uses defined authentication parameters in Named Credentials.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

External Services *doesn't* respect:

- A schema's `securityDefinitions` and `security` sections.
- Authentication data stored in a flow. The standard HTTP request headers `Accept`, `Content-Type`, and `Authorization` are handled by the External Service and can't be overridden as OpenAPI operation input parameters. The Authorization header is set by the Named Credential.

To create a Named Credential, see [Define an External Credential and a Named Credential](#).

Schema Definition Support

Learn about External Services' schema limits and other considerations.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

System Limits

External Services maximum system limits for registrations, schema size, operations, objects, properties, and file size.

[Callouts and Callbacks: Limits and Usage](#)

Limits and best approaches for callbacks and callouts to and from External Services.

[Schema Definitions](#)

Learn the basics about External Service's schema support, schema components that are ignored, and supported data types.

[Character Set Encodings](#)

Learn about request and response character set encodings.

[Media Type Directives and Mapping](#)

Learn about OpenApi 2.0 `consumes` / `produces` or OpenAPI 3.0 `content` media type schema directives.

[Naming and Description Conventions](#)

Learn about support for component names and descriptions defined in your schema.

[Defaults and Default Values](#)

Learn about External Services' support for defaults. For OpenAPI 2.0, default values set on `properties` are used. For OpenAPI 3.0 schemas, the "outermost" default is used. Non-supported configurations are ignored.

[Operation Output Parameters](#)

Learn about how External Services interprets an operation's output parameters.

[Structure Data Using `properties` and `additionalProperties`](#)

In an Open API specification, the properties keyword defines one or more properties on an object including the property name and data type. The additionalProperties keyword gives you the flexibility to add additional properties with an undefined schema. The named properties are accessible as Apex properties with a matching data type. The additionalProperties in the spec are accessible in Apex as a Map property.

Apex Class Names and Developer Names

Learn how External Services derives Apex class names from your schema.

Apex Reserved Keywords

Learn about External Service's support for Apex reserved keywords.

Null Values

Learn null values are interpreted by External Services.

System Limits

External Services maximum system limits for registrations, schema size, operations, objects, properties, and file size.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: Enterprise, Performance, Unlimited, and Developer Editions

Limit	Value
Max External Service registrations	700 per org
Max schema size (JSON)	10,000,000 characters (10.0 MB)
Max schema size (YAML)	3,000,000 characters (3.0 MB)
Max active operations	3,000 per org
Max active and inactive operations	10,000 total per org
Max active objects	3,000 per org
Max active and inactive objects	10,000 total per org
Max object properties	400,000 total per org
Max file size for upload or download	16 MB

What Are Operations and Objects, Active and Inactive?

An **operation** is a schema construct that defines a singular API capability in terms of an endpoint path and HTTP method, such as GET, PUT, or DELETE. A path can have several methods, each representing an operation. In the example below, the GET capability on the `/accounts` resource is displayed as the

`getAccount` operation (2). `getAccount` represents one operation.

An **active operation** is an operation that you select to include when you register or edit your schema (1). External Services transforms active operations into invocable actions in Salesforce. **Inactive operations** are the ones that weren't selected during registration (2).

Operations	Description	Input parameters	Output parameters
<input checked="" type="checkbox"/> addAccount	Add account	name, accountName	default, 409, 201, responseCode, 400, 404
<input type="checkbox"/> DeleteAccount	Delete account	accountName	204, 404, default, 400, responseCode
<input type="checkbox"/> getAccount	Retrieve account	name	default, 200, responseCode, 404, 400
<input type="checkbox"/> updateAccount	Update account	name, accountType	400, default, 404, 200, responseCode

An **object** is any complex data structure that isn't a simple data type, such as string, integer, date, date time, time, or float.

An **active object** is an object that's at least referenced by one active operation in the schema, or if its parent object is active. An **inactive object** isn't referenced by an active operation or an active parent object.

Objects can be defined as a named resource in the schema and can be referenced by operations and parent objects for object schema reuse. In this way, multiple operations can use the same object. If one operation is active, but another is inactive, and they use the same object, the object is considered active. An inline object is private to an operation or parent object.

Inactive operations and objects are counted against system limits. Even though they aren't transformed into invocable actions and dynamic Apex classes, inactive operations and objects consume a certain amount of system resources.

To see counters that track the sum of active and inactive operations and objects in your org, go to the External Services home page in Setup.

Considerations for File Upload and Download

These are the supported binary file types.

- Adobe Acrobat (.pdf)
- Archive files with an extension of .zip or .gzip
- Audio files with an extension of .mpeg or .wav
- Comma-separated values (.csv)
- Image files with an extension of .jpeg, .png, .gif, .bmp, .tiff, or .svg

- Microsoft Excel files with an extension of .xls, .xlt, .xla, or .xlsm
- Microsoft PowerPoint files with an extension of .ppt, .pot, .pps, .ppa, or .pptx
- Microsoft Word files with an extension of .doc, .dot, or .docx
- Rich text
- Text
- Video files with an extension of .mp4, .avi, .wmv, or .mov

See Also

[External Services Considerations](#)

Callouts and Callbacks: Limits and Usage

Limits and best approaches for callbacks and callouts to and from External Services.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

- External service callouts aren't allowed when there are pending, uncommitted transactions.
 - In Flow, close your transaction, and resume Flow using the flow element Pause.
 - In Apex, perform the callout before any DML statement such as inserting, updating, or deleting objects. See [Callout Limits and Limitations](#) in *Apex Developer Guide*.
- Callouts to External Services time out after 120 seconds.

External Services Asynchronous Callbacks

 **Note** External Services' asynchronous callback operations are supported in Apex but not in Flow.

The timeout duration of a callback operation is configured in the Apex client, and can be set for a maximum of twenty-four hours.

Accepted callback HTTP methods are PUT and POST. All other methods are ignored.

One callback parameter must contain only one callback operation, as a one-to-many relationship isn't supported. If one-to-many is detected, only the first callback operation under the callback parameter is processed.

In [Example API Specification With Callback Operation](#), the first callback parameter is `applicationOutcomeApproved`. It contains one callback operation `#ref/.../ApplicationApproved`. The third callback parameter is `applicationError`. It contains one callback operation `"{$request.query.callbackUrlForErrorCases}": {"post": "....."}" 3.3`. We ignore the callback operation response declaration and always send 200, 404, 408 (timeout), or 500 HTTP response to the callback sender.

A callback URL can be defined as a query parameter or added to the request body parameter. A callback URL can be:

- a string type
- a property of an object type
- an element of an array

Some valid callback URL expression formats are:

- ' `${request.query.callbackUrl} '`
- ' `${request.body#/callbackUrl} '`
- ' `${request.body#/callbackUrl/1} '`

Other complex expressions aren't currently supported. For example, expressions with static text combined with variable references, or expressions that have multiple variable references. In these unsupported cases, the callback is considered invalid and the operation is attempted as a synchronous callout with a 120-second time out.

A callback operation can define an `operationId`. For example, in the [Example API Specification With Callback Operation](#), see:

```
"operationId": "approvedCallback"
```

This `operationId` can't be duplicated with any other `operationId` - with a regular top-level operation

```
"operationId": "SubmitApplication"
```

or any other callback `operationId`.

If a callback operation doesn't declare an `operationId`, Salesforce uses its callback component reference name as the `operationId` - in this case - `ApplicationRejected`.

If the callback operation is declared inline, then its `operationId` is constructed as

```
parentAsyncOperationId_CB_callbackParameterName
```

For example, `SubmitApplication_CB_applicationError`.

Schema Definitions

Learn the basics about External Service's schema support, schema components that are ignored, and supported data types.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

When you create your API spec, keep the following in mind.

- You can use the `GET`, `PATCH`, `PUT`, `POST`, and `DELETE` methods in a schema. A `GET` method with an empty request body results in a validation error.
- A property must include a value.
- Each parameter must have a name.
- Request headers are supported.
- Response headers aren't supported.
- Form parameters are supported.
- Security settings defined in the API spec are ignored and defer to security settings in the Named Credential.

The following OpenAPI schema components are ignored:

- Security requirement objects and security definitions
- Tag objects
- External documentation objects
- For `allOf`, `oneOf`, and `anyOf`, the schema object property `discriminator` is supported. The `discriminator/mapping` in OpenAPI 3.0 is ignored. The `discriminator` property determines the schema referenced by its type name. For more information, see the Swagger OpenAPI 3.0 specification [Inheritance and Polymorphism](#).

Supported data types:

- `binary`
- `boolean`
- `date`
- `datetime`
- `double`
- `float`
- `integer`
- `long`
- `string`
- any type (as Apex Object)
- `object`: top level named and nested anonymous objects
- anonymous and top-level named lists (can nest both named and anonymous arrays)
- `additionalProperties` as maps
- `allOf` as an object composition
- OpenAPI 3.0 only:
 - `anyOf` (any of primitive, list, or object schema types; only object schemas can constitute composition types)
 - `oneOf` (one of primitive, list, or object schema types)

See Also

Schema Examples

Character Set Encodings

Learn about request and response character set encodings.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

The request body is encoded by the character set in the operation's resolved request schema directive.

For example, `application/json; charset=ISO-8859-1` encodes the HTTP request in Latin-1 before sending it to the server.

The response body is encoded by the character set in the operation's resolved response schema directive. For example, `application/x-www-form-urlencoded; charset=SHIFT_JIS` decodes the percent URL encoded form with the Shift JIS character set.

The default character set for encoding request bodies is UTF-8.

The default character set for decoding server response entities is defined by the server response's Content-Type header. If the Content-Type header is missing from the response, then it uses UTF-8.

Reserved HTTP request header characters aren't escaped and are encoded by the request body's character set.

The supported encoding set is as defined by the Java JDK. For example, for version 11, supported encoding is documented here (replace the "11" in this URL with your Java SE Platform version number): <https://docs.oracle.com/en/java/javase/11/intl/supported-encodings.html> - and it's recommended to use the standard character set encoding name as described by the [IANA Charset Registry](#).

See Also

Schema Examples

Media Type Directives and Mapping

Learn about OpenAPI 2.0 `consumes / produces` or OpenAPI 3.0 `content` media type schema directives.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

Unsupported schema directives invalidate the API spec registration.

Schema directives incompatible with a request body or response entity schema cause errors during callout.

You can register specifications with nonsupported or incompatible `consumes`, `produces`, or OpenAPI 3.0 content media type directives by mapping the media types to supported media types. For more information about media type mapping, see [What is Media Type Mapping?](#).

Missing schema directives are defaulted to:

- `application/json` –if the request body or the response body schema is either an object or an array
- `text/plain` –if the request or response schema body is a primitive type such as a string or an integer number

If no request body parameter is defined for methods POST, PUT, and PATCH, form data request parameters are sent in the request body as `application/x-www-form-urlencoded`.

The applicable media type in the OpenAPI 2.0 `produces` list, OpenAPI 3.0 `response` schema media types, or a default (`application/json`, or `text/plain`), is set as the HTTP Accept header.

What is Media Type Mapping?

You can map each nonsupported, custom media type instance (defined in your API spec) to a supported media type *of the same format*. Supported media types for both requests and responses are:

- `application/json` –Structured data in JSON format.
- `text/plain` –Unstructured data as plain text.
- `application/x-www-form-urlencoded` –URL encoded form.

For example, if your spec includes a customized media type `application/abc.api+json` and the formatting adheres to standard JSON formatting, then map this custom media type to `application/json`.

You can perform this mapping by either [Mapping Media Types During Registration](#) (the declarative, easy way) or by [Mapping Media Types with Metadata API](#) (which involves manual coding).

Custom media types with a supported media type suffix are recognized as one of the compatible supported media types. For example, a custom or vendor media type `application/vnd.api+json` with media type suffix json is recognized as `application/json`.

Supported media type suffixes:

- `json` –Structured data in JSON format.

For information about OpenAPI 2.0 `consumes` / `produces` or OpenAPI 3.0 `content` media type schema directives, see [Schema Definition Support](#).

Naming and Description Conventions

Learn about support for component names and descriptions defined in your schema.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

The operation name is derived from the schema's path `operationId`. If missing, it's derived from the HTTP method and resource path.

Implement case consistently for object name definitions, property names, parameter names, and field names in a schema.

The derived Apex class name and Apex object property names require valid Apex identifier characters. Ensure that the object name in the schema object definition section matches the naming convention and the object property names.

External Services can handle up to 1,333 characters for any optional description string used to describe operations and parameters in your JSON or YAML schema definition. If the description string exceeds this limit, External Services stores only the first 1,333 characters of the description. Truncating the operation or parameter description doesn't affect the integrity of your schema definition.

See Also

[Schema Examples](#)

[Apex Class Names and Developer Names](#)

Defaults and Default Values

Learn about External Services' support for defaults. For OpenAPI 2.0, default values set on `properties` are used. For OpenAPI 3.0 schemas, the "outermost" default is used. Non-supported configurations are ignored.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

Default Values

For both OpenAPI 2.0 and 3.0 specifications,

- The type of the default value must match the type of the property/parameter. For example, if the type

- is `integer` and the default value provided is `50.2`, the default value doesn't translate to Apex.
- For `integer` defaults to show up, a format must be specified, for example, `int32` or `int64`.

The following sections describe Open API version-specific support for defaults.

OpenAPI 2.0 Default Definitions

For OpenAPI 2.0 schemas, External Services supports a subset of primitive data types for default values set on object properties. These primitive data types are limited to:

- `string`
- `number`
- `integer`
- `boolean`

For OpenAPI 2.0 schemas, only the default values set on `properties` are used.

For example, in the code snippet below, the default set for the `month` property is supported. In the Apex class, you see a comment in the `birthday` object with

```
// Default value: January
```

However, the default value that's set at the object level

```
"default": { "month": "December" }
```

is ignored.

```
"birthday": {
    "type": "object",
    "properties": {
        "month": {
            "type": "string",
            "default": "January"
        },
        "day": {
            "type": "integer"
        },
        "year": {
            "type": "integer"
        }
    },
    "default": {
        "month": "December"
    }
}
```

Default values set anywhere else other than on the property (such as `object/composed object`) are ignored.

In the following example, the default value is 123 because that's what is set on the property.

```
"Fax": {
    "allof": [
        {
            "$ref": "#/components/schemas/Contact"
        },
        {
            "type": "object",
            "properties": {
                "faxNumber": {
                    "type": "integer",
                    "default": 123
                }
            }
        },
        {
            "default": {
                "faxNumber": 456
            }
        }
    ],
    "default": {
        "faxNumber": 789
    }
}
```

OpenAPI 3.0 Default Definitions

For OpenAPI 3.0 schemas, the "outermost" `default` is used. For the same birthday object registered using an OAS3 schema,

```
"birthday": {
    "type": "object",
    "properties": {
        "month": {
            "type": "string",
            "default": "January"
        },
        "day": {
```

```

        "type": "integer"
    },
    "year": {
        "type": "integer"
    }
},
"default": {
    "month": "December"
}
}

```

the comment in the Apex class is

```
// Default value: December
```

because the `default` set at the object level is the outermost one, and is therefore the one that's used.

Defaults set at the composed object level for `anyOf` and `oneOf` object compositions aren't supported. In the following example, the

```
"default": { "phoneNumber": "1 (000) 000 0000"}
```

isn't used.

```

"Phone": {
    "oneOf": [
        {
            "$ref": "#/components/schemas/Contact"
        },
        {
            "type": "object",
            "properties": {
                "phoneNumber": {
                    "type": "string"
                }
            }
        }
    ],
    "default": {
        "phoneNumber": "1 (000) 000 0000"
    }
}

```

However, defaults set at the composed object level for `allof` are supported. Again, in this scenario, the "outermost" default values override the others. In the following fax object example, if the only default set

was `123`, then that's what is shown in the Apex class. But if both `123` and `456` were set, `456` would override `123` and `456` would be shown in the apex class, and similarly if `123`, `456`, `789` were all set, `789` would be shown in the apex class.

```
"Fax": {  
    "allof": [  
        {  
            "$ref": "#/components/schemas/Contact"  
        },  
        {  
            "type": "object",  
            "properties": {  
                "faxNumber": {  
                    "type": "integer",  
                    "default": 123  
                }  
            }  
        },  
        {  
            "default": {  
                "faxNumber": 456  
            }  
        }  
    ],  
    "default": {  
        "faxNumber": 789  
    }  
}
```

Operation Output Parameters

Learn about how External Services interprets an operation's output parameters.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

- Every operation captures the HTTP response code with an output parameter `responseCode`. The value of `responseCode` reflects the HTTP response code after the external call completes.
- If the operation's response parameter (OpenAPI) is defined under an HTTP response code value or is defined as `default`, then the operation output parameter name is the HTTP response code value or `default`. For an output parameter object type, its name contains the HTTP response code value.

- For all operations where the HTTP response codes aren't defined in the OpenAPI specification, the default output parameter with name `default` and of type `String` returns the external call's response as a string.
- These response content types are supported:
 - `application/json`
 - `application/octet-stream`
 - `application/x-www-form-urlencoded`
 - `text/plain`

The response content type is specified in the responses section of the spec. For example:

```
responses:  
  '200':  
    description: Successful response  
    content:  
      application/json:  
        schema:  
          type: object
```

See Also

[Schema Examples](#)

Structure Data Using properties and additionalProperties

In an Open API specification, the `properties` keyword defines one or more properties on an object including the property name and data type. The `additionalProperties` keyword gives you the flexibility to add additional properties with an undefined schema. The named properties are accessible as Apex properties with a matching data type. The `additionalProperties` in the spec are accessible in Apex as a Map property.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

- The `properties` and `additionalProperties` under an OpenAPI schema directive show as formal object properties and as a dictionary property. If declared under a property or `additionalProperties` type, then the OpenAPI parser ignores one or the other. The registration process doesn't throw an error.
- OpenAPI named `properties` are properties in Apex with the same name and property data type.
- OpenAPI `additionalProperties` are grouped as the Apex property with name `properties` and type `Map<String, Type>`, where `Type` is the declared `additionalProperties` type. The Map property type is a collection of key-value pairs.
- OpenAPI `additionalProperties` are always declared as an Apex object map property, even if it could be declared as a standalone Apex Map type. The result is consistently handled named object

properties defined together with `additionalProperties`.

- OpenAPI `properties` and `additionalProperties` can both be declared under an OpenAPI parameter schema or schema in the definitions section. The OpenAPI parser ignores either `properties` or `additionalProperties` if declared as an object property type. An object property type must only define named `properties` or `additionalProperties`, but not both. To work around, place the object property definition as a named schema under definitions and reference it by name.
- The OpenAPI parser doesn't differentiate between literal declarations or untyped schemas. Declarations like `additionalProperties: true`, `additionalProperties: false`, or `additionalProperties: {}` are interpreted as untyped. Untyped `additionalProperties` are ignored. There isn't a workaround to define `additionalProperties` that can be of any type.
- Flow doesn't allow access or manipulation of Apex object types with Map properties, but transparently preserves the content when assigned to variables of the same Apex object type. To manipulate map data structure in flow, call an Apex invocable action that can access the map data structure. For an example with External Service maps in action, see Example 9 in [External Services OpenAPI 2.0 Schema](#).
- In any external service registration you create in system version 7 and later, you can define a property with a name that begins with a number. When that property is accessed in Flow or Apex code, the encoding process adjusts the name of the property.

```
{  
    "UserAvatarUrls": {  
        "type": "object",  
        "properties": {  
            "24x24": {  
                "type": "string",  
                "description": "The URL of the user's 24x24 pixel avatar.",  
                "format": "uri"  
            },  
            "32x32": {  
                "type": "string",  
                "description": "The URL of the user's 32x32 pixel avatar.",  
                "format": "uri"  
            },  
            "16x16": {  
                "type": "string",  
                "description": "The URL of the user's 16x16 pixel avatar.",  
                "format": "uri"  
            },  
            "48x48": {  
                "type": "string",  
                "description": "The URL of the user's 48x48 pixel avatar.",  
                "format": "uri"  
            }  
        }  
    }  
}
```

```
    }  
}
```

See Also

[Apex Developer Guide: Apex Properties](#)

Apex Class Names and Developer Names

Learn how External Services derives Apex class names from your schema.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

The derived Apex class name for a registered API specification schema object can be a maximum of 255 characters. Ensure that class names are the correct length. To create a shorter name if errors are encountered during registration, choose a short service name, or edit the schema object name definition.

The *operation* developer name is derived from the registered API specification. The *object* developer name is derived from the Apex class name. The developer name must fit within a maximum of 80 characters. If the name is larger than 80 characters, a unique developer name is automatically derived to fit within 80 characters. If the individual derived object or operation unique developer names in the nested data structure don't themselves exceed 80 characters, then the nested data structure name can go up to 255 characters. If name collision errors are encountered for existing registrations, for example, to activate additional operations, try these steps:

- If the registration has been distributed in a managed package or is in use by flows, save a new copy of the registration.
- If the registration isn't in use, either delete and recreate the registration or make a copy, delete the original registration, and save the copy as the original registration.

See Also

[Schema Examples](#)

[Invoke External Service Callouts Using Apex](#)

Apex Reserved Keywords

Learn about External Service's support for Apex reserved keywords.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

Although not a best practice, External Services doesn't prohibit schema entities with the same name as the Apex reserved keywords. If your schema name conflicts with an Apex reserved keyword, External Services creates a unique name by encoding it to be compatible with Apex identifier rules.

For example, you have an account object with property name `type`. Because the property name `type` conflicts with the Apex reserved keyword `type`, External Services encodes the property name `type` as `z0type` to make it Apex compliant. During callout, the property name `type` is used according to the original OpenAPI schema.

Special characters that aren't valid Apex identifiers are UTF-8 encoded. For example,

```
5getOpen-bankingV2.2Atms
```

is encoded as

```
x35getOpenx2dbankingV2x2e2Atms
```

for a valid Apex identifier.

Although the character underscore “`_`” is a valid Apex identifier character, it's UTF-8 encoded and is used for separating parts lexicographically defined in an External Services spec. For example,

```
fixed_array_of_AutoContext
```

includes the special character underscore “`_`”. The character underscore is used as a parts separator for External Services hierarchical object names. In this example, the character underscore is encoded as

```
fixedx5farrayx5fofx5fAutoContext
```

for a valid Apex identifier. You don't need to change the name in the schema, as External Services runtime translates this back to the character underscore when calling the service.

See Also

[Schema Examples](#)

[Invoke External Service Callouts Using Apex](#)

Null Values

Learn null values are interpreted by External Services.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

OpenAPI 2.0 doesn't support the JSON or YAML schema type `null`. See [JSON Schema - 4.2.1 Instance Data Model](#).

OpenAPI 3.0 has added support to allow values to be nullable. See [OpenAPI 3 - Schema Object](#).

External services allow null values in the request payload or the response payload for both OpenAPI 2.0 and 3.0 specifications regardless of whether the specification forbids or explicitly allows its use.

 **Tip** If null values aren't allowed for your service integration, validate the request payload for null values before invoking an external service.

Schema Update Support

You can register an updated schema version for one currently in use in flow or Apex that includes *supported* components. This section provides details about whether changes are supported.

Supported

Adding:

- Actions / Operations
- Objects
- Properties
- Parameters

Deleting:

- Actions: inactive, or unused active operations
- Parameters: required parameters from operations not in use
- Parameters: optional parameters from operations in use or not in use
- Properties: required properties from objects not in use
- Properties: optional properties from objects in use or not in use

Deleted parameters or properties used in flow or Apex can lead to flow errors. Inspect your flow or Apex class for any errors due to type changes.

Changing:

- The description or example, which doesn't affect the name
- Changing the casing of an operation, parameter, object or property name

To edit a registration, or to update a schema with a new version, see [Manage External Services](#).

Not Supported

Deleting or omitting:

- Active actions / Operations in use by flow or Apex
- Required parameters from operations in use
- Required properties from objects in use

Changing:

- A name change is considered deleting the element with that name and adding the element with the new name.
- A required property or required parameter type change is only allowed if it's not in use.
- An optional property or optional parameter type change is allowed. Errors can result if used in flow or Apex. Inspect your flow or Apex class for any errors due to type changes.

Register an External Service

Provide an API spec that describes your endpoint's services and methods. The API spec's schema generates the external service operations with corresponding input and output parameters. You can also edit an existing registration, register an external service with a Mulesoft API, or register an external service using Flow Builder's HTTP Callout functionality.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

[Define an External Credential and a Named Credential](#)

First, create an external credential to specify an authentication protocol and permission set or profile to use when authenticating to an external system. Second, in order for External Services to authenticate, create a named credential and specify it as the callout endpoint.

[Register an External Service](#)

Registering an external service takes only a few steps and no code.

[Register an HTTP Callout Action in a Flow](#)

Automatically register an external service by declaratively defining an HTTP Callout action in Flow Builder.

[Manage External Services](#)

Find services that you already registered, view actions for a service, edit, clone, and delete a service. You can update a registered schema version currently in use in flow or Apex, if the update includes supported components.

Media Type Mapping in External Service Registrations

You can register specifications with nonsupported `consumes`, `produces`, or OpenAPI 3.0 content media type directives by mapping the media types to supported media types. Specify a mapping from a dropdown menu on the registration screen, or provide the mapping with code by using Metadata API.

Define a Charset in the Schema

To specify a `charset` for external callouts, populate the `Content-Type` header. A `charset` can be set if specifically defined in the schema.

Define an External Credential and a Named Credential

First, create an external credential to specify an authentication protocol and permission set or profile to use when authenticating to an external system. Second, in order for External Services to authenticate, create a named credential and specify it as the callout endpoint.

REQUIRED EDITIONS

Important In Winter '23, Salesforce introduced an improved named credential that is extensible, customizable, and more secure. We strongly recommend that you use this preferred credential instead of legacy named credentials, which are no longer updated or enhanced. For information on extensible, customizable named credentials, see [Named Credentials Schema](#).

Available in: both Salesforce Classic and Lightning Experience

Available in: All Editions

USER PERMISSIONS NEEDED

To view named credentials: View Setup and Configuration

To create, edit, or delete named credentials: Customize Applications

Ensure that you have the OpenAPI endpoint information for the service that you're registering. You use the endpoint information to set up a named credential. A named credential is the method External Services uses for authentication. A named credential specifies the URL of a callout endpoint (the service you want to access) and its authentication parameters. For example,

`https://my_endpoint.example.com`

If you use OpenAPI 2.0, your endpoint is relative to the base URL. So if your named credential URL is `https://my_endpoint.example.com` and the basePath is `/basepath`, they combine as the final endpoint of `https://my_endpoint.example.com/basepath`.

If you use OpenAPI 3.0, the named credential URL is composed of the spec's server URL's root path. The server's URL relative path is combined with the named credential's endpoint. If your named credential URL is `https://my_endpoint.example.com` and the server URL is `https://my_endpoint.example.com/relative/path`, then the server URL's relative path

`relative/path` is combined with the named credentials endpoint `https://my_endpoint.example.com`. You can reuse the same named credentials for external services with the same domain.

External Services supports all named credential authoring schemes. OpenAPI security schemes are ignored.

1. [Create and Edit an External Credential](#).

An external credential represents the details of how Salesforce authenticates to an external system via an authentication protocol. It also links to a user's permission set. Before creating a named credential, create one or more external credentials for it to link to.

2. [Create and Edit a Named Credential](#).

A named credential specifies the URL of a callout endpoint and its required authentication parameters in one definition. A named credential can be specified as an endpoint to simplify the setup of authenticated callouts. Named credentials connect to external credentials.

3. Set up a named credential for the BankService examples in this document.

If you're creating a Named Credential to follow the BankService examples, use "legacy" Named Credentials. For these examples, there's no need to set up an External Credential. When creating the Named Credential, select **New Legacy** from the dropdown menu next to the New button instead of clicking **New**. Create the **Name** and enter the **URL**. Don't change any of the other default settings. For more information, see [Define a Legacy Named Credential](#).

For additional information and configuration details such as per-user authorization, custom headers, and user External Credentials, see [Named Credentials and External Credentials](#).

Register an External Service

Registering an external service takes only a few steps and no code.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To manage connections:	Customize Application
------------------------	-----------------------

You can upload your spec for registration in the following ways.

[Submit an API Spec With a Relative URL](#)

Provide a relative URL to the spec.

[Submit an API Spec With an Absolute URL](#)

Provide a full, absolute URL to the spec.

Paste the JSON or YAML formatted Schema Into Your Browser

Copy and paste the JSON or YAML formatted schema into your browser.

Upload a Local File

Using your browser, drag, or browse files to upload the JSON or YAML file from your computer.

See Also

[External Services](#)

[Named Credentials](#)

[Choose an Authentication Protocol](#)

[External Services OpenAPI 2.0 Schema](#)

[External Services OpenAPI 3.0 Schema](#)

[Using the Schema Examples](#)

Submit an API Spec With a Relative URL

Provide a relative URL to the spec.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To manage connections: Customize Application

To register an API spec using a relative URL:

1. From Setup in Lightning Experience, enter *External Services* in the Quick Find box, and select **External Services**.
2. Click **New External Service**.
3. On the Select an API Source page, select whether you're importing an API spec **From Mulesoft Anypoint Platform**, or **From API Specification**. Since we're using our own spec in this example, select **From API Specification**.
4. For External Service Name, enter a unique service name.
5. Enter an optional description.
A good, short description can help distinguish one service from another in a long list of services.
6. Click under **Select a Named Credential** and select a named credential from the list.
7. Under **Service Schema**, select **Relative URL**.
8. Enter a relative URL path to the schema (endpoint) under **URL**.
It must begin with "/" and be a relative path. For example, `/accountSchema.json`.
9. If one or more errors are found during the validation process, a message (or series of messages) gives specific details about the error. You make corrections to the schema by editing it directly in the text editor and clicking **Validate** to revalidate your changes, or by modifying an offline copy and re-registering it. There are two types of errors:

- Syntax or logical errors—*Must be fixed to complete the registration process.* To resolve an error, use the error message to discover the type and location of the error. Make corrections, and click **Validate**. Syntax error messages provide line and column number location.
- Syntax or logical warnings—You don't need to fix warnings to complete a registration. But the respective schema components are ignored by the External Services and aren't registered. To fix warnings, use the guidance in the warning message, make corrections, and click **Validate**.

10. Click **Save & Next**.



Note If your schema includes non-supported media types, the system displays the **Associate operations with supported media types** page. Select a valid mapping for each non-supported type, as described in [Mapping Media Types During Registration](#).

11. Select the operations you want to import into your external service registration. You can select up to 3,000 *operations* per org, and up to 3,000 active *objects* per org. If you exceed either of these limits, divide the operations in your schema across separate registrations. For information about limits, see [External Services Considerations](#). For information about “objects” in Open API schemas, see [Basic Structure \(OpenAPI 2.0\)](#) and [Basic Structure \(OpenAPI 3.0\)](#)

12. Click **Next**.

13. Click **Done**.

Submit an API Spec With an Absolute URL

Provide a full, absolute URL to the spec.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To manage connections: Customize Application

Important Before you can use an absolute URL, you must [add the URL to your Remote Sites settings](#).

To register an API spec using an absolute URL:

1. Make sure that the URL has been [added to your Remote Sites settings](#).
2. From Setup in Lightning Experience, enter *External Services* in the Quick Find box, and select **External Services**.
3. Click **New External Service**.
4. On the **Select an API Source** page, select whether you're importing an API spec **From Mulesoft Anypoint Platform**, or **From API Specification**. Since we're using our own spec in this example, select **From API Specification**.
5. Under **External Service Name**, enter a unique service name.

6. Under **Description**, enter an optional description. A good, short description can help distinguish one service from another in a long list of services.
 7. Click under **Select a Named Credential** and select a named credential from the list.
 8. Under **Service Schema**, select **Absolute URL** to upload and register your schema. The URL must point to a valid OpenAPI 2.0 or 3.0, and a JSON or YAML-compliant schema.
 9. Enter an absolute URL path to the schema under **URL**. For example,
`https://my_endpoint.example.com/accountSchema.json`. The absolute URL can point to a different domain than the one specified in Named Credentials.
 10. Press Enter on your keyboard, or click outside of the URL box. The system begins to validate the schema.
 11. If one or more errors are found during the validation process, a message (or series of messages) gives specific details about the error. You make corrections to the schema by editing it directly in the text editor and clicking **Validate** to revalidate your changes, or by modifying an offline copy and re-registering it. There are two types of errors:
 - Syntax or logical errors—*Must be fixed to complete the registration process*. To resolve an error, use the error message to discover the type and location of the error. Make corrections, and click **Validate**. Syntax error messages provide line and column number location.
 - Syntax or logical warnings—You don't need to fix warnings to complete a registration. But the respective schema components are ignored by the External Services and aren't registered. To fix warnings, use the guidance in the warning message, make corrections, and click **Validate**.
 12. Click **Save & Next**.
-  **Note** If your schema includes non-supported media types, the system displays the **Associate operations with supported media types** page. Select a valid mapping for each non-supported type, as described in [Mapping Media Types During Registration](#).
13. Select the operations you want to import into your external service registration. You can select up to 3,000 *operations* per org, and up to 3,000 active *objects* per org. If you exceed either of these limits, divide the operations in your schema across separate registrations. For information about limits, see [External Services Considerations](#). For information about “objects” in Open API schemas, see [Basic Structure \(OpenAPI 2.0\)](#) and [Basic Structure \(OpenAPI 3.0\)](#)
 14. Click **Next**.
 15. Click **Done**.

Paste the JSON or YAML formatted Schema Into Your Browser

Copy and paste the JSON or YAML formatted schema into your browser.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To manage connections:

Customize Application

To register an API spec by copying and pasting its JSON or YAML-compliant schema into your browser:

1. From Setup in Lightning Experience, enter *External Services* in the Quick Find box, and select **External Services**.
2. Click **New External Service**.
3. On the **Select an API Source** page, select whether you're importing an API spec **From Mulesoft Anypoint Platform**, or **From API Specification**. Since we're using our own spec in this example, select **From API Specification**.
4. Under **External Service Name**, enter a unique service name.
5. Under **Description**, enter an optional description. A good, short description can help distinguish one service from another in a long list of services.
6. Click under **Select a Named Credential** and select a named credential from the list.
7. Under **Service Schema**, select **Complete Schema**. Provide a schema that's complete, valid for OpenAPI 2.0 or 3.0, and is JSON or YAML-compliant.
8. Paste your JSON or YAML-compliant schema into the provided text field.
9. Click **Save & Next**. Your uploaded schema is validated by External Services.



Note If your schema includes non-supported media types, the system displays the **Associate operations with supported media types** page. Select a valid mapping for each non-supported type, as described in [Mapping Media Types During Registration](#).

10. If one or more errors are found during the validation process, a message (or series of messages) gives specific details about the error. You make corrections to the schema by editing it directly in the text editor and clicking **Validate** to revalidate your changes, or by modifying an offline copy and re-registering it. There are two types of errors:
 - Syntax or logical errors—*Must be fixed to complete the registration process*. To resolve an error, use the error message to discover the type and location of the error. Make corrections, and click **Validate**. Syntax error messages provide line and column number location.
 - Syntax or logical warnings—You don't need to fix warnings to complete a registration. But the respective schema components are ignored by the External Services and aren't registered. To fix warnings, use the guidance in the warning message, make corrections, and click **Validate**.
11. Select the operations you want to import into your external service registration. You can select up to 3,000 *operations* per org, and up to 3,000 active *objects* per org. If you exceed either of these limits, divide the operations in your schema across separate registrations. For information about limits, see [External Services Considerations](#). For information about “objects” in Open API schemas, see [Basic Structure \(OpenAPI 2.0\)](#) and [Basic Structure \(OpenAPI 3.0\)](#)
12. Click **Next**.
13. Click **Done**.

Upload a Local File

Using your browser, drag, or browse files to upload the JSON or YAML file from your computer.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To manage connections: Customize Application

To upload a local file:

1. From Setup in Lightning Experience, enter *External Services* in the Quick Find box, and select **External Services**.
2. Click **New External Service**.
3. On the **Select an API Source** page, select whether you're importing an API spec **From Mulesoft Anypoint Platform**, or **From API Specification**. Since we're using our own spec in this example, select **From API Specification**.
4. Under **External Service Name**, enter a unique service name.
5. Under **Description**, enter an optional description. A good, short description can help distinguish one service from another in a long list of services.
6. Under **Service Schema**, select **Upload from local**. Provide a complete, valid OpenAPI 2.0 or 3.0, and a JSON or YAML-compliant schema.
7. Click under **Select a Named Credential** and choose a named credential from the list.
8. Either click **Upload from local** and navigate to the JSON or YAML file on your local machine, or drag the file within the dotted box.
9. If one or more errors are found during the validation process, a message (or series of messages) gives specific details about the error. You make corrections to the schema by editing it directly in the text editor and clicking **Validate** to revalidate your changes, or by modifying an offline copy and re-registering it. There are two types of errors:
 - Syntax or logical errors—*Must be fixed to complete the registration process*. To resolve an error, use the error message to discover the type and location of the error. Make corrections, and click **Validate**. Syntax error messages provide line and column number location.
 - Syntax or logical warnings—You don't need to fix warnings to complete a registration. But the respective schema components are ignored by the External Services and aren't registered. To fix warnings, use the guidance in the warning message, make corrections, and click **Validate**.
10. Click **Save & Next**.



Note If your schema includes non-supported media types, the system displays the **Associate operations with supported media types** page. Select a valid mapping for each non-supported type, as described in [Mapping Media Types During Registration](#).

11. Select the operations you want to import into your external service registration. You can select up to 3,000 *operations* per org, and up to 3,000 active *objects* per org. If you exceed either of these limits, divide the operations in your schema across separate registrations. For information about limits, see [External Services Considerations](#). For information about “objects” in Open API schemas, see [Basic Structure \(OpenAPI 2.0\)](#) and [Basic Structure \(OpenAPI 3.0\)](#)
12. Click **Next**.
13. Click **Done**.

Register an HTTP Callout Action in a Flow

Automatically register an external service by declaratively defining an HTTP Callout action in Flow Builder.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

After you configure the HTTP callout action in a flow, Flow Builder auto-generates an external service registration, an invocable action, and Apex-defined types. You can then use the data output of the API request as input within Flow Builder and across Salesforce.

See Also

[HTTP Callout](#)

Manage External Services

Find services that you already registered, view actions for a service, edit, clone, and delete a service. You can update a registered schema version currently in use in flow or Apex, if the update includes supported components.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

[Locate an External Service](#)

See a list of your registered external services.

[Edit an External Service](#)

Edit an external service's named credentials, description, schema, and included operations.

[Edit an External Service Created by HTTP Callout](#)

You can edit an external service's registration details, or use the declarative editor to change the operation's configuration.

[Recreate an External Service \(Save As\)](#)

Clone your external service under a new name, and update the related flows.

[Update a Schema](#)

For an existing External Services registration, you can update the current schema with a new version.

[Delete an External Service](#)

Delete an external service permanently.

[View Actions](#)

View actions available for an external service.

View an Action's Unique Apex Name

Use this example to understand how External Services actions in Apex or Flow map to specific actions in an External Service registration. An object's unique Apex class name is specified in View Actions > More Details. This name also appears when searching for actions in Flow Builder.

Track Usage and Limits

When you first enter External Services from Setup, five visual (flat) gauges at the top of the page show your current usage and maximum values for each per org limit.

Locate an External Service

See a list of your registered external services.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

From Setup in Lightning Experience, enter *External Services* in the Quick Find box, and select **External Services**.

All of your registered external services are listed.

Edit an External Service

Edit an external service's named credentials, description, schema, and included operations.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

You can register a new version of an API specification over an existing registration. This update works only if operations and schema objects in use by flows or referenced by Apex classes *aren't removed* in the new specification. Remove the *unused* operations and schema objects before updating to the new API specification version.

When updating a schema that has operations/objects in use by a flow, non-structural changes, such as updating a description are allowed. However, it's possible that structural changes, such as updating an object's properties or parameters, don't register.

-  **Tip** If the new schema version isn't compatible, the edit workflow notifies you which operations and schema objects are in use by which flows and by which Apex classes. With this information, you know which existing references are incompatible so that you can remove them before saving your updated registration.

For details about supported and non-supported changes, see [Schema Update Support](#).

1. From Setup, in the Quick Find box, enter *External Services* and select **External Services**.
2. Click the arrow in the service's Actions column, and then select **Edit**. You can modify these service settings.
 - Named Credential
 - Description
 - Service Schema
 - Operations
3. Optionally update the schema with a new version by submitting a new schema, or by using the inline text editor to make changes.
4. Use the instructions in [Register an External Service](#) to submit an updated schema, validate, and select operations for your edited external service.

Edit an External Service Created by HTTP Callout

You can edit an external service's registration details, or use the declarative editor to change the operation's configuration.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

[Edit the Schema or Named Credential](#)

To edit the details of an external service created by HTTP Callout, you can edit the schema directly, change the Description, or change the Named Credential.

[Edit an Operation Declaratively](#)

To edit an external service's operation created by HTTP Callout, you can use the same declarative tools you used to create the HTTP Callout in Flow Builder.

Edit the Schema or Named Credential

To edit the details of an external service created by HTTP Callout, you can edit the schema directly, change the Description, or change the Named Credential.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

1. From Setup in Lightning Experience, enter *External Services* in the Quick Find box, and select **External Services**.

2. Click the name of the external service that you want to edit.
3. Click the **Edit** button.
4. Optionally, change the **Description**, or change the Named Credential under **Select a Named Credential**.

When you save your changes, the Creation source shown on the external service's detail page remains **HTTP Callout in Flow Builder**, and therefore, can be edited again using the declarative tools.

5. Optionally, use the JSON schema editor to edit the schema.

If you change the schema and save it, the **Creation source** changes to **From API specification**. This external service's operation can no longer be edited using declarative tools, but is edited like any other external service.

6. Click **Save & Next**.
7. Select the operation(s) to include in this registration.
8. Click **Next**.
9. Review your edited action, and click **Finish**.

Edit an Operation Declaratively

To edit an external service's operation created by HTTP Callout, you can use the same declarative tools you used to create the HTTP Callout in Flow Builder.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

1. From Setup in Lightning Experience, enter *External Services* in the Quick Find box, and select **External Services**.
2. Click the name of the external service that you want to edit.
3. Locate the operation you want to edit. In the right-most column, pull down the arrow and select **Edit HTTP Callout Action**.
4. To edit your invocable action, configure the declarative settings in the **Edit HTTP Callout** window. For more details about using the declarative editor, see steps five through the end of the section in [Configure an HTTP Callout Action](#).
5. Click **Save**.

Recreate an External Service (Save As)

Clone your external service under a new name, and update the related flows.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To manage connections:

Customize Application

1. From Setup in Lightning Experience, enter *External Services* in the Quick Find box, and select **External Services**.
2. Under Actions, next to the name of the external service that you want to recreate, click **Save As**.
3. Modify the external service name so that it's different from the original registration. If you don't provide a name, External Services automatically adds the prefix **CopyOf** in front of the existing external service name.
4. Verify or change the named credential used, description, schema URL, YAML, or JSON.
5. Click **Save & Next**.
6. If needed, modify the selected operations.
7. Click **Next**.
8. Click **Done**.
9. Update the flows that use the actions from the original external service. To locate actions for External Services registrations, filter the actions by type, and select **External Service**. To find actions that aren't from External Services registrations, filter by type, and select **Apex Action**.
10. Return to External Services in Setup.
11. Under Actions, for the original external service that you recreated, click **Delete**.

See Also

[Register an External Service](#)

Update a Schema

For an existing External Services registration, you can update the current schema with a new version.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

To update the schema of an existing registration.

1. From Setup in Lightning Experience, enter *External Services* in the Quick Find box, and select **External Services**.
2. Click the arrow in the service's Actions column, and select **Edit**.
3. On the Select an API Source page, select whether you're importing an API spec From Mulesoft Anypoint Platform, or From API Specification.
4. Update the named credential, if necessary.
5. If your schema includes operations that aren't supported, you can filter them out by toggling on Only include supported operations (default). To learn why operations can't be included and to show errors, toggle it off. Toggling it off and showing errors is useful when you're troubleshooting your API spec. For example, it can help identify wrong data types, undefined object references, and wrong or invalid

OpenAPI attributes. If you aren't sure, leave it toggled on.

6. Enter a schema URL path (or endpoint) that points to the updated schema, or provide the complete, updated schema. If you enter a schema URL path, it must begin with "/" and be a relative path. For example, `/accountSchema.json`. If you provide a schema, it must be complete, valid OpenAPI 2.0 or 3.0, and JSON or YAML compliant.
7. Click **Save and Next**. If your schema includes non-supported media types, the system displays the Associate operations with supported media types page. Select a valid mapping for each non-supported type, as described in Mapping Media Types during Registration.
8. Select the operations you want to import into your external service registration, or deselect the ones you want to delete.
9. Click **Next**.
10. Click **Done**.

Delete an External Service

Delete an external service permanently.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

To delete an external service, click the arrow in the service's Actions column, and select **Delete**.

View Actions

View actions available for an external service.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

1. From Setup in Lightning Experience, enter `External Services` in the Quick Find box, and select **External Services**.
2. Click the arrow in the external service's Actions column, and select **View Actions**.

A list of registered actions for the external service is displayed.

View an Action's Unique Apex Name

Use this example to understand how External Services actions in Apex or Flow map to specific actions in an External Service registration. An object's unique Apex class name is specified in View Actions > More Details. This name also appears when searching for actions in Flow Builder.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

For example, if your API spec defines a response `accountDetails` for a `getAccount` operation, referenced in the spec as

```
"paths": {
    "/accounts/{accountName}": {
        "get": {
            "operationId": "getAccount",
            "summary": "Retrieves an account",
            "description": "Retrieves the account with specific name",
            "consumes": [
                "text/plain"
            ],
            "produces": [
                "application/json"
            ],
            "parameters": [
                {
                    "name": "accountName",
                    "in": "path",
                    "required": true,
                    "type": "string",
                    "description": "Name of the account"
                }
            ],
            "responses": {
                "200": {
                    "description": "The response when system finds an account with given name",
                    "schema": {
                        "$ref": "#/definitions/accountDetails"
                    }
                }
            }
        }
    }
}
```

Where the definition for `accountDetails` is defined at the end of the spec as

```
"definitions": {
    "accountDetails": {
        "required": [
            "id",
            "name"
        ],
        "properties": {
            "id": {
                "type": "string",
                "description": "Unique identifier for the account"
            },
            "name": {
                "type": "string",
                "description": "Name of the account"
            },
            "type": {
                "type": "string",
                "description": "Type of account (e.g., Company, Individual, etc.)"
            },
            "status": {
                "type": "string",
                "description": "Current status of the account (e.g., Active, Pending, Inactive)"
            },
            "size": {
                "type": "number",
                "description": "Size of the account (e.g., Number of employees or assets)"
            },
            "lastUpdated": {
                "type": "string",
                "format": "date-time",
                "description": "Last updated date and time of the account"
            }
        }
    }
}
```

```

        "name",
        "type",
        "availableBal"
    ],
    "properties": {
        "id": {
            "type": "string",
            "description": "id"
        },
        "name": {
            "type": "string",
            "description": "name"
        },
        "type": {
            "type": "string",
            "description": "type"
        },
        "availableBal": {
            "type": "string",
            "description": "availableBal"
        }
    }
},

```

When the spec is registered in External Services, you can see the Apex Class name during registration.



Or you can navigate to the action in External Services and then click **View Actions > More Details**.



External Services automatically registered our `accountDetails` object as `BankService_accountDetails`.

Now in Flow, when you add a resource for a new action, it's clear that you're selecting the correct action from the correct registration.

Track Usage and Limits

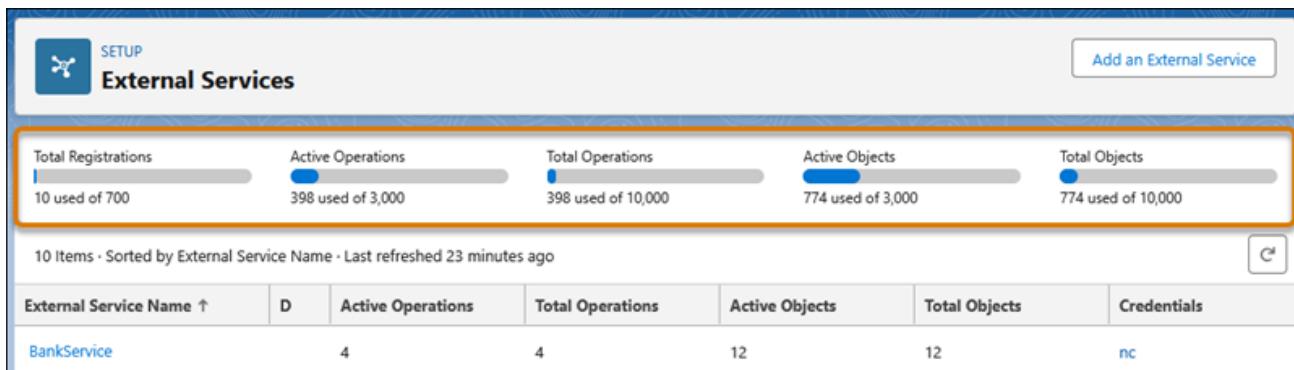
When you first enter External Services from Setup, five visual (flat) gauges at the top of the page show your current usage and maximum values for each per org limit.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

- Total Registrations—Number of External Services registrations
- Active Operations—Number of operations selected for use across all registered specs
- Total Operations—The sum of all active and inactive (not selected) operations across all registered specs
- Active Objects—Number of objects in use across all active operations in registered specs
- Total Objects—The sum of objects in use across all active operations in registered specs, and unused objects across all inactive operations in registered specs



Media Type Mapping in External Service Registrations

You can register specifications with nonsupported `consumes`, `produces`, or OpenAPI 3.0 content media type directives by mapping the media types to supported media types. Specify a mapping from a dropdown menu on the registration screen, or provide the mapping with code by using Metadata API.

Mapping Media Types During Registration

If you attempt to register a schema that includes a nonsupported media type, you're asked to provide a mapping on the **Associate operations with supported media types** page.

 **Note** The **Associate operations with supported media types** page isn't shown during registration, unless there's a nonsupported media type specified in the schema.

To associate operations with a supported media type:

1. Note the first nonsupported media type on the left.
2. Map the non supported media type to a supported media type by selecting a supported media type from the pull-down menu directly to the right.
3. If there are additional nonsupported media types in the list, continue down the list and select a mapping for each nonsupported media type listed.

4. Click **Next**.

Continue registration by selecting operations, as described in step #10 of [Register an External Service](#).

Mapping Media Types with Metadata API

To manually map a media type, the nonsupported media type must be mapped to a supported media type as a mapping in the field `serviceBinding` of the `ExternalServiceRegistration` metadata file.

To demonstrate the manual method of mapping media types, consider this Acme Bank example.

Acme bank specifies an OpenAPI 2.0 service for their credit rating service. It defines its own media type for handling the JSON compatible payload: `application/x-acme-json`:

```
{  
    "swagger": "2.0",  
    "info": {  
        "description": "A service for checking credit for an account.",  
        "version": "1.0.0",  
        "title": "Credit Decision"  
    },  
    "host": "<ACME's host>",  
    "paths": {  
        "/account/credit-rating": {  
            "get": {  
                "operationId": "getCreditRating",  
                "summary": "Evaluates credit rating for offered payment terms.",  
                "consumes": [  
                    "application/x-acme-json"  
                ],  
                "produces": [  
                    "application/x-acme-json"  
                ],  
                "parameters": [ {  
                    "description": "Account",  
                    "in": "body",  
                    "name": "body",  
                    "required": true,  
                    "schema": {  
                        "$ref": "#/definitions/Account"  
                    }  
                } ],  
                "responses": {  
                    "200": {  
                        "description": "Success",  
                        "content": {  
                            "application/x-acme-json": {  
                                "schema": {  
                                    "$ref": "#/definitions/CreditRating"  
                                }  
                            }  
                        }  
                    }  
                }  
            }  
        }  
    }  
}
```

```
        "200": {
            "description": "Credit rating",
            "schema": {
                "$ref": "#/definitions/CreditRating"
            }
        }
    }
},
"definitions": {
    "Account": {
        "type": "object",
        "properties": {
            "accountId": {
                "type": "string"
            },
            "accountHolder": {
                "type": "string"
            }
        }
    },
    "CreditRating": {
        "type": "object",
        "properties": {
            "creditRating": {
                "type": "string"
            }
        }
    }
}
}
```

To map the BankService media type to the supported media type `application/json`, open a command-line terminal. Create a directory in which to retrieve the external service's metadata.

```
> mkdir BankService
> cd BankService
```

Create the manifest package.xml for the external service metadata for your BankService to retrieve from your organization:

```
> touch package.xml
```

Edit the package.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>BankService</members>
    <name>ExternalServiceRegistration</name>
  </types>
  <version>53.0</version>
</Package>
```

Retrieve the metadata for the external service with the Salesforce CLI command after successful authentication to your organization:

```
> sf project retrieve start --target-metadata-dir . --manifest package.xml
```

Unzip the package zip with the metadata and navigate to the directory with the external service:

```
> unzip unpackaged.zip
> cd unpackaged/externalServiceRegistrations
```

Edit service bindings in the external service registration metadata file

`BankService.externalServiceRegistration`. Don't break the `<serviceBinding>` section (in bold) with line breaks as it breaks deserialization.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExternalServiceRegistration xmlns="http://soap.sforce.com/2006/04/metadata">
  <label>MediaTypeMap</label>
  <namedCredential>BankServiceEndpoint</namedCredential>
  <operations>
    <active>true</active>
    <name>getcreditrating</name>
  </operations>
  <registrationProviderType>Custom</registrationProviderType>
  <schema>{
    ...
  }
  </schema>
  <schemaType>OpenApi</schemaType>
  <serviceBinding>{&quot;compatibleMediaTypes&quot;:
  :{&quot;application/x-acme-json&quot;;
  :&quot;application/x-acme-json&quot;}}</serviceBinding>
```

```
<status>Complete</status>
</ExternalServiceRegistration>
```

The service binding specifies in JSON format the nonsupported media types defined by this external service registration:

```
{ "compatibleMediaTypes":{
    "application/x-acme-json":"application/x-acme-json"
} }
```

Map the nonsupported media type to the supported media type for the external service payload serialization:

```
{ "compatibleMediaTypes":{
    "application/x-acme-json":"application/json"
} }
```

The updated metadata for the bank rating external service sample registration respects the media type mapping to the supported media type when deployed in a package:

```
<?xml version="1.0" encoding="UTF-8"?>
<ExternalServiceRegistration xmlns="http://soap.sforce.com/2006/04/metadata">
    <label>MediaTypeMap</label>
    <namedCredential>BankService</namedCredential>
    <operations>
        <active>true</active>
        <name>getcreditrating</name>
    </operations>
    <registrationProviderType>Custom</registrationProviderType>
    <schema>{
        ...
    }
    </schema>
    <schemaType>OpenApi</schemaType>
    <serviceBinding>{&quot;compatibleMediaTypes&quot;;
    :{&quot;application/x-acme-json&quot;;
    :&quot;application/json&quot;} }</serviceBinding>
        <status>Complete</status>
    </ExternalServiceRegistration>
```

Don't break the `<serviceBinding>` section with line breaks as it breaks deserialization.

Save the edited external service registration metadata file and deploy it:

```
> cd ../../  
> sf project deploy start --metadata-dir unpackaged
```

For more information, see:

- [ExternalServiceRegistration](#)
- [Salesforce DX Developer Guide](#)

Define a Charset in the Schema

To specify a `charset` for external callouts, populate the `Content-Type` header. A `charset` can be set if specifically defined in the schema.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

During an external callout, the `Content-Type` header is used to describe the media type (for example, `application/json`, `text/plain`, etc.) of a resource that can be consumed by the operation.

`charset` is an optional parameter of `Content-Type` that specifies the character encoding of the media type.

 **Note** With the Spring '23 release, `charset` in the `Content-Type` header is no longer populated by default when making external callouts. Previously, `Content-Type` included the `charset=UTF-8` parameter by default.

As an example, we'll edit the BankService schema to set `charset` to `UTF-8`. As OpenAPI 2.0 and Open API 3.0 schema versions are implemented differently, we'll use both versions.

Example Define `charset` in an OpenAPI 2.0 Schema

```
{  
    "swagger": "2.0",  
    "basePath": "/",  
    "info": {  
        "version": "1.0",  
        "title": "External Service for demo bank",  
        "description": "### External Service for demo bank",  
        "x-vcap-service-name": "DemoBankRestServices"  
    },
```

```
"securityDefinitions": {
    "basicAuth": {
        "type": "basic"
    }
},
"security": [
    {
        "basicAuth": [
            ]
    }
],
"tags": [
    {
        "name": "DemoBankRestServices"
    }
],
"paths": {
    "/accounts/{accountName)": {
        "get": {
            "operationId": "getAccount",
            "summary": "Retrieves an account",
            "description": "Retrieves the account with specific name",
            "consumes": [
                "text/plain; charset=UTF-8"
            ],
            "produces": [
                "application/json"
            ],
            "parameters": [
                {
                    "name": "accountName",
                    "in": "path",
                    "required": true,
                    "type": "string",
                    "description": "Name of the account"
                }
            ],
            "responses": {
                "200": {
                    "description": "The response when system finds an account with given name",
                    "schema": {

```

```
        "$ref": "#/definitions/accountDetails"
    }
},
"400": {
    "description": "Error response if the account name
parameter is less than minimum characters",
    "schema": {
        "$ref": "#/definitions/errorModel"
    }
},
"404": {
    "description": "Error response if the account is no
t supported by service or account is not found",
    "schema": {
        "$ref": "#/definitions/errorModel"
    }
}
},
"delete": {
    "operationId": "DeleteAccount",
    "summary": "Deletes an account",
    "description": "Deletes the account with specific name",
    "consumes": [
        "text/plain; charset=UTF-8"
    ],
    "produces": [
        "application/json"
    ],
    "parameters": [
        {
            "name": "accountName",
            "in": "path",
            "required": true,
            "type": "string",
            "description": "Name of the account"
        }
    ],
    "responses": {
        "204": {
            "description": "The response when system finds an a
ccount with given name",
            "schema": {

```

```
        "type":"string"
    }
},
"400": {
    "description":"Error response if the account name
parameter is less than minimum characters",
    "schema": {
        "$ref": "#/definitions/errorModel"
    }
},
"404": {
    "description":"Error response if the account is no
t supported by service or account is not found",
    "schema": {
        "$ref": "#/definitions/errorModel"
    }
}
},
"post": {
    "operationId": "addAccount",
    "summary": "Add an account",
    "description": "Add an account to the database",
    "consumes": [
        "text/plain; charset=UTF-8"
    ],
    "produces": [
        "application/json"
    ],
    "parameters": [
        {
            "name": "accountName",
            "in": "path",
            "required": true,
            "type": "string",
            "description": "Name of the account"
        },
        {
            "name": "accountType",
            "in": "query",
            "required": true,
            "type": "string",
            "description": "The type of account"
        }
    ]
}
```

```
        }
    ],
    "responses": {
        "201": {
            "description": "The response when the account does not already exist and we can create one",
            "schema": {
                "$ref": "#/definitions/accountDetails"
            }
        },
        "409": {
            "description": "The response when the account already exists and we cannot create one",
            "schema": {
                "$ref": "#/definitions/accountDetails"
            }
        },
        "400": {
            "description": "Error response if the account name parameter is less than minimum characters",
            "schema": {
                "$ref": "#/definitions/errorModel"
            }
        },
        "404": {
            "description": "Error response if the account is not supported by service or account is not found",
            "schema": {
                "$ref": "#/definitions/errorModel"
            }
        }
    }
},
"put": {
    "operationId": "updateAccount",
    "summary": "Updates an account",
    "description": "Updates the account with specified name",
    "consumes": [
        "text/plain; charset=UTF-8"
    ],
    "produces": [
        "application/json"
    ],
}
```

```
"parameters": [
    {
        "name": "accountName",
        "in": "path",
        "required": true,
        "type": "string",
        "description": "Name of the account"
    },
    {
        "name": "accountType",
        "in": "query",
        "required": true,
        "type": "string",
        "description": "The type of account"
    }
],
"responses": {
    "200": {
        "description": "The response when system finds an account with given name",
        "schema": {
            "$ref": "#/definitions/accountDetails"
        }
    },
    "400": {
        "description": "Error response if the account name parameter is less than minimum characters",
        "schema": {
            "$ref": "#/definitions/errorModel"
        }
    },
    "404": {
        "description": "Error response if the account is not supported by service or account is not found",
        "schema": {
            "$ref": "#/definitions/errorModel"
        }
    }
}
},
"definitions": {
```

```
"accountDetails":{  
    "required": [  
        "id",  
        "name",  
        "type",  
        "availableBal"  
    ],  
    "properties": {  
        "id": {  
            "type": "string",  
            "description": "id"  
        },  
        "name": {  
            "type": "string",  
            "description": "name"  
        },  
        "type": {  
            "type": "string",  
            "description": "type"  
        },  
        "availableBal": {  
            "type": "string",  
            "description": "availableBal"  
        }  
    }  
},  
"errorModel":{  
    "required": [  
        "errorCode",  
        "errorMessage"  
    ],  
    "properties": {  
        "errorCode": {  
            "type": "string",  
            "description": "A service-specific error code."  
        },  
        "errorMessage": {  
            "type": "string",  
            "description": "A service-specific error code."  
        }  
    }  
}
```

```
}
```

Example Define `charset` in an OpenAPI 3.0 Schema

```
{
  "openapi": "3.0.1",
  "info": {
    "title": "External Service for demo bank",
    "description": "### External Service for demo bank",
    "version": "1.0",
    "x-vcap-service-name": "DemoBankRestServices"
  },
  "servers": [
    {
      "url": "/"
    }
  ],
  "security": [
    {
      "basicAuth": []
    }
  ],
  "tags": [
    {
      "name": "DemoBankRestServices"
    }
  ],
  "paths": {
    "/accounts/{accountName)": {
      "get": {
        "summary": "Retrieves an account",
        "description": "Retrieves the account with specific name",
        "operationId": "getAccount",
        "parameters": [
          {
            "name": "accountName",
            "in": "path",
            "description": "Name of the account",
            "required": true,
            "schema": {
              "type": "string"
            }
          }
        ]
      }
    }
  }
}
```

```
],
  "responses": {
    "200": {
      "description": "The response when system finds an account with given name",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/accountDetails"
          }
        }
      }
    },
    "400": {
      "description": "Error response if the account name parameter is less than minimum characters",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/errorModel"
          }
        }
      }
    },
    "404": {
      "description": "Error response if the account is not supported by service or account is not found",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/errorModel"
          }
        }
      }
    }
  },
  "put": {
    "summary": "Updates an account",
    "description": "Updates the account with specified name",
    "operationId": "updateAccount",
    "parameters": [
      {
        "in": "body",
        "name": "account",
        "schema": {
          "$ref": "#/components/schemas/accountDetails"
        }
      }
    ]
  }
}
```

```
        "name": "accountName",
        "in": "path",
        "description": "Name of the account",
        "required": true,
        "schema": {
            "type": "string"
        }
    },
{
    "name": "accountType",
    "in": "query",
    "description": "The type of account",
    "required": true,
    "schema": {
        "type": "string"
    }
}
],
"requestBody": {
    "content": {
        "application/x-www-form-urlencoded; charset=UTF-8": {
            "schema": {
                "type": "string"
            }
        }
    }
},
"responses": {
    "200": {
        "description": "The response when system finds an account with given name",
        "content": {
            "application/json": {
                "schema": {
                    "$ref": "#/components/schemas/accountDetails"
                }
            }
        }
    },
    "400": {
        "description": "Error response if the account name parameter is less than minimum characters",
        "content": {

```

```
        "application/json": {
            "schema": {
                "$ref": "#/components/schemas/errorModel"
            }
        }
    },
    "404": {
        "description": "Error response if the account is not supported by service or account is not found",
        "content": {
            "application/json": {
                "schema": {
                    "$ref": "#/components/schemas/errorModel"
                }
            }
        }
    }
},
"post": {
    "summary": "Add an account",
    "description": "Add an account to the database",
    "operationId": "addAccount",
    "parameters": [
        {
            "name": "accountName",
            "in": "path",
            "description": "Name of the account",
            "required": true,
            "schema": {
                "type": "string"
            }
        },
        {
            "name": "accountType",
            "in": "query",
            "description": "The type of account",
            "required": true,
            "schema": {
                "type": "string"
            }
        }
    ]
}
```

```
],
  "requestBody": {
    "content": {
      "application/x-www-form-urlencoded; charset=UTF-8": {
        "schema": {
          "type": "string"
        }
      }
    }
  },
  "responses": {
    "201": {
      "description": "The response when the account does not already exist and we can create one",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/accountDetails"
          }
        }
      }
    },
    "400": {
      "description": "Error response if the account name parameter is less than minimum characters",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/errorModel"
          }
        }
      }
    },
    "404": {
      "description": "Error response if the account is not supported by service or account is not found",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/errorModel"
          }
        }
      }
    }
  }
}
```

```
        },
        "409": {
            "description": "The response when the account already exists a
nd we cannot create one",
            "content": {
                "application/json": {
                    "schema": {
                        "$ref": "#/components/schemas/accountDetails"
                    }
                }
            }
        }
    },
    "delete": {
        "summary": "Deletes an account",
        "description": "Deletes the account with specific name",
        "operationId": "DeleteAccount",
        "parameters": [
            {
                "name": "accountName",
                "in": "path",
                "description": "Name of the account",
                "required": true,
                "schema": {
                    "type": "string"
                }
            }
        ],
        "responses": {
            "204": {
                "description": "The response when system finds an account with
given name",
                "content": {
                    "application/json": {
                        "schema": {
                            "type": "string"
                        }
                    }
                }
            },
            "400": {
                "description": "Error response if the account name parameter i

```

```
s less than minimum characters",
    "content": {
        "application/json": {
            "schema": {
                "$ref": "#/components/schemas/errorModel"
            }
        }
    }
},
"404": {
    "description": "Error response if the account is not supported by service or account is not found",
    "content": {
        "application/json": {
            "schema": {
                "$ref": "#/components/schemas/errorModel"
            }
        }
    }
}
},
"components": {
    "schemas": {
        "accountDetails": {
            "required": [
                "availableBal",
                "id",
                "name",
                "type"
            ],
            "type": "object",
            "properties": {
                "id": {
                    "type": "string",
                    "description": "id"
                },
                "name": {
                    "type": "string",
                    "description": "name"
                },
                ...
            }
        }
    }
}
```

```
    "type": {
      "type": "string",
      "description": "type"
    },
    "availableBal": {
      "type": "string",
      "description": "availableBal"
    }
  },
  "errorModel": {
    "required": [
      "errorCode",
      "errorMessage"
    ],
    "type": "object",
    "properties": {
      "errorCode": {
        "type": "string",
        "description": "A service-specific error code."
      },
      "errorMessage": {
        "type": "string",
        "description": "A service-specific error code."
      }
    }
  },
  "securitySchemes": {
    "basicAuth": {
      "type": "http",
      "scheme": "basic"
    }
  }
}
```

Use Agentforce to Invoke External Service Actions

Add capabilities to agents with customizable actions configured via External Services. Use External Services to declaratively connect your Salesforce org to an external API. After you register an OpenAPI schema, the actions defined in that schema become automatically available as custom agent actions.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

1. Register an external service.

See [Submit an API Spec With a Relative URL](#), [Submit an API Spec With an Absolute URL](#), [Paste the JSON or YAML Formatted Schema Into Your Browser](#), or [Upload a Local File](#).

Select the operations that you want to import into your external service registration. If you already have an external service registered, those actions automatically appear as custom agent actions.

The screenshot shows a table titled 'Select Operations' with three selected out of three available operations. The columns are 'Operation ↑', 'Description', 'Input parameters', and 'Output parameters'. The operations listed are:

Operation ↑	Description	Input parameters	Output parameters
<input checked="" type="checkbox"/>	CheckBackground Return the background status for a given employee email	body	500, 200, responseCode, default
<input checked="" type="checkbox"/>	GenerateOneTimeToken Obtain a one-time token to be used in CheckBackground	body	500, responseCode, 404, 400, default, 200
<input checked="" type="checkbox"/>	ReviewResults Review the result of the program	body	responseCode, 200, 500, default

At the bottom, there are 'Back' and 'Next' buttons, with the 'Next' button being blue and highlighted.

2. Create an Agent Action.

See [Create a Custom Agent Action](#).

When you create the agent action, you select **External Services** for the External Action Category. The external service operations are available in the Reference Action field.

Create an Agent Action

Connect an existing action

Actions are how an agent gets things done. To create an agent action, start with the functionality you already have in Salesforce, such as flows or prompt templates. Select an action you want the agent action to reference, and then configure it for use with an agent.

* Reference Action Type
API

* Reference Action Category
External Services

* Reference Action
Select an action...

Check Background

Generate One Time Token

Review Results

3. Associate the action with a topic

See [Add an Action to a Topic](#).

Use Flow to Invoke External Service Actions

In this Flow example, design and test the automation that sends a user's information from Salesforce to the external employee banking system. Create the variables for user phone numbers, and combine different phone numbers into one piece of data. Then use the external service action to create the user.

[End-to-end Example with Flow](#)

In this Flow example, design and test the automation that sends a user's information from Salesforce to the external employee banking system. Create the variables for user phone numbers, and combine different phone numbers into one piece of data. Then use the external service action to create the user.

[View an Action's Unique Apex Name in Flow](#)

Map an action in Flow to an action in your external service. When searching for an External Service action in Flow Builder, you see the unique Apex name in the list of actions.

See Also

[External Services](#)

[Build a Flow](#)

End-to-end Example with Flow

In this Flow example, design and test the automation that sends a user's information from Salesforce to the external employee banking system. Create the variables for user phone numbers, and combine different phone numbers into one piece of data. Then use the external service action to create the user.

1. Define a named credential

For this simplified example, use "legacy" Named Credentials. Instead of clicking New, select **New Legacy** from the dropdown menu next to the New button. For the named credential that your org uses to access the banking system, assign the *Bank* label. Assign a placeholder URL, such as `https://api.example.com`. Use `example.com` because you *paste in* the schema at registration time, instead of using a URL to point to an API spec.

In the case where you use a URL, the named credential URL corresponds to the declared host of the API spec and one of its transfer protocol schemes or URL protocols. The URL can point to a different endpoint as long as it hosts the same external service. The base path is added to the named credential URL on callout.

2. Register the employee banking system's external service, using the steps described in [Register an External Service](#) as a guide. Use the *BankService* name and the *Bank* named credential, and then copy the schema from the "External Services API Spec Example 2", found in the topic [External Services OpenAPI 2.0 Schema](#).

3. Click **New Flow**.

4. Select **Start From Scratch** and click **Next**.

5. Then, select **Screen Flow** and click **Create**.

This example uses static values for simplicity. In contrast, a production flow with user data includes elements to get user records, store values to variables, and communicate these values to the external service.

6. To handle the multiple phone numbers nested below the `BankService__Phone` value, add the two variable resources.

- a. Click **Manager**, and then click **New Resource**.
- b. For the resource type, select **Variable**.
- c. For the API name, enter `WorkPhone`.
- d. For the data type, select **Apex-Defined**.
- e. For the Apex class, select `ExternalService__BankService__Phone`.

New Resource

* Resource Type
Variable

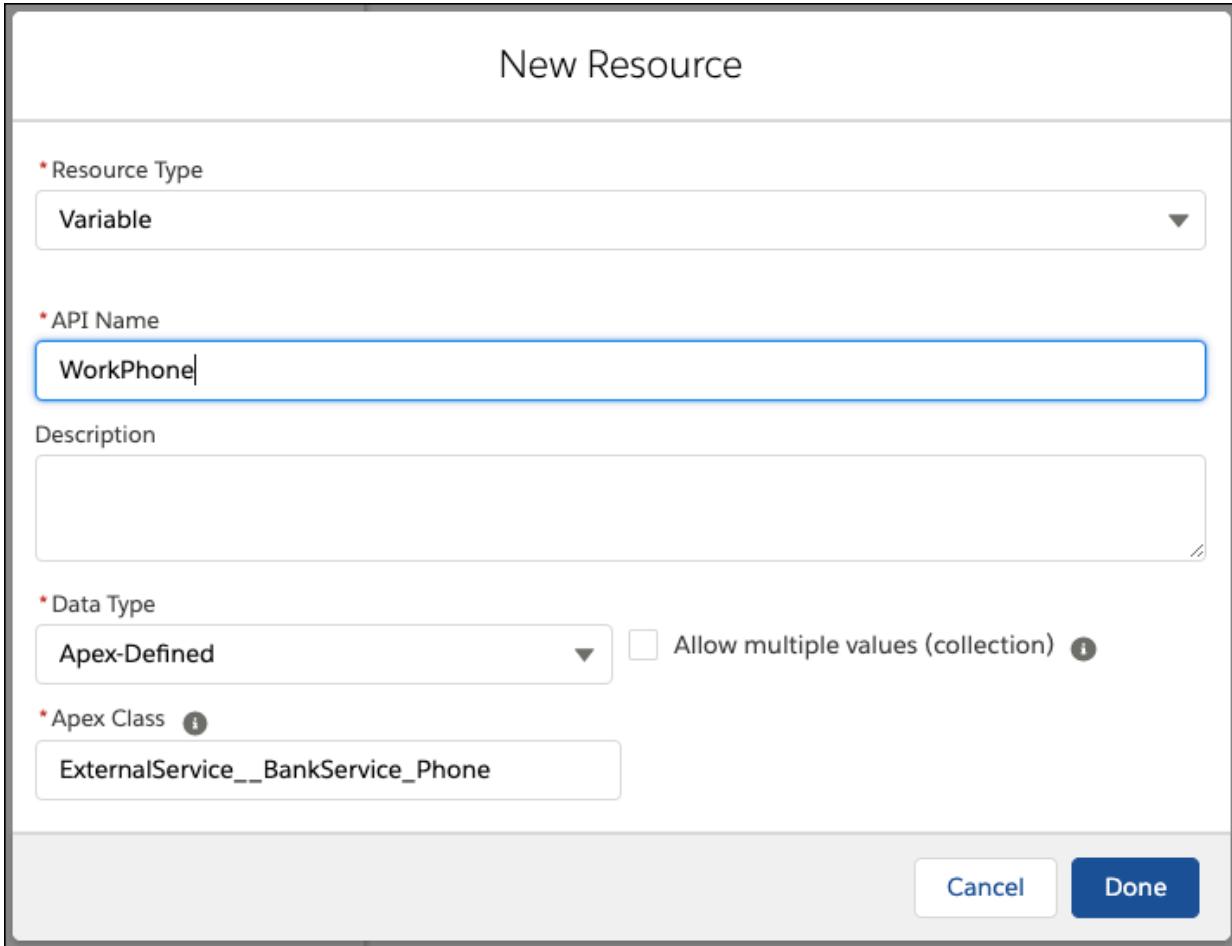
* API Name
WorkPhone

Description

* Data Type
Apex-Defined Allow multiple values (collection) i

* Apex Class i
ExternalService__BankService__Phone

Cancel Done



- f. Click **Done**.
 - g. Repeat these steps using the API name *CellPhone*.
7. Add the variable resource that acts as the array for the phone number values.
- a. Click **New Resource**.
 - b. For the resource type, select **Variable**.
 - c. For the API name, enter *Phones*.
 - d. For the data type, select **Apex-Defined**, and select the **Allow multiple values (collection)** option.
 - e. For the Apex class, select *ExternalService__BankService__Phone*.

New Resource

* Resource Type
Variable

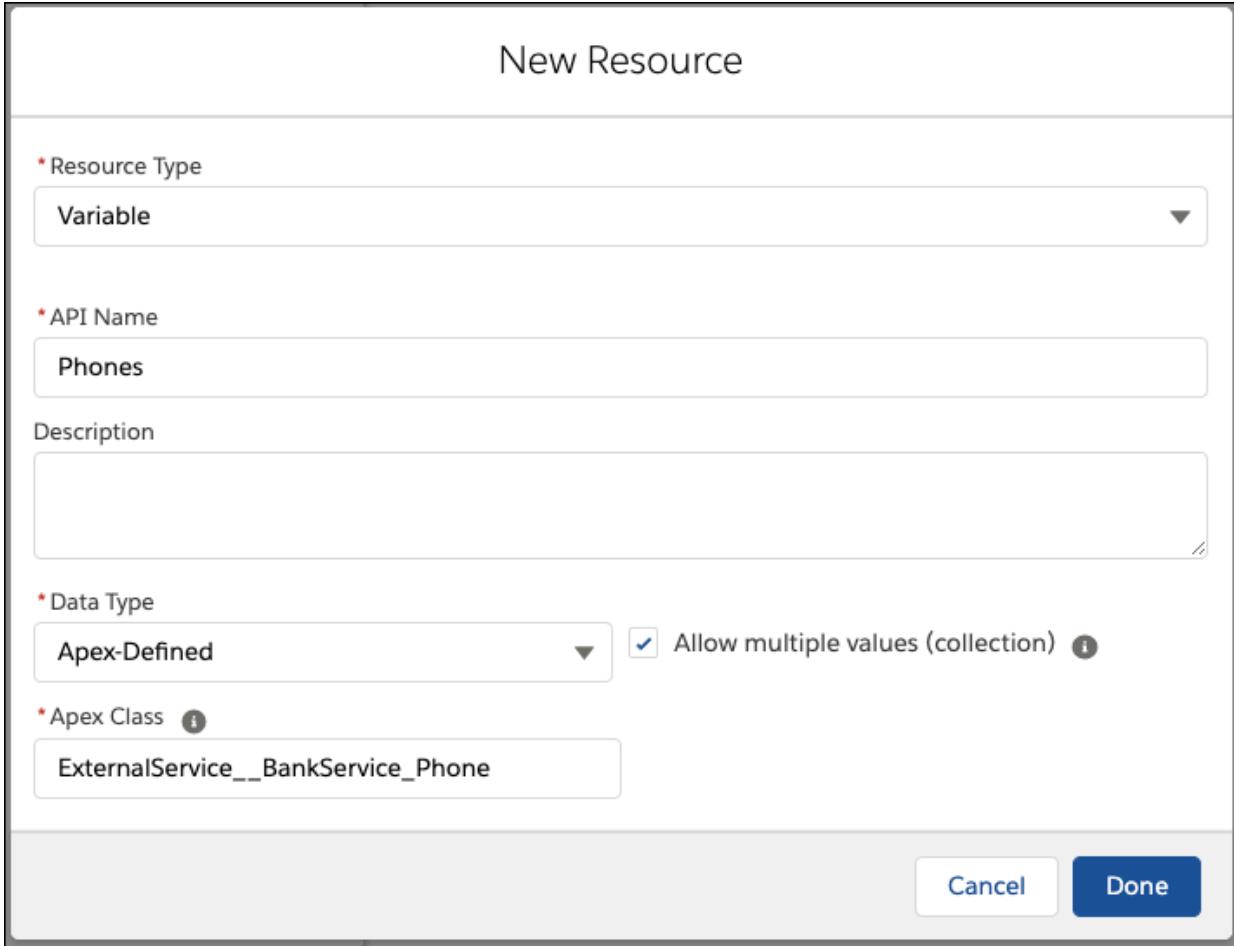
* API Name
Phones

Description

* Data Type
Apex-Defined Allow multiple values (collection) i

* Apex Class i
ExternalService__BankService__Phone

Cancel **Done**



- f. Click **Done**.
8. Add the variable resource that stores the user's name.
 - a. Click **New Resource**.
 - b. For the resource type, select **Variable**.
 - c. For the API name, enter *User*.
 - d. For the data type, select **Apex-Defined**.
 - e. For the Apex class, select `ExternalService__BankService__User`.
 - f. Click **Done**.
9. Assign values to the work phone variable.
 - a. From the Toolbox, click **Elements**.
 - b. Drag an **Assignment** element to the canvas.
 - c. For Label, enter *Assign Work Phone*, and let the API name autopopulate.
 - d. Click **Search variables**, select **WorkPhone**, then select **phone**. The variable `{ !WorkPhone.phone }` appears.
 - e. For Value, enter *1234567890*.
 - f. Click **Add Assignment**.
 - g. Click **Search variables**, select **WorkPhone**, then select **typeofphone**. The variable `{ !WorkPhone.typeofphone }` appears.
 - h. For Value, enter *Work*.

New Assignment

* Label	* API Name	
Assign Work Phone	Assign_Work_Phone	
Description		
Set Variable Values Each variable is modified by the operator and value combination.		
Variable	Operator	Value
{!WorkPhone.phone}	Equals	1234567890
Variable	Operator	Value
{!WorkPhone.typeofphone}	Equals	Work
+ Add Assignment		
		Cancel Done

- i. Click **Done**.
 - j. Connect the Start element to this assignment element.
10. Assign values to the cell phone variable.
- a. Drag an **Assignment** element to the canvas.
 - b. For Label, enter *Assign Cell Phone*, and let the API name autopopulate.
 - c. Click **Search variables**, select **CellPhone**, then select **phone**. The variable `{ !CellPhone.phone }` appears.
 - d. For Value, enter *0987654321*
 - e. Click **Add Assignment**.
 - f. Click **Search variables**, select **CellPhone**, then select **typeofphone**. The variable `{ !CellPhone.typeofphone }` appears.
 - g. For Value enter *Cell*.
 - h. Click **Done**.
 - i. Connect the last element to this one.
11. Assign multiple phone values to the single phone array variable.
- a. Drag an **Assignment** element to the canvas.
 - b. For Label, enter *Assign Phones*, and let the API name autopopulate.
 - c. Click **Search variables** then select **Phones**. The variable `{ !Phones }` appears.
 - d. Change the **Operator** to **Add**.
 - e. For Value, select **Workphone**.
 - f. Click **Add Assignment**.

- g. Click **Search variables** then select **Phones** again.
- h. Change the Operator to **Add**.
- i. For Value select **Cellphone**.

New Assignment

* Label	* API Name	
Assign Phones	Assign_Phones	
Description		
<pre>if(Phone != null) { if(Phone.WorkPhone != null) { WorkPhone = Phone.WorkPhone; } if(Phone.CellPhone != null) { CellPhone = Phone.CellPhone; } }</pre>		
Set Variable Values		
Each variable is modified by the operator and value combination.		
Variable	Operator	Value
{!Phones}	Add	{!WorkPhone}
Variable	Operator	Value
{!Phones}	Add	{!CellPhone}
+ Add Assignment		
		Cancel Done

- j. Click **Done**.
 - k. Connect the last element to this one.
12. Assign the user values.
- a. Drag an **Assignment** element to the canvas.
 - b. For Label, enter *Assign User*, and let the API name autopopulate.
 - c. Click **Search variables**, select **User**, then **id**. The variable `{ !User.id }` appears.
 - d. For Value, enter *1234*.
 - e. Click **Add Assignment**.
 - f. Click **Search variables**, select **User**, then **name**. The variable `{ !User.name }` appears.
 - g. For Value, enter *Maria*.
 - h. Click **Add Assignment**.
 - i. Click **Search variables**, select **User**, then **phones**. The variable `{ !User.phones }` appears.
 - j. For Value, select **Phones**.

New Assignment

* Label	* API Name
<input type="text" value="Assign User"/>	<input type="text" value="Assign_User"/>
Description	
<div style="border: 1px solid #ccc; height: 80px; width: 100%;"></div>	

Set Variable Values

Each variable is modified by the operator and value combination.

Variable	Operator	Value	
<input type="text" value="{!!User.id}"/>	<input style="width: 100%; height: 100%;" type="button" value="Equals"/>	<input type="text" value="1234"/>	<input style="width: 100%; height: 100%;" type="button" value="Delete"/>
<input type="text" value="{!!User.name}"/>	<input style="width: 100%; height: 100%;" type="button" value="Equals"/>	<input type="text" value="Maria"/>	<input style="width: 100%; height: 100%;" type="button" value="Delete"/>
<input type="text" value="{!!User.phones}"/>	<input style="width: 100%; height: 100%;" type="button" value="Equals"/>	<input type="text" value="{!!Phones}"/>	<input style="width: 100%; height: 100%;" type="button" value="Delete"/>

+ Add Assignment

- k. Click **Done**.
 - l. Connect the last element to this one.
13. To create users on the external bank system, add the action generated by your external service schema.
- a. Drag an **Action** element to the canvas.
 - b. Filter the actions by type, and select **External Service**.
 - c. Select **postUser**.
 - d. For Label, enter *Create a new user*, and let the API name autopopulate.
 - e. Next to **>_user**, toggle **Included** to include input values.
 - f. From the lookup, select **User**.

New Action

Filter By ▾

postUsers

Core Action

Apex Action

Apex Action (Legacy)

Email Alert

External Service

Use values from earlier in the flow to set the inputs for the "postUsers" {1}. To use its outputs later in the flow, store them in variables.

* Label

* API Name

Description

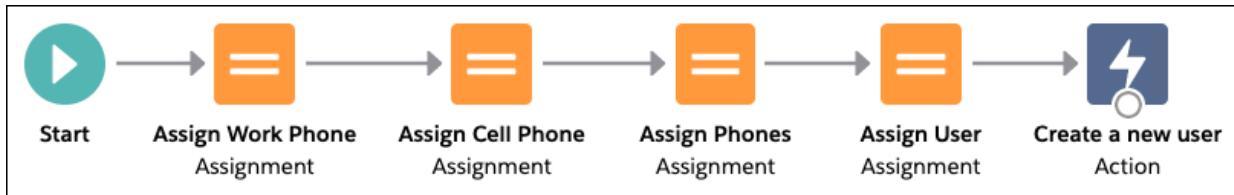
Set Input Values

>_ user

Include

g. Click Done.

h. Connect the last element to this one. Your canvas includes these elements.



14. Save and debug the flow. A successful debug includes the assignment and External Service callout.

ASSIGNMENT: Assign_User

```
{!User.id} Equals 1,234
{!User.name} Equals Maria
{!User.phones} Equals {!Phones}
```

Result

```
{!User.phones} = "[BankService_Phone : {
  "typeofphone" : "Work", "phone" :
  "1234567890" },BankService_Phone : {
  "typeofphone" : "Cell", "phone" : "0987654321"
}]"
{!User.name} = "Maria"
{!User.id} = "1,234"
```

BANKSERVICE.POSTUSERS (EXTERNAL**SERVICES): Create_a_new_user****Inputs:**

```
user = {!User} (BankService_User : { "phones" : [ {
  "typeofphone" : "Work", "phone" :
  "1234567890" }, { "typeofphone" : "Cell",
  "phone" : "0987654321" } ], "name" : "Maria",
  "id" : 1234 })
```

Outputs:

None.

View an Action's Unique Apex Name in Flow

Map an action in Flow to an action in your external service. When searching for an External Service action in Flow Builder, you see the unique Apex name in the list of actions.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

When you add a resource for a new action, you can select the correct action from the correct registration by virtue of its automatically generated Apex name.



Invoke External Service Callouts Using Apex

You can call external service registrations natively from Apex. Make a callout to an external service like the Apex `Http Class` without the need to write boilerplate code. The registered services are strongly typed in Apex with the registration's schema as Apex types. These Apex types reflect your registered service's specification, making the Apex compiler do the heavy lifting for you.

[External Service Registrations in Apex](#)

See how registered external service schema types map to Apex types under the namespace `ExternalService`.

[Use Apex to Create a Callout to an External Service](#)

Design and test the automation that sends a user's information from Salesforce to the external employee banking system. Using the examples provided, create the variables for user phone numbers, and combine different phone numbers into one piece of data. Then use the external service action to create the user.

[Asynchronous Callback Operations Using Apex](#)

Use Apex to create, test, and monitor asynchronous callbacks. Callbacks are right for integrations that require potentially delayed responses of more than 120 seconds from an external source. For example, a mortgage application API, a shipping notification API, or a payment confirmation API.

[View Apex Names in Apex Class Viewer](#)

View all External Services auto-generated Apex classes in the Apex Class viewer in Setup.

External Service Registrations in Apex

See how registered external service schema types map to Apex types under the namespace `ExternalService`.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

The example Apex types in this section refer to the [Example 1: Basic OpenAPI Spec with Request and Response \(OAS 2.0\)](#) with a registered name "CreditScore".

External Service

The external service registration with name `ServiceName` is the Apex class `ExternalService.ServiceName` making the external service callable in Apex. To call operations on the external service in Apex, instantiate an instance with the default constructor.

For example, create an instance of `CreditScore` to make callouts:

```
ExternalService.CreditScore creditScore = new ExternalService.CreditScore();
```

External Service Operation

A registered external service's derived operation name is declared as an Apex method on the Apex external service class.

- The operation input parameters are declared in the method's request class in the order of declaration in the external service's specification and the corresponding parameter data type in Apex.
- The operation output parameters with successful HTTP response codes are declared in the method's return response class. The API specification's HTTP response codes are declared in the response class. The response class' property `ResponseCode` captures the callout's HTTP response code value.
- The operation output parameters with HTTP error response codes are declared in the method's exception class for HTTP error codes. The API specification's HTTP error response codes are declared in the response exception class. The property `ResponseCode` captures the callout's HTTP error response code value. The property `DefaultResponse` the operation's default response message value for non declared HTTP response codes.

For example, the CreditScore's operation `getAccountLastCreditRating` is declared as Apex method on the Apex class `ExternalService.CreditScore` with corresponding input arguments, return response type and exception object:

```
global class CreditScore {  
    global getAccountLastCreditRating_Response getAccountLastCreditRating(  
        getAccountLastCreditRating_Request input) {...}  
  
    global class getAccountLastCreditRating_Request {  
        global accountId accountId {get; set;}  
    }  
  
    global class getAccountLastCreditRating_Response {  
        global Integer ResponseCode {get; set;}  
        CreditScore_creditRating Code200 {get; set;}  
    }  
  
    global class getAccountLastCreditRating_ResponseException {  
        global Integer ResponseCode {get; set;}  
        global String DefaultResponse {get; set;}  
        global String Code405 {get; set;}  
    }  
}
```

In the example, the method `getAccountLastCreditRating` throws exception

`getAccountLastCreditRating_ResponseException` that you can catch to process HTTP error codes.

To get the account's last credit rating in Apex:

```
public class CreditScoreRater {  
    // Application domain exception  
    public class CreditScoreException extends Exception {}  
  
    public String rateCreditScore(String accountIdentifier) {  
        ExternalService.CreditScore creditScore = new ExternalService.CreditScore();  
  
        // Set the account ID  
        ExternalService.CreditScore_accountId accountId =  
            new ExternalService.CreditScore_accountId();  
        accountId.accountIdString = accountIdentifier;  
  
        // Call to get the last credit rating  
        try {  
            // Construct the request  
            ExternalService.CreditScore.getAccountLastCreditRating_Request request =  
                new ExternalService.CreditScore.getAccountLastCreditRating_Request();  
            request.body = accountId;  
  
            // Call the external credit rating service  
            ExternalService.CreditScore.getAccountLastCreditRating_Response response =  
                creditScore.getAccountLastCreditRating(request);  
  
            // Get the credit rating response for HTTP status code 200  
            ExternalService.CreditScore_creditRating creditRating = response.Code200;  
  
            // The rating  
            return creditRating.creditRatingString;  
  
            // Handle callout error and translate to application domain exception  
        } catch (ExternalService.CreditScore.getAccountLastCreditRating_ResponseException e) {  
            // Invalid input is flagged with status code 405  
            if (e.ResponseCode == 405) {  
                throw new CreditScoreException('Invalid input for account: ' +
```

```
        accountId.accountIdString);
    }

    // Handle generic callout error - for example internal server error code 500
    throw new CreditScoreException('Unknown error: ' + e.ResponseCode
+ ': '
+ e.DefaultResponse);
}

}
```

External Service Data Types

OpenAPI schema data types are mapped to their corresponding Apex types as follows:

OpenAPI 2/3 Data Types (type, format)	Apex Type
integer	Integer
integer, int32	Integer
integer, int64	Long
number, float	Double
number, double	Double
string	String
string, byte	Blob
string, binary	Blob
string, date	Date
string, date-time	Datetime
boolean	Boolean
array	List<>
object	Apex Property Class, Map<>

Array

The OpenAPI type array is mapped as Apex `List<ElementType>`. The items type is mapped to `ElementType`.

Object

The OpenAPI type `object` is mapped as a named Apex class where the Apex class properties match the OpenAPI `properties` by property name and type. `additionalProperties` are mapped as Apex `Map<String, ValueType>`. The `additionalProperties` name is the map key, the `additionalProperties` type is the map's `ValueType`.

Object Naming

OpenAPI schema object types defined as a named reference with `definitions` in OpenAPI 2 or `schema components` in OpenAPI 3 map to the Apex Property Class type:

`ExternalService.<Service Name>_<Reference Name>`. For example, the credit score's `creditRating` object schema is `ExternalService.CreditScore_creditRating`.

Schema object types anonymously declared inline instead of a named schema reference must be identifiable as an Apex property class by name with this naming scheme:

- Operation request body object schema:
 - `ExternalService.<Service Name>_<Operation Name>_IN`
- Operation response object schema with successful HTTP status code (< 300):
 - `ExternalService.<Service Name>_<Operation Name>_OUT_<Response Code>`
- Operation response object schema with error HTTP status code (>= 300):
 - `ExternalService.<Service Name>_<Operation Name>_EXC_<Response Code>`
- Object schema as property type of another schema object:
 - `ExternalService.<Service Name>_<Property's Object Name>_<Property Name>`
- Polymorphic object schema in allOf schema composition with discriminator:
 - `ExternalService.<Service Name>_<Composition Schema Reference Name>_KT_PT`

Array item type with anonymous object schema - Apex Properties Class for list ElementType:

- Element type for named schema referenced array:
 - `ExternalService.<Service Name>_VT_<Array Reference Name>`
- Element type for anonymous schema array follows the object schema naming scheme

Object's additionalProperties type with anonymous object schema - Apex Properties Class for map ValueType:

- Map value type for additionalProperties in named reference object schema:
 - `ExternalService.<Service Name>_KT_VN_<additionalProperties' Object Reference Name>`
- Map value type for additionalProperties in anonymous object schema:
 - `<additionalProperties' Anonymous Object Name>_KT_V`

For more object schema type examples, see: [Schema Examples](#).

Variable Naming

When Salesforce maps variable names from your schema into Apex variable names, certain characters are translated.

Schema Variable	Translates Into Apex As	Example
_(underscore)	x5f	account_type (schema) accountx5ftype (Apex)
- (hyphen)	x2d	account-status (schema) accountx2dstatus (Apex)
(reserved keyword)	prepended with z0	User (schema) z0User (Apex)

Use Apex to Create a Callout to an External Service

Design and test the automation that sends a user's information from Salesforce to the external employee banking system. Using the examples provided, create the variables for user phone numbers, and combine different phone numbers into one piece of data. Then use the external service action to create the user.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

1. Define a named credential

For this simplified example, use "legacy" Named Credentials. Instead of clicking New, select **New Legacy** from the dropdown menu next to the New button. For the named credential that your org uses to access the banking system, assign the *Bank* label. Assign a placeholder URL, such as <https://api.example.com>. Use *example.com* because you *paste in* the schema at registration time, instead of using a URL to point to an API spec.

In the case where you use a URL, the named credential URL corresponds to the declared host of the API spec and one of its transfer protocol schemes or URL protocols. The URL can point to a different endpoint as long as it hosts the same external service. The base path is added to the named credential URL on callout.

2. Register the employee banking system's external service, using the steps described in [Register an External Service](#) as a guide. Use the `BankService` name and the `Bank` named credential, and then copy the schema from the "External Services API Spec Example 2", found in the topic [External Services OpenAPI 2.0 Schema](#).
3. Create the Apex class `BankService`:

```
public class BankService {  
}
```

4. To store the user details, add a member variable:

```
public class BankService {  
    private ExternalService.BankService_z0User user;  
}
```

5. To initialize an empty user with an empty list of phones, add a method:

```
public class BankService {  
    private ExternalService.BankService_z0User user;  
  
    private void setEmptyUser() {  
        if (this.user == null) {  
            this.user = new ExternalService.BankService_z0User();  
            this.user.phones = new List<ExternalService.BankService_Phone>();  
        }  
    }  
}
```

6. To assign the `work` phone value, add a method:

```
public class BankService {  
    private ExternalService.BankService_z0User user;  
  
    public void setWorkPhone(String phoneNumber) {  
        setEmptyUser();  
        ExternalService.BankService_Phone phone =  
            new ExternalService.BankService_Phone();  
        phone.phone = phoneNumber;  
        phone.typeofphone = 'Work';  
        this.user.phones.add(phone);  
    }  
  
    private void setEmptyUser() {
```

```

        if (this.user == null) {
            this.user = new ExternalService.BankService_z0User();
            this.user.phones = new List<ExternalService.BankService_Phon
e>();
        }
    }
}

```

7. To assign the *cell* phone value, add a method:

```

public class BankService {
    private ExternalService.BankService_z0User user;

    public void setWorkPhone(String phoneNumber) {
        setEmptyUser();
        ExternalService.BankService_Phone phone =
            new ExternalService.BankService_Phone();
        phone.phone = phoneNumber;
        phone.typeofphone = 'Work';
        this.user.phones.add(phone);
    }

    public void setCellPhone(String phoneNumber) {
        setEmptyUser();
        ExternalService.BankService_Phone phone =
            new ExternalService.BankService_Phone();
        phone.phone = phoneNumber;
        phone.typeofphone = 'Cell';
        this.user.phones.add(phone);
    }

    private void setEmptyUser() {
        if (this.user == null) {
            this.user = new ExternalService.BankService_z0User();
            this.user.phones = new List<ExternalService.BankService_Phon
e>();
        }
    }
}

```

8. To assign a user, add a method:

```

public class BankService {
    private ExternalService.BankService_z0User user;
}

```

```
public void setUser(Integer userId, String userName) {
    setEmptyUser();
    this.user.id = userId;
    this.user.name = userName;
}

public void setWorkPhone(String phoneNumber) {
    setEmptyUser();
    ExternalService.BankService_Phone phone =
        new ExternalService.BankService_Phone();
    phone.phone = phoneNumber;
    phone.typeofphone = 'Work';
    this.user.phones.add(phone);
}

public void setCellPhone(String phoneNumber) {
    setEmptyUser();
    ExternalService.BankService_Phone phone =
        new ExternalService.BankService_Phone();
    phone.phone = phoneNumber;
    phone.typeofphone = 'Cell';
    this.user.phones.add(phone);
}

private void setEmptyUser() {
    if (this.user == null) {
        this.user = new ExternalService.BankService_z0User();
        this.user.phones = new List<ExternalService.BankService_Phon
e>();
    }
}
```

9. Create a custom exception class, since you can't throw built-in Apex exceptions.

```
Public class BankServiceException extends Exception{}
```

For more information, see [Create Custom Exceptions](#) in *Apex Developer Guide*.

10. To create users on the external bank system, add a method to call the external BankService:

```
public class BankService {
    private ExternalService.BankService_z0User user;
```

```
public void createUser() {
    if (this.user == null || this.user.id == null) {
        throw new BankServiceException('Set the user to create first');
    }

    ExternalService.BankService bankService = new ExternalService.BankService();
    try {
        ExternalService.BankService.postUsers_Request request = new ExternalService.BankService.postUsers_Request();
        request.z0User = this.user;
        bankService.postUsers(request);
    } catch (ExternalService.BankService.postUsers_ResponseException e) {
        throw new BankServiceException('Couldn\'t create user with ID: ' + this.user.id);
    }
}

public void setUser(Integer userId, String userName) {
    setEmptyUser();
    this.user.id = userId;
    this.user.name = userName;
}

public void setWorkPhone(String phoneNumber) {
    setEmptyUser();
    ExternalService.BankService_Phone phone =
        new ExternalService.BankService_Phone();
    phone.phone = phoneNumber;
    phone.typeofphone = 'Work';
    this.user.phones.add(phone);
}

public void setCellPhone(String phoneNumber) {
    setEmptyUser();
    ExternalService.BankService_Phone phone =
        new ExternalService.BankService_Phone();
    phone.phone = phoneNumber;
    phone.typeofphone = 'Cell';
    this.user.phones.add(phone);
}

private void setEmptyUser() {
    if (this.user == null) {
        this.user = new ExternalService.BankService_z0User();
        this.user.phones = new List<ExternalService.BankService_Phon
```

```
    e>() ;  
}  
}  
}
```

11. Open the developer console and debug by calling the `BankService` with some sample values with a code snippet in an anonymous execution class. Have the debug logs open and compare with this sample callout log:

```
BankService bankService = new BankService();  
bankService.setWorkPhone('1234567890');  
bankService.setCellPhone('0987654321');  
bankService.setUser(1234, 'Maria');  
bankService.createUser();
```

Asynchronous Callback Operations Using Apex

Use Apex to create, test, and monitor asynchronous callbacks. Callbacks are right for integrations that require potentially delayed responses of more than 120 seconds from an external source. For example, a mortgage application API, a shipping notification API, or a payment confirmation API.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

 **Note** External Services' asynchronous callback operations are supported in Apex but not in Flow.

How External Services Asynchronous Callbacks Work

External Services asynchronous operations are described in an OpenAPI 3.x compliant specification with a callback operation. Asynchronous operations are registered by the system as a special type of invokable action that allows for longer response times. In contrast, External Services synchronous operations time out after 120 seconds. With asynchronous operations, you use Apex to define the callback and the time out for the delayed asynchronous response.

Validate Support for Callback URL Expressions

Verify that your callback URL expression is supported and registered by External Services as an asynchronous action.

Use Apex to Create an Asynchronous Callout to an External Service

When you register a schema containing a callback, External Services creates an invocable Apex operation with an automatically generated Apex class. Salesforce creates a callback URL on the asynchronous callout (initial callout) that's read-only. Create an Apex client that's capable of handling the callback by using the generated Apex interfaces. The client waits for an asynchronous response from the external system for an extended time (up to twenty-four hours).

Edit an Asynchronous Callout Class Definition

To edit an External Services asynchronous callback class, a related background operation job can't be running.

Create Unit Testing for Asynchronous Callouts

You can use the `ExternalServiceTest` method to mock callback and asynchronous responses.

Monitor and Debug Asynchronous Callouts

Asynchronous callouts are monitored as jobs using the Background Operations app or with Apex log lines in the Developer Console. To debug your code at runtime, use Apex log lines.

How External Services Asynchronous Callbacks Work

External Services asynchronous operations are described in an OpenAPI 3.x compliant specification with a callback operation. Asynchronous operations are registered by the system as a special type of invokable action that allows for longer response times. In contrast, External Services synchronous operations time out after 120 seconds. With asynchronous operations, you use Apex to define the callback and the time out for the delayed asynchronous response.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

 **Note** External Services' asynchronous callback operations are supported in Apex but not in Flow.

Example API Specification With Callback Operation

This example API specification features a callback operation within a mortgage application process for the fictitious company "Acme Mortgages". Callback-related definitions are in bold. We'll return to this example in other topics in this section.

```
{  
  "openapi": "3.0.0",  
  "servers": [{  
    "url": "/"  
  }],  
  "info": {  
    "version": "1.0",  
    "title": "Acme Mortgages",  
    "description": "Acme Mortgages"  
  },  
  "paths": {  
    "/applications": {  
      "post": {  
        "responses": {  
          "200": {  
            "description": "Success",  
            "content": {  
              "application/json": {  
                "schema": {  
                  "$ref": "#/components/schemas/Application"  
                }  
              }  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

```
"operationId": "SubmitApplication",
"description": "Submit a new mortgage application",
"parameters": [
    {
        "name": "callbackUrlForErrorCases",
        "in": "query",
        "schema": {
            "type": "string"
        }
    },
    "requestBody": {
        "content": {
            "application/json": {
                "schema": {
                    "type": "object",
                    "properties": {
                        "applicant": {
                            "$ref": "#/components/schemas/Contact"
                        }
                    }
                }
            }
        }
    }
],
"responses": {
    "201": {
        "description": "Mortgage loan application submission initial response",
        "content": {
            "application/json": {
                "schema": {
                    "type": "object",
                    "properties": {
                        "applicationNumber": {

```

```
        "type": "string"
    }
}
}
}
},
"callbacks": {
    "applicationOutcomeApproved": {
        "$ref": "#/components/callbacks/ApplicationApproved"
    },
    "applicationOutcomeRejected": {
        "$ref": "#/components/callbacks/ApplicationRejected"
    },
    "applicationError": {
        "${request.query.callbackUrlForErrorCases)": {
            "post": {
                "parameters": [
                    {
                        "in": "header",
                        "name": "applicationNumber",
                        "schema": {
                            "type": "string"
                        }
                    }
                ],
                "requestBody": {
                    "content": {
                        "application/json": {
                            "schema": {
                                "$ref": "#/components/schemas/MortgageApplicationErro
r"
                            }
                        }
                    }
                }
            },
            "responses": {
                "200": {
                    "description": "Mortgage application callback error accept
ed"
                }
            }
        }
    }
}
```

```
        }
    }
},
"components": {
  "schemas": {
    "MortgageApplication": {
      "required": [
        "applicationNumber",
        "status"
      ],
      "properties": {
        "applicationNumber": {
          "type": "string"
        },
        "status": {
          "description": "One of pending, approved, rejected",
          "type": "string"
        },
        "approvedAmount": {
          "type": "number"
        }
      }
    },
    "Contact": {
      "type": "object",
      "required": [
        "name"
      ],
      "properties": {
        "name": {
          "type": "string"
        },
        "address": {
          "type": "string"
        }
      }
    },
    "MortgageApplicationError": {
      "properties": {
        "errorMessage": {
          "type": "string"
        },
        "applicationNumber": {

```

```
        "type": "string"
    }
}
}
},
"callbacks": {
    "ApplicationApproved": {
        "${request.body#/callbackUrlForOutcomes/approved)": {
            "post": {
                "description": "Application has been approved.",
                "operationId": "approvedCallback",
                "requestBody": {
                    "content": {
                        "application/json": {
                            "schema": {
                                "$ref": "#/components/schemas/MortgageApplication"
                            }
                        }
                    }
                },
                "responses": {
                    "200": {
                        "description": "Approved application callback result has been retrieved successfully."
                    }
                }
            }
        }
    },
    "ApplicationRejected": {
        "${request.body#/callbackUrlForOutcomes/rejected)": {
            "post": {
                "description": "Application is rejected",
                "requestBody": {
                    "content": {
                        "application/json": {
                            "schema": {
                                "$ref": "#/components/schemas/MortgageApplicationError"
                            }
                        }
                    }
                }
            }
        },
        "responses": {
    
```

Example Apex Interface With Asynchronous Operations

When registered, the `AcmeMortgages` external service is automatically rendered in Apex by External Services. The Apex interface for `AcmeMortgages` appears in the Apex Classes page in Setup as shown here. Additional content that defines the asynchronous operations is in bold. Not shown are the dynamic Apex objects that represent schema object data types that are the same as invocable actions.

Each callback is owned by its parent asynchronous operation. A unique, read-only callback URL is determined by Salesforce (on the first callout).

```
global class AcmeMortgages {
    // Acme Mortgages SubmitApplication asynchronous operation
    global SubmitApplication_Response SubmitApplication(
        SubmitApplication_Request input,
        SubmitApplication_Callback callback,
        DateTime callbackTimeout
    ) {...}

    global class SubmitApplication_Request {
        global AcmeMortgages_SubmitApplication_IN_body body {get; set;}

        // [REQUEST] Callback URL for asynchronous operation: callbackUrlForErrorCases
        global String callbackUrlForErrorCases {get;}
    }

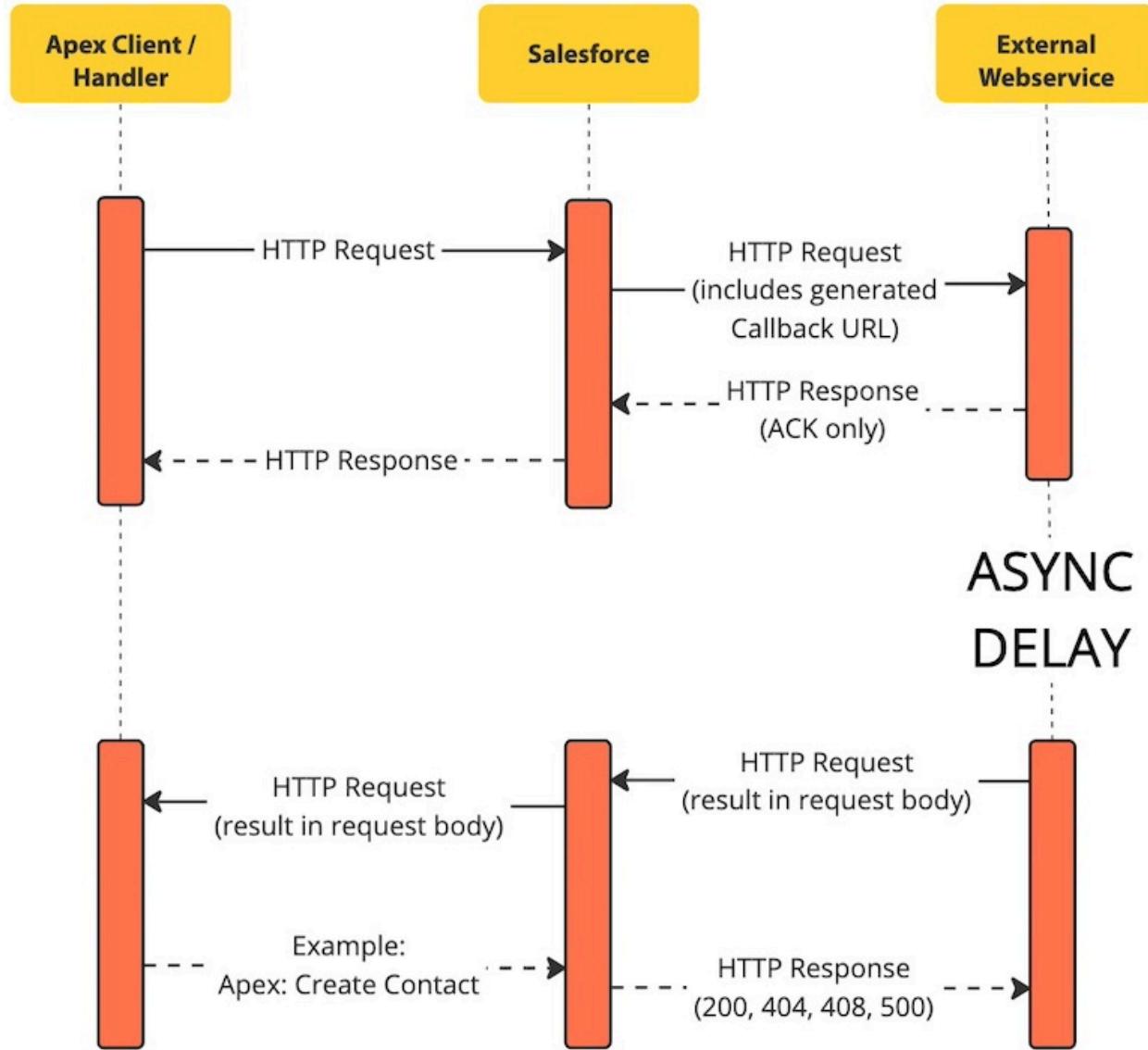
    // Synchronous response - here getting the application docket number
    global class SubmitApplication_Response {
        global Integer responseCode {get; set;}
        global AcmeMortgages_SubmitApplication_OUT_201 Code201 {get; set;}
        // Get invocation ID for this asynchronous response to query callback
    }
}
```

```
tatus  
    global String invocationId {get;}  
}  
  
global class SubmitApplication_ResponseException {  
    global Integer responseCode {get; set;}  
    global String defaultResponse {get; set;}  
}  
  
// Asynchronous response callback handler for SubmitApplication.  
// Each method corresponds to the schema's declared callback key.  
// Default implementations throws an exception alerting that a non-implemented callback wa  
s called.  
global virtual class SubmitApplication_Callback {  
    global virtual void applicationOutcomeApproved(  
        List<SubmitApplication_applicationOutcomeApproved_Callback> callbacks) {  
        throw new SubmitApplication_ResponseException(404,  
            'Callback not handled: applicationOutcomeApproved');  
    }  
    global virtual void applicationOutcomeRejected(List<SubmitApplication_applicationOut  
comeRejected_Callback> callbacks) {...}  
    global virtual void applicationError(List<SubmitApplication_applicationError_Callback>  
        callbacks) {...}  
}  
  
// Asynchronous callback response payloads for callback applicationOutcomeApproved  
global class SubmitApplication_applicationOutcomeApproved_Callback {  
    global SubmitApplication_Request request {get; set;}  
    global DateTime submitTime {get; set;}  
    global DateTime callbackTimeout {get; set;}  
    global CallbackStatus callbackStatus {get; set;}  
    global SubmitApplication_applicationOutcomeApproved_CallbackResponse response {g  
et; set;}  
}  
  
global class SubmitApplication_applicationOutcomeApproved_CallbackResponse {  
    global AcmeMortgages_MortgageApplication body {get; set;}  
}  
  
global class SubmitApplication_applicationOutcomeRejected_Callback {...}  
  
global class SubmitApplication_applicationOutcomeRejected_CallbackResponse {...}
```

```
global class SubmitApplication_applicationError_Callback {...}  
  
global class SubmitApplication_applicationError_CallbackResponse {...}  
}
```

Example Callback Data Flow

-  **Example** In this example, a Salesforce Apex developer uses External Services to register a bank's mortgage application API specification. The spec contains callback operations, as delays of up to one day between application submission and acceptance are expected. The Apex developer writes an Apex client with a callback handler. The Apex client specifies that after Salesforce receives the delayed response from the external system, Salesforce uses the resultant list of names and other mortgage application information to automatically create Contacts. This sequence diagram demonstrates the functional data flow between the developer's Apex code ("Apex Client / Handler" in the diagram), Salesforce, and the bank's mortgage application API endpoint ("External Webservice" in the diagram). There are *two responses* from the external web service. The first is synchronous and is the typical, initial ACK response. The second is the asynchronous, delayed response (technically a request) that includes the result payload in the request body.



First, the Apex developer sends an HTTP request (a callout) to the bank server (an external service) with the Apex client. The external web service immediately sends back an HTTP response acknowledging receipt of the request. This first acknowledgment times out after 120 seconds. The Salesforce developer monitors the status of the request by using the Background Operations page or the Apex Debug log. The bank starts its internal process to ingest the mortgage application, approve or deny it, and eventually prepare a final result (in the diagram as "Async Delay"). Within one day, the bank sends the completed result back to the Apex developer as an HTTP request. The result data (approved amount, status, and so on) for the mortgage application is contained inside the body of the request. Salesforce receives the result and processes it according to the Apex callback handler implementation. Salesforce returns an HTTP response acknowledgment to the external web service. If Salesforce processes the data successfully—in this case, to create contacts—then it sends a response to the external webservice with a 200 status code and a success message. If Salesforce encounters an error, it sends a 404, 408, or a 500 status code to the external webservice with a generic error message. The Apex developer can see more

error message details by using the Background Operations page or with the Apex Debug log.

Validate Support for Callback URL Expressions

Verify that your callback URL expression is supported and registered by External Services as an asynchronous action.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

For supported callback expressions, see [External Services Asynchronous Callbacks](#).

1. From Setup, in the Quick Find box, enter *External Services*, and then select **External Services**.
2. Register the API specification that contains the callback expression.
3. Navigate to the Detail page of the external service that contains the callback operation.
 - If the callback expression is supported for a given operation, then the **Callback Parameter** column lists the callback parameter names.
 - If there's no value in the **Callback Parameters** column, then the callback expression isn't supported, or the operation doesn't contain a callback operation.

 **Note** If the callback isn't supported, External Services attempts to register the operation as a normal, synchronous operation with a 120-second timeout on all callouts.

Use Apex to Create an Asynchronous Callout to an External Service

When you register a schema containing a callback, External Services creates an invocable Apex operation with an automatically generated Apex class. Salesforce creates a callback URL on the asynchronous callout (initial callout) that's read-only. Create an Apex client that's capable of handling the callback by using the generated Apex interfaces. The client waits for an asynchronous response from the external system for an extended time (up to twenty-four hours).

 **Important** Connected apps creation is restricted as of Spring '26. You can continue to use existing connected apps during and after Spring '26. However, we recommend using [external client apps](#) instead. If you must continue creating connected apps, contact Salesforce Support. See [New connected apps can no longer be created in Spring '26](#) for more details.

Here's some prerequisite reading and configuration to do before you create your own asynchronous callout.

- Review the fictitious schema on which our examples of asynchronous callbacks are based. See [Example API Specification With Callback Operation](#).

- To authenticate and create requests [Define an External Credential and a Named Credential](#).
- To receive the asynchronous response, create a connected app as described in [Authorization Through Connected Apps and OAuth 2.0](#). Salesforce Platform callback endpoints are exposed with TLS in Connect API. The caller must authenticate to the Salesforce org by setting up the appropriate connected app.
- Register the API specification, using the normal workflow as described in [Register an External Service](#).

As a comparison, a *synchronous* callout typically looks like this:

```
ExternalService.AcmeMortgages acme = new ...();  
  
ExternalService.AcmeMortgages.GetApplication_Request request = new ...();  
request.applicationNumber = 'X202290';  
  
ExternalService.AcmeMortgages.GetApplication_Response response =  
    acme.GetApplication(request);  
  
ExternalService.AcmeMortgages_MortgageApplication application = response.Code2  
00;
```

To invoke an *asynchronous* callout, you must write your own Apex code to make the callout. Then, you must implement your own callback object and pass it into your asynchronous invocation method within your Apex code.

The callback handler must be stateless. Salesforce doesn't preserve state information when the response is called back.

Don't nest the callback class. The callback class should implement the callback interface directly. For example:

- Correct: class `MyCallback` extends `ES.ServiceName.OperationName_Callback {...}`
- Not Recommended: class `SomeIntermediateClass` extends
`ES.ServiceName.OperationName_Callback {...}` and then `MyCallback` extends
`SomeIntermediateClass {...}`

1. From Setup, in the Quick Find box, enter *Apex Classes* in the Quick Find box, and then select **Apex Classes**.
2. Find the Apex class that contains the callback interface you want to use.
3. Create a callback. Write an Apex callback object that implements the External Services generated callback interface by referencing the corresponding Apex class.
4. After you have a callback definition, pass this callback into your asynchronous invocation method within your Apex code.

Example

```
// The callback gets called with the mortgage application update
// and handles each outcome, for example application is approved, rejected, ...
global class MyMortgageApplicationCallback extends ExternalService.AcmeMortgage.SubmitApplication_Callback {
    // Application approved
    global override void applicationOutcomeApproved(
        List<ExternalService.AcmeMortgages.SubmitApplication_applicationOutcomeApproved_Callback> callbacks) {
        ExternalService.AcmeMortgages.SubmitApplication_applicationOutcomeApproved_Callback callback = callbacks.get(0);
        if (callback.callbackStatus == CallbackStatus.COMPLETED) {
            Double loanAmount = callback.response.body.approvedAmount;
            // Create a Contact based on the response ...
        }
    }

    // Application rejected
    global override void applicationOutcomeRejected(...) { ... }

    ...
}

public class MyMortgageApplicationProcess {
    public void submitApplication(...) {
        // Submit mortgage application process
        ExternalService.AcmeMortgages acme = new ...();

        ExternalService.AcmeMortgages.SubmitApplication_Request request =
new ...();
        ExternalService.AcmeMortgages.Contact applicant = new ...();
        applicant.Name = 'Joe Miller';
        applicant.Address = '555 Miller St, CA, Milltown';
        request.applicant = applicant;

        // Time out if no response after 1 day
DateTime timeout = DateTime.now().addDays(1);

        ExternalService.AcmeMortgages.SubmitApplication_Response response =
acme.SubmitApplication(request, new MyMortgageApplicationCallback(), timeout;

        // Get back application number immediately
    }
}
```

```
String appNr = response.Code201.applicationNumber;

    // Get invocationId from a successful initial response to check status of the background operation
    String invocationId = response.invocationId;
    ...
}

}
```

Edit an Asynchronous Callout Class Definition

To edit an External Services asynchronous callback class, a related background operation job can't be running.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

The Apex callback class is editable when:

- the job is completed; an async response is received and the callback executes successfully
- the job triggers an error when there is an initial response failure
- the job triggers an error when the asynchronous response can't be processed successfully
- the job is manually cancelled on the Background Operations page

If the background operation job is still running (not yet completed), and hasn't thrown an error, then editing or deleting the respective Apex callback handler class definition triggers an error message and the edits aren't saved. For example:

```
This External Service callback handler class has
background operation jobs running:
<background operation ids>.
Retry when there are no other jobs in progress.
```

Create Unit Testing for Asynchronous Callouts

You can use the `ExternalServiceTest` method to mock callback and asynchronous responses.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

The code examples in this topic are based on the fictitious schema in [Example API Specification With Callback Operation](#).

This snippet shows an example unit test for the Acme Mortgage service. To test external services with callbacks in Apex, the Acme Mortgage callout mock class mocks the submit-mortgage application request and asserts the expected request payload. The test class triggers the mocked external service's callback response through an instance of test External Service methods.

```
global class AcmeMortgageHttpCalloutMock implements HttpCalloutMock {  
    private String callbackUrl;  
  
    // Mock asynchronous response  
    public void callBack() {  
        HttpRequest request = new HttpRequest();  
        request.setEndpoint(callbackUrl);  
        request.setMethod('POST');  
        request.setHeader('Content-Type', 'application/json');  
        request.setBody('{{' +  
            '"applicationNumber": "900-X", ' +  
            '"status": "Approved", ' +  
            '"approvedAmount": 10000' +  
        '}}');  
  
        HttpResponse response = Test.getExternalService().sendCallback(request);  
        Assert.AreEqual(200, response.getStatusCode());  
    }  
  
    // Mock initial response  
    global HttpResponse respond(HttpRequest request) {  
        if (request.getMethod().equals('POST') && request.getEndpoint().equals('callout:AcmeMortgageCredential/applications')) {  
            return respondSubmitApplication(request);  
        }  
        ...  
        System.assert(false, 'Request method and/or endpoint not valid: ' + request.getMethod() + ' ' + request.getEndpoint());  
        return null;  
    }  
}
```

```
private HttpResponse respondSubmitApplication(HttpServletRequest request) {
    // Retrieve the callback URL from request
    ExternalService.AcmeMortgage_SubmitApplication_IN_Body body =
(ExternalService.AcmeMortgage_SubmitApplication_IN_Body)
                Json.deserialize(request.getBody(),
(ExternalService.AcmeMortgage_SubmitApplication_IN_Body.class));
    callbackUrl = body.callbackUrlForOutcomes.approved;

    HttpResponse response = new HttpResponse();
    response.setHeader('Content-Type', 'application/json');
    response.setBody('{"applicationNumber": "900-X"}');
    response.setStatusCode(201);
    return response;
}
}
```

The test class sets up the callout mock, sends the mortgage application request, and then mocks the callback with the mortgage application outcome:

```
@IsTest
public class AcmeMortgageApplicationTest {
    @IsTest
    static void testSubmitApplicationWithApprovalCallback() {
        AcmeMortgageHttpCalloutMock calloutMock = new AcmeMortgageHttpCalloutMock();
        Test.setMock(HttpCalloutMock.class, calloutMock);
        Test.startTest();

        ExternalService.AcmeMortgage acme = new ...();
        ...acme.SubmitApplication(...);

        // Call back with mock responses
        calloutMock.callBack();
        // Send all registered callback
        Test.stopTest();

        // Assert expected end result - e.g approved mortgage object created
        ...
    }
}
```

Monitor and Debug Asynchronous Callouts

Asynchronous callouts are monitored as jobs using the Background Operations app or with Apex log lines in the Developer Console. To debug your code at runtime, use Apex log lines.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

Use the Background Operations app for the most detailed status information about your asynchronous callouts.

Use Apex log lines in the Developer Console when it's more convenient for your workflow.

External Services Asynchronous Callout States

Here are the External Services asynchronous callout states that appear in both the Background Operations app and in the Debug log.

State	Description
Completed	The initial synchronous response and processing of the asynchronous response are successful.
Running	The job is waiting for the asynchronous response.
Error	There are two possibilities: The synchronous, initial response from the external web service has encountered an error (40x, 50x, and so on). Or Salesforce has received the asynchronous response and encountered a processing error.

Monitor and Debug Asynchronous Callbacks Using the Background Operations App

For the most detailed status and debugging information, use a filtered list view in the Background Operations App.

1. From the App Launcher, find and select **Background Operations**.
2. To see all External Services asynchronous jobs, create a custom list. Using the **List View Controls**, select **New**.
3. Give the new list view a name, for example, "External Services Asynchronous Callouts" and save your changes.
4. Under Filter, click **Add a Filter**.

5. Click **Field**, and select **Type**.
 6. Set **Operator** to **Equals**.
 7. Under Value, select **ExternalServiceCallback**, and then click **Done**.
 8. Save your changes.
- Optionally, pin the new list view to have it show right away when you open the Background Operations app.
9. To review operation details, click the background operation name.

The Error Message field provides information for detailed debugging. If there are no errors listed, and the Status of the job is Completed, then the job and the resulting Salesforce processing (for example, "create Contacts") have finished successfully.

Monitor and Debug Asynchronous Callbacks Using Apex Log Lines in the Developer Console

Debug logs allow fine-grained logging for each logging category.

-  **Note** If you have a top-level input parameter defined for callback URLs, and it's a valid callback expression, then in the Input Parameter column, the details of the input parameter has a `callback URL` description. For example, `https://MyDomainName.my.salesforce.com/services/data/v60.0/externalservices/callback/ 0EXAMPLE0000000q/applicationOutcomeApproved`.

Although External Services callbacks are inbound calls that don't involve a named credential, External Services log under the category Callout as part of Named Credentials callout events. Use these events to log end-to-end asynchronous callouts.

Event Name	Description	Category	Log Level
EXTERNAL_SERVICE_REQUEST	Logs external service callout request. Schema version, protocol, operation, request payload in the JSON up to a given size.	Callout	INFO and above
EXTERNAL_SERVICE_RESPONSE	Logs the initial, synchronous HTTP response. Schema version, protocol, normal output or exception, parameter name, operation. Logs the initial, synchronous HTTP response payload (output or exception) in JSON up to a given size.	Callout	INFO and above
EXTERNAL_SERVICE_CALLBACK	Logs the asynchronous callback	Callout	INFO and above

Event Name	Description	Category	Log Level
LLBACK	response. Includes Schema version, protocol, and the asynchronous callback payload in JSON up to a given size.		

Find Debug Logs for Asynchronous Callouts

Use Apex logging for monitoring, troubleshooting, and debugging External Services Asynchronous Callouts.

To open debug logs, select the **Logs** tab in the Developer Console.

Logs open in the Log Inspector.

See Also

[Working with Logs in the Developer Console](#)

View Apex Names in Apex Class Viewer

View all External Services auto-generated Apex classes in the Apex Class viewer in Setup.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

As an example, let's examine the BankServices' Apex classes.

1. From Setup, in the Quick Find box, enter *Apex Classes*, and then select **Apex Classes**.
2. Scroll down to **Dynamic Apex Classes**.
3. To filter the view, click **Create New View**.
4. Enter a **View Name**.
5. Under Filter By Additional Fields (optional), select
 - a. Field - Class Name
 - b. Operator - starts with
 - c. Value - “BankService”
6. Save your changes.

Now the list includes only classes that begin with “BankService”.

To see the external service associated with your class, click **Open** under the Action column.

To see details about the Apex class, click the Class Name. In this example, click **BankService**. In this detailed view, you can see all of the Apex names. The order of components is the same as they appeared in the original JSON-formatted schema. Operations, Requests, and Responses are highlighted in all caps

to make it easier to browse the Apex code. In this partial snippet, you can view the details for `BankService_accountDetails`.



Add External Service Actions to an Einstein Bot

Integrating your Einstein bot with a registered external service is now as easy as adding an action to a dialog. You can add an external service action to your bot from Einstein's Bot Builder.

Sometimes the data that your Einstein bot uses to personalize customer requests is stored outside of Salesforce. For example, a banking customer wants to update their account password and billing address in a single bot conversation. Or a retail customer starts a return process for a purchase and expects the bot to return a unique tracking number. Now you can automate these multi-platform requests directly from the Bot Builder, without writing a single line of code.

See Also

[Add an External Service Action](#)

Invoke External Services from Omnistudio Assets

You can invoke External Services' registered actions from Omnistudio Integration Procedures in OmniScripts and FlexCards.

To invoke External Services' registered actions with Omnistudio Integration Procedures, first create a custom Apex Class that implements the `System.Callable` interface. Create the class with methods and parameters, and then call the External Services API. For example:

```
public with sharing class RemoteActionClass implements System.Callable
{
    public Object call(String action, Map<String, Object> args)
    {
        Map<String, Object> inputMap = (Map<String, Object>) args.get('input');
        Map<String, Object> outputMap = (Map<String, Object>) args.get('output');
        Map<String, Object> options = (Map<String, Object>) args.get('options');

        if (action == 'methodName')
            Object result = Call External Services API;

        outputMap.put('result', result);

        return outputMap;
    }
}
```

{}

Then use an Integration Procedure to call an External Services invocable action with a Remote Action.

See Also

- [Workflow for Remote Action Apex Class Example](#)
- [Create a Remote Action for an Invocable Action](#)
- [Remote Action for Integration Procedures](#)
- [Remote Action Properties for Integration Procedures](#)
- [Trailhead: OmniStudio Integration Procedures](#)

Install or Create Packages

You can install or create packages that contain an external service. Read these tips before you begin.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

Consuming an External Services Package

Any flow within the package or newly added flows outside the package can reference external service actions.

Creating an External Services Package

To ensure that subscriber orgs that install a package can use the External Service actions from the External Services registration: Add named credential components to the external services registration package. Alternatively, a subscriber can create a named credential in the subscriber org using the same name as the one specified in the external services registration that references it.

See Also

- [External Services](#)
- [Components Available in Managed Packages](#)

Testing External Services

Test the example flow `MyAtmFlow` with a flow action calling the external service `MyAtmExternalService` by implementing the `HttpCalloutMock` interface. The interface provides a test response without performing the callout to the external service endpoint. The sample Apex unit test performs a mock test against the same HTTP callout mock implementation.

```
{  
    "swagger": "2.0",  
    "basePath": "/",  
    "info": {  
        "version": "1.0",  
        "title": "My ATM External Service Demo",  
        "description": "Demo ATM OpenAPI service for mock testing"  
    },  
    "paths": {  
        "/accounts/{accountId)": {  
            "get": {  
                "operationId": "getBalance",  
                "summary": "Get account balance",  
                "description": "Get the current account balance for the given account  
ID and pin",  
                "consumes": [  
                    "text/plain"  
                ],  
                "produces": [  
                    "application/json"  
                ],  
                "parameters": [{  
                    "name": "accountId",  
                    "in": "path",  
                    "required": true,  
                    "type": "string",  
                    "description": "Account ID"  
                }, {  
                    "name": "pin",  
                    "in": "header",  
                    "required": true,  
                    "type": "integer",  
                    "description": "Account holder's PIN"  
                }],  
                "responses": {  
                    "200": {  
                        "description": "The response when system finds an account with giv  
en name",  
                        "schema": {  
                            "required": [  
                                "accountId",  
                                "name",  
                                "type",  
                                "availableBal"  
                            ]  
                        }  
                    }  
                }  
            }  
        }  
    }  
}
```

```
        ],
        "type": "object",
        "properties": {
            "accountId": {
                "type": "string",
                "description": "Unique account ID"
            },
            "name": {
                "type": "string",
                "description": "Account name"
            },
            "type": {
                "type": "string",
                "description": "Account type",
                "example": "Checking"
            },
            "availableBal": {
                "type": "integer",
                "description": "Available balance"
            }
        }
    },
    "401": {
        "description": "Invalid authentication or invalid account"
    }
}
}
```

`MyAtmExternalService` is registered with a Named Credential `MyAtmNamedCredential`, has operation `getBalance` with String `accountId` as path parameter and Integer `pin` as header parameter. Its output is a JSON data structure with the account ID, account holder, balance, and account type.

The HTTP callout mock implementation reacts to an expected test request by responding with the appropriate HTTP response output:

```
global class MyAtmHttpCalloutMock implements HttpCalloutMock {  
    global HttpResponse respond(HTTPRequest request) {  
        // Assert expected request test data  
        System.assertEquals('GET', request.getMethod());
```

```
        System.assertEquals('callout:MyAtmNamedCredential/A123-456', request.getEndpoint());
        System.assertEquals('1234', request.getHeader('pin'));

        // Send response test data
        HttpResponse response = new HttpResponse();
        response.setHeader('Content-Type', 'application/json');
        response.setBody('{"availableBal": 0, "name": "Account Holder", "type": "Checking", "accountId": "A123-456"}');
        response.setStatusCode(200);
        return response;
    }
}
```

The flow test class sets up the HTTP callout mock for the external service action it's calling. It creates a flow interview with input parameter values to test, runs the flow interview, and then asserts the actual flow output parameters with expected test values:

```
@IsTest
public class MyAtmFlowTest {
    @IsTest
    static public void testMyAtmFlow() {
        // Set HTTP callout mock to match flow's external service action invocation
        Test.setMock(HttpCalloutMock.class, new MyAtmHttpCalloutMock());

        // Set flow input variables and create the flow interview
        Map<String, Object> inputVariables = new Map<String, Object>();
        inputVariables.put('accountId', 'A123-456');
        inputVariables.put('pin', 1234);
        Flow.Interview myAtmFlow = Flow.Interview.createInterview('MyAtmFlow',
            inputVariables);

        // Start flow interview with set input variables
        myAtmFlow.start();

        // Assert flow output variables
        ExternalService.MyAtmExternalService_getBalance_OUT_200 expected =
            new ExternalService.MyAtmExternalService_getBalance_OUT_200();
        expected.availableBal = 0;
        expected.name = 'Account Holder';
        expected.type = 'Checking';
        expected.accountId = 'A123-456';
        System.assertEquals(expected.toString(),
```

```
        (String)myAtmFlow.getVariableValue('myFlowOutput'));  
    }  
}
```

The Apex test class sets up the HTTP callout mock for the external service method it's calling. It calls directly the external service in Apex with input parameter values to test and then asserts the actual output parameters with expected test values. This Apex test class illustrates how you can test an external service directly, for example, to assert an expected behavior for a registered service. Application code using external services in Apex can perform unit testing following the same pattern:

```
@IsTest  
public class MyAtmApexTest {  
    @IsTest  
    static public void testMyAtmCallout() {  
        // Set HTTP callout mock  
        Test.setMock(HttpCalloutMock.class, new MyAtmHttpCallout());  
  
        // Call with the expected mock accountId and pin  
        ExternalService.MyAtmExternalService myAtm =  
            new ExternalService.MyAtmExternalService();  
        ExternalService.MyAtmExternalService.getBalance_Request request =  
            new ExternalService.MyAtmExternalService.getBalance_Request();  
        request.accountId = 'A123-456';  
        request.pin = 1234;  
        ExternalService.MyAtmExternalService_getBalance_OUT_200 actual =  
            myAtm.getBalance(request).Code200;  
  
        // Assert the response variables  
        ExternalService.MyAtmExternalService_getBalance_OUT_200 expected =  
            new ExternalService.MyAtmExternalService_getBalance_OUT_200();  
        expected.availableBal = 0;  
        expected.name = 'Account Holder';  
        expected.z0type = 'Checking';  
        expected.accountId = 'A123-456';  
        System.assertEquals(expected.toString(), actual.toString());  
    }  
}
```

Schema Examples

Explore various scenarios with OpenAPI 2.0 and OpenAPI 3.0 compliant JSON or YAML schemas supported by External Services. The examples cover schema elements like HTTP header as input

parameters, and include example usage in Apex. Understanding the examples helps with proper syntax and code placement.

External Services OpenAPI 2.0 Schema

This section features External Services OpenAPI 2.0 schema examples.

External Services OpenAPI 3.0 Schema

This section features External Services OpenAPI 3.0 schema examples.

Using the Schema Examples

See how the respective Open API 2.0 and 3.0 examples are implemented with Apex.

External Services OpenAPI 2.0 Schema

This section features External Services OpenAPI 2.0 schema examples.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

Example 1: Basic OpenAPI Spec with Request and Response (OAS 2.0)

Here's an example of an API spec that contains a supported JSON schema for OpenAPI 2.0. The parameters (in bold) contain the definition for the `accountId` input. The responses (also in bold) contain the definition for the output, which is `creditRating`. Parameters and responses translate to inputs and outputs, respectively, for your flow actions.

```
{
    "swagger": "2.0",
    "info": {
        "description": "A service for checking credit for an account.",
        "version": "1.0.0",
        "title": "Credit Decision",
        "termsOfService": "http://swagger.io/terms/",
        "contact": {
            "email": "apiteam@swagger.io"
        },
        "license": {
            "name": "Apache 2.0",
            "url": "http://www.apache.org/licenses/LICENSE-2.0.html"
        }
    },
    "host": "<YourHostName>",
}
```

```
"paths": {
    "/account/lastCreditRating": {
        "get": {
            "summary": "Evaluates credit rating and decides what payment terms to offer.",
            "description": "",
            "consumes": [
                "application/json",
                "application/xml"
            ],
            "produces": [
                "application/xml",
                "application/json"
            ],
            "parameters": [
                {
                    "in": "body",
                    "name": "body",
                    "description": "Specifies input parameters to calculate payment term",
                    "required": true,
                    "schema": {
                        "$ref": "#/definitions/accountId"
                    }
                }
            ],
            "responses": {
                "200": {
                    "description": "success",
                    "schema": {
                        "$ref": "#/definitions/creditRating"
                    }
                },
                "405": {
                    "description": "Invalid input"
                }
            }
        }
    }
},
"definitions": {
    "accountId": {
        "type": "object",
        "properties": {
            "accountIdString": {
                "type": "string"
            }
        }
    }
}
```

```
        }
    },
    "xml": {
        "name": "accountId"
    }
},
"creditRating": {
    "type": "object",
    "properties": {
        "creditRatingString": {
            "type": "string"
        }
    },
    "xml": {
        "name": "creditRating"
    }
}
}
```

Example 2: Named Object Schema Reference (OAS 2.0)

Basic Setup

- For the named credential that your org uses to access the banking system, assign the *Bank* label and a placeholder URL, such as `https://api.example.com`. Use `example.com` because you'll *paste in* the schema at registration time, instead of using a URL to point to an API spec.
 - Register the employee banking system's external service. Use the *Bank* name and the *Bank* named credential, and then copy and paste in the following schema.
 - The banking system's external service shows two ways to define parameters with object types: a named object type under a "*definitions*" block or an anonymously declared object type inline.

Here's a JSON-formatted schema with the named object type defined under the "definitions" block. The phone object type's name is `ExternalService_Bank_Phone` in Apex or Flow Builder.

```
{
  "swagger": "2.0",
  "info": {
    "title": "bankService",
    "description": "API description in Markdown."
  }
}
```

```
"version": "1.0.0"
},
"host": "api.example.com",
"basePath": "/v1",
"schemes": [
  "https"
],
"definitions": {
  "User": {
    "properties": {
      "id": {
        "type": "integer"
      },
      "name": {
        "type": "string"
      },
      "phones": {
        "type": "array",
        "items": {
          "type": "object",
          "$ref": "#/definitions/Phone"
        }
      }
    }
  },
  "Phone": {
    "properties": {
      "typeofphone": {
        "type": "string"
      },
      "phone": {
        "type": "string"
      }
    }
  }
},
"paths": {
  "/users/{userId)": {
    "get": {
      "summary": "Returns a user by ID.",
      "parameters": [
        {
          "in": "path",
          "name": "userId",
          "required": true,
        }
      ]
    }
  }
}
```

```
        "type": "integer"
    ],
    "responses": {
        "200": {
            "description": "OK",
            "schema": {
                "$ref": "#/definitions/User"
            }
        }
    }
},
"/users": {
    "post": {
        "summary": "Creates a new user.",
        "parameters": [
            {
                "in": "body",
                "name": "user",
                "schema": {
                    "$ref": "#/definitions/User"
                }
            }
        ],
        "responses": {
            "200": {
                "description": "OK"
            }
        }
    }
}
}
```

Here's the same schema, but formatted with YAML.

```
swagger: '2.0'
info:
  title: bankService
  description: API description in Markdown.
  version: 1.0.0
host: api.example.com
basePath: /v1
schemes:
  - https
definitions:
```

```
User:  
  properties:  
    id:  
      type: integer  
    name:  
      type: string  
  phones:  
    type: array  
    items:  
      type: object  
      $ref: '#/definitions/Phone'  
  
Phone:  
  properties:  
    typeofphone:  
      type: string  
    phone:  
      type: string  
  
paths:  
/users/{userId}:  
  get:  
    summary: Returns a user by ID.  
    parameters:  
      - in: path  
        name: userId  
        required: true  
        type: integer  
    responses:  
      '200':  
        description: OK  
        schema:  
          $ref: '#/definitions/User'  
  
/users:  
  post:  
    summary: Creates a new user.  
    parameters:  
      - in: body  
        name: user  
        schema:  
          $ref: '#/definitions/User'  
    responses:  
      '200':  
        description: OK
```

Example 3: Nested Anonymous Object Schema (OAS 2.0)

Here's an API spec with a JSON schema with anonymous object types defined inline. The derived phone object type's name is `ExternalService__Bank_getUsers_OUT_200_phones`.

-  **Note** For anonymous object types, Salesforce automatically derives object type names based on the external service, parent operation, operation parameter, parent object, and object property names. The derived name must be fewer than 255 characters. For more information, see "External Service Apex Class Names and Developer Names" in [External Services Considerations](#).

```
{  
    "swagger": "2.0",  
    "info": {  
        "title": "bankService",  
        "description": "API description in Markdown.",  
        "version": "1.0.0"  
    },  
    "host": "api.example.com",  
    "basePath": "/v1",  
    "schemes": [  
        "https"  
    ],  
    "paths": {  
        "/users/{userId)": {  
            "get": {  
                "summary": "Returns a user by ID.",  
                "parameters": [{  
                    "in": "path",  
                    "name": "userId",  
                    "required": true,  
                    "type": "integer"  
                }],  
                "responses": {  
                    "200": {  
                        "description": "OK",  
                        "schema": {  
                            "type": "object",  
                            "properties": {  
                                "id": {  
                                    "type": "integer"  
                                },  
                                "name": {  
                                    "type": "string"  
                                },  
                                "phones": {  
                                    "type": "array",  
                                    "items": {  
                                        "type": "object",  
                                        "properties": {  
                                            "number": {  
                                                "type": "string"  
                                            },  
                                            "type": {  
                                                "type": "string"  
                                            }  
                                        }  
                                    }  
                                }  
                            }  
                        }  
                    }  
                }  
            }  
        }  
    }  
}
```

```
        "phones": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "typeofphone": {
                        "type": "string"
                    },
                    "phone": {
                        "type": "string"
                    }
                }
            }
        }
    }
},
"/users": {
    "post": {
        "summary": "Creates a new user.",
        "parameters": [
            {
                "in": "body",
                "name": "user",
                "schema": {
                    "type": "object",
                    "properties": {
                        "id": {
                            "type": "integer"
                        },
                        "name": {
                            "type": "string"
                        },
                        "phones": {
                            "type": "array",
                            "items": {
                                "type": "object",
                                "properties": {
                                    "typeofphone": {
                                        "type": "string"
                                    },
                                    "phone": {

```

Example 4: Apex Object Class Naming (OAS 2.0)

Here's an API spec registered with the external service name BankingAutomaticTellerMachine. However, the derived object names must be no longer than 255 characters:

- ExternalService__BankingAutomaticTellerMachine_VeryImportantCustomer_phone
 - ExternalService__BankingAutomaticTellerMachine_getBalanceAccountTypeChecking_OUT_200

To use Apex or Flow Builder, you must shorten the external service schema name and the schema (shown in Example 5).

-  **Tip** If the derived object name is longer than 255 characters, try one of the following methods to resolve the issue.

- Shorten the external service name.
 - If the object is declared inline under an operation parameter, shorten the operation name by adding an `operationId` to the schema.
 - If the object is declared inline under a parent object property, shorten the parent object name in the schema.
 - Declare the nested object as a top-level object under schema “*definitions*”.

```
{  
  "swagger": "2.0",  
  ...  
  "paths": {
```

```
"/balance/account/{accountId}/type/checking": {
    "get": {
        "parameters": [
            {
                "in": "path",
                "name": "accountId",
                "required": true,
                "type": "integer"
            }
        ],
        "responses": {
            "200": {
                "schema": {
                    "type": "object",
                    "properties": {
                        "balance": {
                            "type": "integer"
                        },
                        "owner": {
                            "$ref": "#/definitions/VeryImportantCustomer"
                        }
                    }
                }
            }
        }
    }
},
"definitions": {
    "VeryImportantCustomer": {
        "type": "object",
        "properties": {
            "name": {
                "type": "string"
            },
            "phone": {
                "type": "object",
                "properties": {
                    "number": {
                        "type": "string"
                    }
                }
            }
        }
    }
}
```

{}

Example 5: Shortening Apex Object Class Names (OAS 2.0)

Here's an example of an API spec with a schema that results in shortened names. It's registered with the shortened name BankAtm.

Shorten the external service schema name to BankAtm, the schema object name to VIP and add operationId as getBalance:

- ExternalService__BankAtm_VIP_phone
- ExternalService__BankAtm_getBalance_200

```
{  
  "swagger": "2.0",  
  ...  
  "paths": {  
    "/balance/account/{accountId}/type/checking": {  
      "get": {  
        "operationId": "getBalance",  
        "parameters": [{  
          "in": "path",  
          "name": "accountId",  
          "required": true,  
          "type": "integer"  
        }],  
        "responses": {  
          "200": {  
            "schema": {  
              "type": "object",  
              "properties": {  
                "balance": {  
                  "type": "integer"  
                },  
                "owner": {  
                  "$ref": "#/definitions/VIP"  
                }  
              }  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

```
,  
  "definitions": {  
    "VIP": {  
      "type": "object",  
      "properties": {  
        "name": {  
          "type": "string"  
        },  
        "phone": {  
          "type": "object",  
          "properties": {  
            "number": {  
              "type": "string"  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

Example 6: Inline Arrays (OAS 2.0)

Here's an example with an inline array definition.

```
{  
  "swagger": "2.0",  
  "host": "Employees.org",  
  "basePath": "/",  
  ...  
  "paths": {  
    "/employees/{employeeId)": {  
      "get": {  
        "operationId": "getEmployee",  
        "parameters": [{  
          "in": "path",  
          "name": "employeeId",  
          "required": true,  
          "type": "string"  
        }],  
        "responses": {  
          "200": {  
            "schema": {  
              "type": "object",  
              "properties": {  
                "name": {  
                  "type": "string"  
                },  
                "phone": {  
                  "type": "object",  
                  "properties": {  
                    "number": {  
                      "type": "string"  
                    }  
                  }  
                }  
              }  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

```

        "type": "object",
        "properties": {
            "employee": {"$ref": "#/definitions/Employee"},
            "manager": {"$ref": "#/definitions/Employee"},
            "team": {
                "type": "array",
                "items": {"$ref": "#/definitions/Employee"}
            }
        }
    },
    "definitions": {
        "Employee" : {
            "type": "object",
            "properties": {
                "employeeId": {"type": "string"},
                "firstName": {"type": "string"},
                "middleName": {"type": "string"},
                "lastName": {"type": "string"},
                "dateOfHire": {"type": "date"}
            }
        }
    }
}

```

Example 7: HTTP Header Parameters (OAS 2.0)

Here's how a header parameter is declared in an OpenAPI schema. In Apex or Flow, the name `"apiKey"` shows up as a string parameter. Now when you set any string value in Apex or a flow to `apiKey`, it functions as an HTTP parameter when making a callout request.

```
{
    "swagger": "2.0",
    "info": {
        "description": "A service for checking credit for an account.",
        "version": "1.0.0",
        "title": "Credit Decision",
        "termsOfService": "http://swagger.io/terms/"
}
```

```
"contact": {
    "email": "apiteam@swagger.io"
},
"license": {
    "name": "Apache 2.0",
    "url": "http://www.apache.org/licenses/LICENSE-2.0.html"
},
},
"host": "<YourHostName>",
"paths": {
    "/account/lastCreditRating": {
        "get": {
            "summary": "Evaluates credit rating and decides what payment terms to offer.",
            "description": "",
            "consumes": [
                "application/json",
                "application/xml"
            ],
            "produces": [
                "application/xml",
                "application/json"
            ],
            "parameters": [
                {
                    "name": "body",
                    "description": "Specifies input parameters to calculate payment term",
                    "in": "body",
                    "required": true,
                    "schema": {
                        "$ref": "#/definitions/accountId"
                    }
                },
                {
                    "name": "apiKey",
                    "description": "Your API Key for calling the credit rating service",
                    "in": "header",
                    "type": "string"
                }
            ],
            "responses": {
                "200": {
                    "description": "success",
                    "schema": {
                        "$ref": "#/definitions/creditRating"
                    }
                }
            }
        }
    }
}
```

```

        },
        "405": {
            "description": "Invalid input"
        }
    }
}
},
"definitions": {
    "accountId": {
        "type": "object",
        "properties": {
            "accountIdString": {
                "type": "string"
            }
        },
        "xml": {
            "name": "accountId"
        }
    },
    "creditRating": {
        "type": "object",
        "properties": {
            "creditRatingString": {
                "type": "string"
            }
        },
        "xml": {
            "name": "creditRating"
        }
    }
}
}
}

```

Example 8: URL Encoded Form Media Type (OAS 2.0)

Request and response form data is declared alongside the matching `consumes` and `produces` directives.

```
{
    "swagger": "2.0",
    "info": {
        "description": "Apply here for your next mortgage",

```

```
"version": "1.0.0",
"title": "My Mortgage Buddy",
"contact": {
    "email": "apiteam@swagger.io"
},
"host": "MyMortgageBuddy.org",
"basePath": "/mortgages",
"paths": {
    "/apply": {
        "post": {
            "operationId": "applyMortgage",
            "consumes": [
                "application/x-www-form-urlencoded; charset=utf-8"
            ],
            "produces": [
                "application/x-www-form-urlencoded; charset=utf-8"
            ],
            "parameters": [
                {
                    "description": "Desired mortgage terms",
                    "name": "terms",
                    "in": "formData",
                    "type": "array",
                    "items": {
                        "type": "integer"
                    },
                    "collectionFormat": "multi",
                    "required": true
                },
                {
                    "description": "Full Name",
                    "name": "fullName",
                    "in": "formData",
                    "type": "string",
                    "required": true
                },
                {
                    "description": "Loan amount",
                    "name": "loanAmount",
                    "in": "formData",
                    "type": "number",
                    "required": true
                }
            ],
            "responses": {
                "200": {
                    "description": "200"
                }
            }
        }
    }
}
```

```
    "schema": {
        "type": "object",
        "properties": {
            "formApplicationId": {
                "type": "string"
            },
            "loanOfficerFullName": {
                "type": "string"
            }
        }
    }
}
```

Example 9: allOf and additionalProperties Schema Directives (OAS 2.0)

This JSON schema definition highlights the constructs `allOf` for schema object composition and `additionalProperties` for dictionary values. The sample schema is registered as an external service `MyBank` with named credential `MyBank`. The registration is invoked by a sample flow that accesses the dictionary properties with an Apex invocable action. A flow Apex unit test ties it all together.

The schema defines a banking service that gets customer details for a customer ID.

- The `Customer` schema object has dictionary properties of type schema object `CreditRating`. In Apex, the class `ExternalService.MyBank_Customer` has property `properties` of type `Map<String, ExternalService_CreditRating>` with the customer's credit ratings.
 - Phones and Emails compose their own properties together with the common `allOf` properties from the schema object `Contact`.

```
{  
    "swagger": "2.0",  
  
    "info": {  
        "title": "myBank",  
        "description": "Sample Banking Service with allOf and additionalProperties  
schema constructs",  
        "version": "1.0.0"  
    },  
}
```

```
"host": "api.mybank.com",
"basePath": "/v1",
"schemes": [
  "https"
],


"definitions": {
  "Customer": {
    "type": "object",
    "properties": {
      "id": {
        "type": "integer"
      },
      "name": {
        "type": "string"
      },
      "phones": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/Phone"
        }
      },
      "emails": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/Email"
        }
      }
    },
    "additionalProperties": {
      "$ref": "#/definitions/CreditRating"
    },
    "required": [
      "id",
      "name"
    ]
  },
  "Contact": {
    "type": "object",
    "properties": {
      "primary": {
        "type": "boolean"
      },
      "label": {
        "type": "string"
      }
    }
  }
}
```

```
    "timeOfDay": {
        "type": "string"
    },
    "required": [
        "primary"
    ],
},
"Phone": {
    "allOf": [
{
    "$ref": "#/definitions/Contact"
},
{
    "type": "object",
    "properties": {
        "typeOfPhone": {
            "type": "string"
        },
        "phoneNumber": {
            "type": "string"
        }
    }
},
    ]
},
"Email": {
    "allOf": [
{
    "$ref": "#/definitions/Contact"
},
{
    "type": "object",
    "properties": {
        "email": {
            "type": "string"
        }
    }
},
    ]
},
"CreditRating": {
```

```
"type": "object",
"properties": {
    "rating": {
        "type": "string"
    },
    "score": {
        "type": "number",
        "format": "double"
    }
},
},
},
}

"paths": {
    "/customers/{customerId)": {
        "get": {
            "summary": "Get the customer by ID.",
            "parameters": [
                {
                    "in": "path",
                    "name": "customerId",
                    "required": true,
                    "type": "integer"
                }
            ],
            "responses": {
                "200": {
                    "description": "OK",
                    "schema": {
                        "$ref": "#/definitions/Customer"
                    }
                }
            }
        }
    }
}
```

To use `allOf` composition and `additionalProperties`, see [For Examples 9: allOf Composition and additionalProperties Use in Flow with Apex Unit Tests](#).

Example 10: allOf and Discriminator Directives (OAS 2.0)

To specify a general extensible list of contacts for a customer, the `discriminator` directive can be combined with `allOf` to declare whether a contact is an email or a phone number:

```
{  
  "swagger": "2.0",  
  
  "info": {  
    "title": "myBank",  
    "description": "Sample Banking Service with allok and discriminator",  
    "version": "1.0.0"  
  },  
  "host": "api.mybank.com",  
  "basePath": "/v1",  
  "schemes": [  
    "https"  
  ],  
  "definitions": {  
    "Customer": {  
      "type": "object",  
      "properties": {  
        "id": {  
          "type": "integer"  
        },  
        "name": {  
          "type": "string"  
        },  
        "contacts": {  
          "type": "array",  
          "items": {  
            "$ref": "#/definitions/Contact"  
          }  
        }  
      },  
      "required": [  
        "id",  
        "name"  
      ]  
    },  
    "Contact": {  
      "type": "object",  
      "discriminator": "contactType",  
      "properties": {  
        "contactType": {  
          "type": "string"  
        },  
        "name": {  
          "type": "string"  
        },  
        "id": {  
          "type": "integer"  
        }  
      }  
    }  
  }  
}
```

```
"primary": {
    "type": "boolean"
},
"timeOfDay": {
    "type": "string"
}
},
"required": [
    "contactType",
    "primary"
]
},
"Phone": {
    "allOf": [
        {
            "$ref": "#/definitions/Contact"
        },
        {
            "type": "object",
            "properties": {
                "typeOfPhone": {
                    "type": "string"
                },
                "phoneNumber": {
                    "type": "string"
                }
            }
        }
    ]
},
"Email": {
    "allOf": [
        {
            "$ref": "#/definitions/Contact"
        },
        {
            "type": "object",
            "properties": {
                "email": {
                    "type": "string"
                }
            }
        }
    ]
}
```

```

    }
  } ,

  "paths": {
    "/customers/{customerId)": {
      "get": {
        "summary": "Get the customer by ID.",
        "parameters": [
          {
            "in": "path",
            "name": "customerId",
            "required": true,
            "type": "integer"
          }],
        "responses": {
          "200": {
            "description": "OK",
            "schema": {
              "$ref": "#/definitions/Customer"
            }
          }
        }
      }
    }
  }
}

```

To use composition and polymorphic OpenAPI schema constructs, see [For Examples 10: Polymorphism with allOf and Discriminator](#).

Array Definition (OAS 2.0)

External Services supports inline array definitions and referenceable named arrays. List types in External Services are identified by their object element type.

Supported Inline Array Definition with Reference to Array Items Definition

```
{
  "swagger": "2.0",
  ...
  "name": "myObjects",
  "in": "body",
  "schema": {
    "type": "array",

```

```

"items": {
    "$ref": "#definitions/MyObject"
}
}

...
"definitions": {
    "MyObject": {
        "type": "object",
        "properties": {...}
    }
}
}

```

In Apex and Flow Builder, declare a variable for the `myObjects` parameter by marking the variable as a collection and picking its Apex element type `ExternalService__RegistrationName__MyObject`.

Inline Array Definition with Inline Array Items Definition

```

{
    "swagger": "2.0",
    ...
    "name": "myObjects",
    "in": "body",
    "schema": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {...}
        }
    }
    ...
    "definitions": {
        ...
    }
}

```

In Apex and Flow Builder, declare a collection variable for the `myObjects` parameter and Apex element type `ExternalService__RegistrationName__OperationName_IN_myObjects`.

Referenceable Array Definition with Reference to Array Item Definitions

```
{
    "swagger": "2.0",
}
```

```

...
  "name": "myObjects",
  "in": "body",
  "schema": {
    "$ref": "#definitions/MyObjectList"
  }
}

...
"definitions": {
  "MyObjectList": {
    "type": "array",
    "items": {
      "$ref": "#definitions/MyObject"
    }
  }
  "MyObject": {
    "type": "object",
    "properties": {...}
  }
}
}
}

```

In Apex and Flow Builder, declare a collection variable for the `myObjects` parameter and Apex element type `ExternalService__RegistrationName__MyObject`.

Referenceable Array Definition with Inline Array Items Definition

```

{
  "swagger": "2.0",
  ...
  "name": "myObjects",
  "in": "body",
  "schema": {
    "$ref": "#definitions/MyObjectList"
  }
}

...
"definitions": {
  "MyObjectList": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {...}
    }
  }
}

```

```
    }  
}
```

In Apex and Flow Builder, declare a collection variable for the `myObjects` parameter and Apex element type `ExternalService__RegistrationName__MyObjectList`.

See Also

[External Services OpenAPI 3.0 Schema](#)

[Using the Schema Examples](#)

[External Services Considerations](#)

External Services OpenAPI 3.0 Schema

This section features External Services OpenAPI 3.0 schema examples.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

Example 1: Basic OpenAPI Spec with Request and Response (OAS 3.0)

Here's an example of an API spec that contains a supported JSON schema for OpenAPI 3.0. The parameters contain the definition for the `accountId` input. The responses contain the definition for the output, which is `creditRating`. Parameters, requests, and responses translate to inputs and outputs, respectively, for your flow actions.

```
{  
  "openapi": "3.0.0",  
  "info": {  
    "description": "A service for checking credit for an account.",  
    "version": "1.0.0",  
    "title": "Credit Decision",  
    "termsOfService": "http://swagger.io/terms/",  
    "contact": {  
      "email": "apiteam@swagger.io"  
    },  
    "license": {  
      "name": "Apache 2.0",  
      "url": "http://www.apache.org/licenses/LICENSE-2.0.html"  
    }  
},
```

```
"servers": [
  {
    "url": "https://<YourHostName>"
  }
],
"paths": {
  "/account/lastCreditRating": {
    "get": {
      "summary": "Evaluates credit rating and decides what payment terms to offer.",
      "description": "",
      "requestBody": {
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/accountId"
            }
          },
          "application/xml": {
            "schema": {
              "$ref": "#/components/schemas/accountId"
            }
          }
        },
        "description": "Specifies input parameters to calculate payment term",
        "required": true
      },
      "responses": {
        "200": {
          "description": "success",
          "content": {
            "application/xml": {
              "schema": {
                "$ref": "#/components/schemas/creditRating"
              }
            },
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/creditRating"
              }
            }
          }
        }
      }
    }
  }
}
```

```
        },
        "405": {
            "description": "Invalid input"
        }
    }
}
},
"components": {
    "schemas": {
        "accountId": {
            "type": "object",
            "properties": {
                "accountIdString": {
                    "type": "string"
                }
            },
            "xml": {
                "name": "accountId"
            }
        },
        "creditRating": {
            "type": "object",
            "properties": {
                "creditRatingString": {
                    "type": "string"
                }
            },
            "xml": {
                "name": "creditRating"
            }
        }
    }
}
```

Example 2: Named Object Schema Reference (OAS 3.0)

Basic Setup

- For the named credential that your org uses to access the banking system, assign the *Bank* label and a placeholder URL, such as <https://api.example.com>. Use *example.com* because you *paste in* the schema at registration time, instead of using a URL to point to an API spec.
- Register the employee banking system's external service. Use the *Bank* name and the *Bank* named

credential, and then copy and paste in the schema.

- The banking system's external service shows two ways to define parameters with object types: a named object type under a "*components/schema*" block or an anonymously declared object type inline.



Note The defined or derived parameter object type name, or a property with an object type, must be fewer than 255 characters to be used in Apex or Flow Builder.

Here's a schema with the named object type defined under the "*components/schema*" block. The phone object type's name is *ExternalService__Bank__Phone* in Apex or Flow Builder.

```
{  
  "openapi": "3.0.0",  
  "info": {  
    "title": "bankService",  
    "description": "API description in Markdown.",  
    "version": "1.0.0"  
  },  
  "servers": [  
    {  
      "url": "https://api.example.com/v1"  
    }  
  ],  
  "components": {  
    "schemas": {  
      "User": {  
        "properties": {  
          "id": {  
            "type": "integer"  
          },  
          "name": {  
            "type": "string"  
          },  
          "phones": {  
            "type": "array",  
            "items": {  
              "$ref": "#/components/schemas/Phone"  
            }  
          }  
        }  
      }  
    },  
    "Phone": {  
      "properties": {  
        "typeofphone": {  
          "type": "string"  
        }  
      }  
    }  
  }  
}
```

```
        "type": "string"
    },
    "phone": {
        "type": "string"
    }
}
}
}
}
}

"paths": {
"/users/{userId)": {
    "get": {
        "summary": "Returns a user by ID.",
        "parameters": [
            {
                "in": "path",
                "name": "userId",
                "required": true,
                "schema": {
                    "type": "integer"
                }
            }
        ],
        "responses": {
            "200": {
                "description": "OK",
                "content": {
                    "application/json": {
                        "schema": {
                            "$ref": "#/components/schemas/User"
                        }
                    }
                }
            }
        }
    }
},
"/users": {
    "post": {
        "summary": "Creates a new user.",
        "requestBody": {
            "content": {
                "application/json": {
                    "schema": {

```

```
        "$ref": "#/components/schemas/User"
    }
}
}
},
"responses": {
    "200": {
        "description": "OK"
    }
}
}
}
}
}
```

Example 3: Nested Anonymous Object Schema (OAS 3.0)

Here's an API spec with a JSON schema with anonymous object types defined inline. The derived phone object type's name is `ExternalService_Bank_getUsers_OUT_200_phones`.

-  **Note** For anonymous object types, Salesforce automatically derives object type names. The derived names are based on the external service, parent operation, operation parameter, parent object, and object property names. Each derived name must be fewer than 255 characters. For more information, see [Schema Definition Support](#).

```
{
  "openapi": "3.0.0",
  "info": {
    "title": "bankService",
    "description": "API description in Markdown.",
    "version": "1.0.0"
  },
  "servers": [
    {
      "url": "https://api.example.com/v1"
    }
  ],
  "paths": {
    "/users/{userId)": {
      "get": {
        "summary": "Returns a user by ID.",
        "parameters": [
          {
            "in": "path",
            "name": "userId",
            "required": true,
            "type": "string"
          }
        ]
      }
    }
  }
}
```

```
        "in": "path",
        "name": "userId",
        "required": true,
        "schema": {
            "type": "integer"
        }
    ],
    "responses": {
        "200": {
            "description": "OK",
            "content": {
                "application/json": {
                    "schema": {
                        "type": "object",
                        "properties": {
                            "id": {
                                "type": "integer"
                            },
                            "name": {
                                "type": "string"
                            },
                            "phones": {
                                "type": "array",
                                "items": {
                                    "type": "object",
                                    "properties": {
                                        "typeofphone": {
                                            "type": "string"
                                        },
                                        "phone": {
                                            "type": "string"
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
},
```

```
"/users": {
  "post": {
    "summary": "Creates a new user.",
    "requestBody": {
      "content": {
        "application/json": {
          "schema": {
            "type": "object",
            "properties": {
              "id": {
                "type": "integer"
              },
              "name": {
                "type": "string"
              },
              "phones": {
                "type": "array",
                "items": {
                  "type": "object",
                  "properties": {
                    "typeofphone": {
                      "type": "string"
                    },
                    "phone": {
                      "type": "string"
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  },
  "responses": {
    "200": {
      "description": "OK"
    }
  }
}
```

Example 4: Apex Object Class Naming (OAS 3.0)

Here's an API spec registered with the external service name BankingAutomaticTellerMachine. However, the derived object names must be no longer than 255 characters.

- `ExternalService__BankingAutomaticTellerMachine_VeryImportantCustomer_phone`
- `ExternalService__BankingAutomaticTellerMachine_getBalanceAccountTypeChecking_OUT_200`

To use Apex or Flow Builder, you must shorten the external service schema name and the schema (shown in Example 5).

 **Tip** If the derived object name is longer than 255 characters, try one of these methods to resolve the issue.

- Shorten the external service name.
- If the object is declared inline under an operation parameter, shorten the operation name by adding an `operationId` to the schema.
- If the object is declared inline under a parent object property, shorten the parent object name in the schema.
- Declare the nested object as a top-level object under schema "`components/schemas`".

```
{  
  "openapi": "3.0.0",  
  ...  
  "paths": {  
    "/balance/account/{accountId}/type/checking": {  
      "get": {  
        "parameters": [  
          {  
            "in": "path",  
            "name": "accountId",  
            "required": true,  
            "schema": {  
              "type": "integer"  
            }  
          }  
        ],  
        "responses": {  
          "200": {  
            "description": "",  
            "content": {  
              "*/*": {  
                "schema": {  
                  "type": "object",  
                  "properties": {  
                    "balance": {  
                      "type": "number",  
                      "format": "float"  
                    },  
                    "type": {  
                      "type": "string",  
                      "enum": ["checkings", "savings"]  
                    }  
                  }  
                }  
              }  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

```
        "properties": {
            "balance": {
                "type": "integer"
            },
            "owner": {
                "$ref": "#/components/schemas/VeryImportantCustomer"
            }
        }
    }
}
},
"components": {
    "schemas": {
        "VeryImportantCustomer": {
            "type": "object",
            "properties": {
                "name": {
                    "type": "string"
                },
                "phone": {
                    "type": "object",
                    "properties": {
                        "number": {
                            "type": "string"
                        }
                    }
                }
            }
        }
    }
}
```

Example 5: Shortening Apex Object Class Names (OAS 3.0)

Here's an example of an API spec with a schema that results in shortened names. It's registered with the shortened name BankAtm.

Shorten the external service schema name to `BankAtm`, the schema object name to `VIP` and add

```
operationId as getBalance :  
  
• ExternalService__BankAtm_VIP_phone  
• ExternalService__BankAtm_getBalance_200
```

```
{  
  "openapi": "3.0.0",  
  ...  
  "paths": {  
    "/balance/account/{accountId}/type/checking": {  
      "get": {  
        "operationId": "getBalance",  
        "parameters": [  
          {  
            "in": "path",  
            "name": "accountId",  
            "required": true,  
            "schema": {  
              "type": "integer"  
            }  
          }  
        ],  
        "responses": {  
          "200": {  
            "description": "",  
            "content": {  
              "application/json": {  
                "schema": {  
                  "type": "object",  
                  "properties": {  
                    "balance": {  
                      "type": "integer"  
                    },  
                    "owner": {  
                      "$ref": "#/components/schemas/VIP"  
                    }  
                  }  
                }  
              }  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

```
},
"components": {
  "schemas": {
    "VIP": {
      "type": "object",
      "properties": {
        "name": {
          "type": "string"
        },
        "phone": {
          "type": "object",
          "properties": {
            "number": {
              "type": "string"
            }
          }
        }
      }
    }
  }
}
```

Example 6: Inline Arrays (OAS 3.0)

Here's an example with an inline array definition.

```
{
  "openapi": "3.0.0",
  ...
  "paths": {
    "/employees/{employeeId)": {
      "get": {
        "operationId": "getEmployee",
        "parameters": [
          {
            "in": "path",
            "name": "employeeId",
            "required": true,
            "schema": {
              "type": "string"
            }
          }
        ]
      }
    }
  }
}
```

```
],
"responses": {
  "200": {
    "description": "",
    "content": {
      "application/json": {
        "schema": {
          "type": "object",
          "properties": {
            "employee": {
              "$ref": "#/components/schemas/Employee"
            },
            "manager": {
              "$ref": "#/components/schemas/Employee"
            },
            "team": {
              "type": "array",
              "items": {
                "$ref": "#/components/schemas/Employee"
              }
            }
          }
        }
      }
    }
  }
},
"components": {
  "schemas": {
    "Employee": {
      "type": "object",
      "properties": {
        "employeeId": {"type": "string"},
        "firstName": {"type": "string"},
        "middleName": {"type": "string"},
        "lastName": {"type": "string"},
        "dateOfHire": {"type": "date"}
      }
    }
  }
}
```

{}

Example 7: HTTP Header Parameters (OAS 3.0)

Here's how a header parameter is declared in an OpenAPI schema. In Apex or Flow, the name "`apiKey`" shows up as a string parameter. Now when you set any string value in Apex or a flow to `apiKey`, it functions as an HTTP parameter when making a callout request.

```
{  
  "openapi": "3.0.0",  
  "info": {  
    "description": "A service for checking credit for an account.",  
    "version": "1.0.0",  
    "title": "Credit Decision",  
    "termsOfService": "http://swagger.io/terms/",  
    "contact": {  
      "email": "apiteam@swagger.io"  
    },  
    "license": {  
      "name": "Apache 2.0",  
      "url": "http://www.apache.org/licenses/LICENSE-2.0.html"  
    }  
  },  
  "servers": [  
    {  
      "url": "https://<YourHostName>"  
    }  
  ],  
  "paths": {  
    "/account/lastCreditRating": {  
      "get": {  
        "summary": "Evaluates credit rating and decides what payment terms to offer.",  
        "description": "",  
        "parameters": [  
          {  
            "name": "apiKey",  
            "description": "Your API Key for calling the credit rating service",  
            "in": "header",  
            "schema": {  
              "type": "string"  
            }  
          }  
        ]  
      }  
    }  
  }  
}
```

```
        }
    ],
    "requestBody": {
        "content": {
            "application/json": {
                "schema": {
                    "$ref": "#/components/schemas/accountId"
                }
            },
            "application/xml": {
                "schema": {
                    "$ref": "#/components/schemas/accountId"
                }
            }
        },
        "description": "Specifies input parameters to calculate payment term",
        "required": true
    },
    "responses": {
        "200": {
            "description": "success",
            "content": {
                "application/xml": {
                    "schema": {
                        "$ref": "#/components/schemas/creditRating"
                    }
                },
                "application/json": {
                    "schema": {
                        "$ref": "#/components/schemas/creditRating"
                    }
                }
            }
        },
        "405": {
            "description": "Invalid input"
        }
    }
},
"components": {
    "schemas": {
```

```
        "accountId": {
            "type": "object",
            "properties": {
                "accountIdString": {
                    "type": "string"
                }
            },
            "xml": {
                "name": "accountId"
            }
        },
        "creditRating": {
            "type": "object",
            "properties": {
                "creditRatingString": {
                    "type": "string"
                }
            },
            "xml": {
                "name": "creditRating"
            }
        }
    }
}
```

Example 8: URL Encoded Form Media Type (OAS 3.0)

Request and response form data is declared with media type `application/x-www-form-urlencoded`.

```
{
  "openapi": "3.0.0",
  "info": {
    "description": "Apply here for your next mortgage",
    "version": "1.0.0",
    "title": "My Mortgage Buddy",
    "contact": {
      "email": "apiteam@swagger.io"
    }
  },
  "servers": [
    {

```

```
"url": "https://MyMortgageBuddy.org/mortgages"
}
],
"paths": {
"/apply": {
"post": {
"operationId": "applyMortgage",
"requestBody": {
"content": {
"application/x-www-form-urlencoded; charset=utf-8": {
"schema": {
"type": "object",
"properties": {
"terms": {
"description": "Desired mortgage terms",
"type": "array",
"items": {
"type": "integer"
}
},
"fullName": {
"description": "Full Name",
"type": "string"
},
"loanAmount": {
"description": "Loan amount",
"type": "number"
}
},
"required": [
"terms",
"fullName",
"loanAmount"
]
}
}
}
}
},
"responses": {
"200": {
"description": "200",
"content": {
"application/x-www-form-urlencoded; charset=utf-8": {
"schema": {
```

Example 9: allOf and additionalProperties Schema Directives (OAS 3.0)

This JSON schema definition highlights the constructs `allOf` for schema object composition and `additionalProperties` for dictionary values. The sample schema is registered as external service `MyBank` with named credential `MyBank`.

The schema defines a banking service that gets customer details for a customer ID.

- The `Customer` schema object has dictionary properties of type schema object `CreditRating`. In Apex, the class `ExternalService.MyBank_Customer` has property `properties` of type `Map<String, ExternalService_CreditRating>` with the customer's credit ratings.
 - `Phones` and `Emails` compose their own properties together with the common `allof` properties from the schema object `Contact`.

```
{
  "openapi": "3.0.0",
  "info": {
    "title": "myBank",
    "description": "Sample Banking Service with allOf and additionalProperties",
    "version": "1.0.0"
  },
  "servers": [
    {
      "url": "http://localhost:8080"
    }
  ]
}
```

```
"url": "https://api.mybank.com/v1"
}
],
"paths": {
"/customers/{customerId)": {
"get": {
"summary": "Get the customer by ID.",
"parameters": [
{
"in": "path",
"name": "customerId",
"required": true,
"schema": {
"type": "integer"
}
}
],
"responses": {
"200": {
"description": "OK",
"content": {
"application/json": {
"schema": {
"$ref": "#/components/schemas/Customer"
}
}
}
}
}
}
}
},
"components": {
"schemas": {
"Customer": {
"type": "object",
"properties": {
"id": {
"type": "integer"
},
"name": {
"type": "string"
},
"phones": {

```

```
        "type": "array",
        "items": {
            "$ref": "#/components/schemas/Phone"
        }
    },
    "emails": {
        "type": "array",
        "items": {
            "$ref": "#/components/schemas/Email"
        }
    }
},
"additionalProperties": {
"$ref": "#/components/schemas/CreditRating"
},
"required": [
    "id",
    "name"
],
"Contact": {
    "type": "object",
    "properties": {
        "primary": {
            "type": "boolean"
        },
        "timeOfDay": {
            "type": "string"
        }
    },
    "required": [
        "primary"
    ]
},
"Phone": {
    "allOf": [
{
    "$ref": "#/components/schemas/Contact"
},
{
    "type": "object",
    "properties": {
        "typeOfPhone": {

```

```
        "type": "string"
    },
    "phoneNumber": {
        "type": "string"
    }
}
]
},
"Email": {
    "allOf": [
{
    "$ref": "#/components/schemas/Contact"
},
{
    "type": "object",
    "properties": {
        "email": {
            "type": "string"
        }
    }
}
]
},
"CreditRating": {
    "type": "object",
    "properties": {
        "rating": {
            "type": "string"
        },
        "score": {
            "type": "number",
            "format": "double"
        }
    }
}
}
}
}
```

To use `allOf` composition and `additionalProperties`, see [For Examples 9: allOf Composition and additionalProperties Use in Flow with Apex Unit Tests.](#)

Example 10: allOf and Discriminator Directives (OAS 3.0)

To specify a general extensible list of contacts for a customer, the `discriminator` directive can be combined with `allOf` to declare whether a contact is an email or a phone number:

```
{  
  "openapi": "3.0.0",  
  "info": {  
    "title": "myBank",  
    "description": "Sample Banking Service with allOf and discriminator",  
    "version": "1.0.0"  
  },  
  "servers": [  
    {  
      "url": "https://api.mybank.com/v1"  
    }  
  ],  
  "paths": {  
    "/customers/{customerId)": {  
      "get": {  
        "summary": "Get the customer by ID.",  
        "parameters": [  
          {  
            "in": "path",  
            "name": "customerId",  
            "required": true,  
            "schema": {  
              "type": "integer"  
            }  
          }  
        ],  
        "responses": {  
          "200": {  
            "description": "OK",  
            "content": {  
              "application/json": {  
                "schema": {  
                  "$ref": "#/components/schemas/Customer"  
                }  
              }  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

```
        }
    },
    "components": {
        "schemas": {
            "Customer": {
                "type": "object",
                "properties": {
                    "id": {
                        "type": "integer"
                    },
                    "name": {
                        "type": "string"
                    },
                    "contacts": {
                        "type": "array",
                        "items": {
                            "$ref": "#/components/schemas/Contact"
                        }
                    }
                },
                "required": [
                    "id",
                    "name"
                ]
            },
            "Contact": {
                "type": "object",
                "discriminator": {
                    "propertyName": "contactType"
                },
                "properties": {
                    "contactType": {
                        "type": "string"
                    },
                    "primary": {
                        "type": "boolean"
                    },
                    "timeOfDay": {
                        "type": "string"
                    }
                },
                "required": [
                    "contactType",
                    "primary"
                ]
            }
        }
    }
}
```

```
        ],
    },
    "Phone": {
        "allOf": [
            {
                "$ref": "#/components/schemas/Contact"
            },
            {
                "type": "object",
                "properties": {
                    "typeOfPhone": {
                        "type": "string"
                    },
                    "phoneNumber": {
                        "type": "string"
                    }
                }
            }
        ]
    },
    "Email": {
        "allOf": [
            {
                "$ref": "#/components/schemas/Contact"
            },
            {
                "type": "object",
                "properties": {
                    "email": {
                        "type": "string"
                    }
                }
            }
        ]
    }
}
```

To use composition and polymorphic OpenAPI schema constructs, see [For Examples 10: Polymorphism with allOf and Discriminator](#).

Example 11: anyOf, oneOf, and Discriminator Directives (OAS 3.0)

`anyOf` and `oneOf` directives model types with parts of a common schema type. `anyOf` and `oneOf` can be combined with the schema construct `discriminator` for polymorphic types.

- Social security number
- The driver's license number
- The customer's first *and* last name with optional middle name

In the example, a customer can be identified by one or more of the following allowed identifiers.

The schema mandates both the customer's first and last name, or to choose any other allowed identification.

This example also highlights a variation of the Contact type polymorphism with `oneOf` and `discriminator` from Example 10 with `allOf`.

```
{  
  "openapi": "3.0.0",  
  "info": {  
    "title": "myBank",  
    "description": "Sample Banking Service with oneOf, anyOf and discriminator",  
    "version": "1.0.0"  
  },  
  "servers": [  
    {  
      "url": "https://api.mybank.com/v1"  
    }  
  ],  
  "paths": {  
    "/customers/{customerId)": {  
      "get": {  
        "summary": "Get the customer by ID.",  
        "parameters": [  
          {  
            "in": "path",  
            "name": "customerId",  
            "required": true,  
            "schema": {  
              "type": "integer"  
            }  
          }  
        ]  
      },  
      "responses": {  
        "200": {  
          "description": "Customer details",  
          "content": {  
            "application/json": {  
              "schema": {  
                "type": "object",  
                "properties": {  
                  "id": {  
                    "type": "string",  
                    "format": "uuid"  
                  },  
                  "name": {  
                    "type": "string",  
                    "format": "full-name"  
                  },  
                  "middleName": {  
                    "type": "string",  
                    "format": "middle-name"  
                  },  
                  "ssn": {  
                    "type": "string",  
                    "format": "ssn"  
                  },  
                  "driverLicense": {  
                    "type": "string",  
                    "format": "driver-license"  
                  }  
                }  
              }  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

```
"200": {
    "description": "OK",
    "content": {
        "application/json": {
            "schema": {
                "$ref": "#/components/schemas/Customer"
            }
        }
    }
},
"components": {
    "schemas": {
        "Customer": {
            "type": "object",
            "properties": {
                "id": {
                    "anyOf": [
                        {
                            "$ref": "#/components/schemas/SSN"
                        },
                        {
                            "$ref": "#/components/schemas/DriversLicense"
                        },
                        {
                            "$ref": "#/components/schemas/FullName"
                        }
                    ]
                },
                "contacts": {
                    "type": "array",
                    "items": {
                        "$ref": "#/components/schemas/Contact"
                    }
                }
            },
            "required": [
                "id"
            ]
        }
    }
}
```

```
  },
  "SSN": {
    "type": "object",
    "properties": {
      "ssn": {
        "type": "string"
      }
    },
    "required": [
      "ssn"
    ]
  },
  "DriversLicense": {
    "type": "object",
    "properties": {
      "dl": {
        "type": "string"
      }
    },
    "required": [
      "dl"
    ]
  },
  "FullName": {
    "type": "object",
    "properties": {
      "firstName": {
        "type": "string"
      },
      "middleName": {
        "type": "string"
      },
      "lastName": {
        "type": "string"
      }
    },
    "required": [
      "firstName",
      "lastName"
    ]
  },
  "Contact": {
    "oneOf": [
```

```
{  
    "$ref": "#/components/schemas/Phone"  
,  
{  
    "$ref": "#/components/schemas/Email"  
}  
],  
"discriminator": {  
    "propertyName": "contactType"  
}  
,  
"Phone": {  
    "type": "object",  
    "properties": {  
        "contactType": {  
            "type": "string"  
},  
        "typeOfPhone": {  
            "type": "string"  
},  
        "phoneNumber": {  
            "type": "string"  
}  
}  
},  
"Email": {  
    "type": "object",  
    "properties": {  
        "contactType": {  
            "type": "string"  
},  
        "email": {  
            "type": "string"  
}  
}  
}  
}
```

To use composition and polymorphic OpenAPI schema constructs, see [For Example 11 \(Open API 3.0\): AnyOf, OneOf, and Discriminator](#).

Example 12: Callbacks (OAS3.0)

External Services specify callback options through the OpenAPI 3.0 schema's `callbacks` object.

This is an example of a callback for a mortgage application process for the fictitious company Acme Mortgages.

```
openapi: 3.0.0

info:
  version: '1.0'
  title: Acme Mortgages
  description: Acme Mortgages

paths:
  # Example of synchronous operation GetApplication
  /applications/{applicationNumber}:
    get:
      operationId: GetApplication
      description: Get the mortgage application status and details
      parameters:
        - name: applicationNumber
          in: path
          required: true
          description: Mortgage Application Number
          schema:
            type: string
      - name: referenceId
        in: query
        description: Reference ID if applicable. Either as query or header
        schema:
          type: string
      - name: referenceId
        in: header
        description: Reference ID if applicable. Either as query or header
        schema:
          type: string
      responses:
        200:
          description: Mortgage application status
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/MortgageApplication'
```

```
/applications:  
  # Example of asynchronous operation with callback SubmitApplication  
  post:  
    operationId: SubmitApplication  
    description: Submit a new mortgage application  
    requestBody:  
      content:  
        application/json:  
          schema:  
            type: object  
            properties:  
              applicant:  
                $ref: '#/components/schemas/Contact'  
              object:  
                $ref: '#/components/schemas/Contact'  
    statusUpdateRequest:  
      $ref: '#/components/schemas/ApplicationStatusUpdateRequest'  
    callbackUrl:  
      type: object  
      properties:  
        outcome:  
          type: object  
          properties:  
            approved:  
              type: string  
            rejected:  
              type: string  
        documentation:  
          type: string  
        outcomeError:  
          type: string  
    responses:  
      200:  
        description: Mortgage loan application submission response  
        content:  
          application/json:  
            schema:  
      type: object  
      properties:  
        applicationNumber:  
          type: string  
    400:
```

```
description: Mortgage application error
content:
  application/json:
    schema:
      $ref: '#/components/schemas/MortgageApplicationError'
callbacks:
applicationOutcome:
'${request.body#/callbackUrl/outcome/approved}':
post:
  operationId: ApplicationApproved
requestBody:
  required: true
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/MortgageApplication'
responses:
  200:
    description: Approved mortgage application callback accepted
'${request.body#/callbackUrl/outcome/rejected}':
post:
  operationId: ApplicationRejected
requestBody:
  required: true
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/MortgageApplication'
responses:
  200:
    description: Rejected mortgage application callback accepted
applicationDocumentation:
'${request.body#/callbackUrl/documentation}':
post:
  parameters:
    - in: query
      name: applicationNumber
      schema:
        type: string
  requestBody:
    required: true
    content:
```

```
application/json:  
  schema:  
    $ref: '#/components/schemas/MortgageApplicationDocumentation'  
  responses:  
    200:  
      description: Mortgage application documentation callback accepted  
  applicationStatus_Update:  
    $ref: '#/components/callbacks/ApplicationStatus_Update'  
  applicationOutcomeError:  
    $ref: '#/components/callbacks/applicationOutcomeError'  
  
servers:  
  - url: '/'  
  
components:  
  schemas:  
    MortgageApplication:  
      required:  
        - applicationNumber  
        - status  
      properties:  
        applicationNumber:  
          type: string  
        status:  
          description: One of pending, approved, rejected  
          type: string  
        requestedAmount:  
          type: number  
        approvedAmount:  
          type: number  
        appliedOn:  
          type: string  
          format: date-time  
        updatedOn:  
          type: string  
          format: date-time  
        applicant:  
          $ref: '#/components/schemas/Contact'  
        object:  
          $ref: '#/components/schemas/Contact'  
  
  ApplicationStatusUpdateRequest:  
    description: Mortgage application status update request
```

```
type: object
properties:
  sendStatusUpdates:
    type: boolean
  statusUpdateCallbackUrl:
    type: string

MortgageApplicationDocumentation:
  description: Required mortgage documentation
  type: array
  items:
    type: object
    properties:
      documentType:
        type: string
      uploadUrl:
        type: string
      instructions:
        type: string

Contact:
  type: object
  properties:
    name:
      type: string
    address:
      type: string

MortgageApplicationError:
  properties:
    errorMessage:
      type: string
    applicationNumber:
      type: string

callbacks:
ApplicationStatus_Update:
'{${request.body#/statusUpdateRequest/statusUpdateCallbackUrl}}':
  post:
    requestBody:
      required: true
      content:
        application/json:
```

```

schema:
  type: object
  properties:
    applicationNumber:
      type: string
    status:
      type: string
    updateMessage:
      type: string
  responses:
    200:
      description: Mortgage application status update callback accepted
    applicationOutcomeError:
      '$request.body#/callbackUrl/outcomeError':
        post:
          requestBody:
            required: true
            content:
              application/json:
                schema:
                  $ref: '#/components/schemas/MortgageApplicationError'
        responses:
          200:
            description: Mortgage application callback error accepted

```

Example 13: File Upload and Download (OAS 3.0)

This spec contains these examples:

- A PUT operation for file upload—the PUT operation takes a `key` parameter with the location/file name of the file to upload and specifies the `content-type` header. The `requestBody` is defined as a binary string, and contains the contents of the file being uploaded.
- A GET operation for file download—the GET operation takes a `key` parameter with the file name of the file to download. The 200 response returns a `content` object defined as a binary string, and contains the contents of the file being downloaded.

This spec references uploading a file to Amazon S3, but the spec can be used to upload files to any system for which you have a named credential configured.

```

openapi: 3.0.0
info:
  title: S3 PutObject and GetObject API

```

```
version: 1.1.0
description: Example API for uploading and downloading files to Amazon S3 (without auth headers)

paths:
/{key}:
  put:
    summary: Upload an object to S3 at the specified key
    description: Upload a file to the given bucket and key using HTTP PUT.
    operationId: putObject
    parameters:
      - name: key
        in: path
        required: true
        description: The key (path) of the S3 object
        schema:
          type: string
      - name: Content-Type
        in: header
        description: MIME type of the uploaded file
        schema:
          type: string
    requestBody:
      required: true
      content:
        application/octet-stream:
          schema:
            type: string
            format: binary
          description: Binary content of the file to upload
    responses:
      '200':
        description: Upload successful
        headers:
          ETag:
            description: The ETag of the uploaded object
            schema:
              type: string
      '400':
        description: Bad request
      '404':
        description: Bucket or key not found
      '500':
```

```
description: Internal server error

get:
  summary: Download an object from S3 at the specified key
  description: Download the file content from the given bucket and key using HTTP GET.
  operationId: getObject
  parameters:
    - name: key
      in: path
      required: true
      description: The key (path) of the S3 object
      schema:
        type: string
  responses:
    '200':
      description: File content returned
      content:
        application/octet-stream:
          schema:
            type: string
            format: binary
    '403':
      description: Access denied
    '404':
      description: Object not found
    '500':
      description: Internal server error
```

To use this functionality, see [For Example 13 \(Open API 3.0\): Binary File Upload](#) and [For Example 13 \(Open API 3.0\): Binary File Download](#).

See Also

[Schema Examples](#)

[Using the Schema Examples](#)

Using the Schema Examples

See how the respective Open API 2.0 and 3.0 examples are implemented with Apex.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

For Examples 9: allOf Composition and additionalProperties Use in Flow with Apex Unit Tests

The External Services registration `MyBank` from Example 9 (from both OpenAPI 2.0 and 3.0 examples) is invoked by a sample flow that accesses the dictionary properties with an Apex invocable action. A flow Apex unit test ties it all together.

```
public class MyBankGetCreditRatings {  
    @InvocableMethod(  
        label='Get Credit Ratings'  
        description='A list of credit ratings for a customer'  
        category='MyBank'  
    )  
    public static List<CustomerCreditRatings>  
    getCreditRatings(List<Customer> inCustomers) {  
  
        List<CustomerCreditRatings> outCreditRatingsList =  
            new List<CustomerCreditRatings>();  
        for (Customer inCustomer: inCustomers) {  
            ExternalService.MyBank_Customer customer = inCustomer.customer;  
            CustomerCreditRatings outCreditRatings = new CustomerCreditRating  
                s();  
            outCreditRatings.customerId = customer.id;  
            outCreditRatings.creditRatings =  
                new List<ExternalService.MyBank_CreditRating>();  
  
            Map<String, ExternalService.MyBank_CreditRating> creditRatings =  
                customer.properties;  
            for (String ratingProperty: creditRatings.keySet()) {  
                ExternalService.MyBank_CreditRating creditRating =  
                    creditRatings.get(ratingProperty);  
                outCreditRatings.creditRatings.add(creditRating);  
            }  
  
            outCreditRatingsList.add(outCreditRatings);  
        }  
  
        return outCreditRatingsList;  
    }  
  
    public class Customer {
```

```

@InvocableVariable(
    label='Customer'
    description='Banking customer'
    required=true
)
public ExternalService.MyBank_Customer customer;

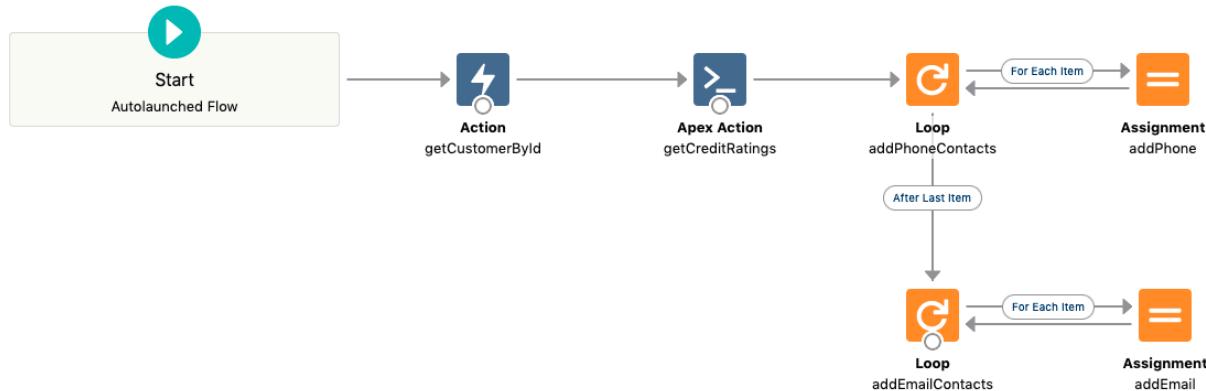
}

public class CustomerCreditRatings {
    @InvocableVariable(
        label='Customer ID'
        description='Bank customer ID'
        required=true
    )
    public Integer customerId;

    @InvocableVariable(
        label='Credit Ratings'
        description='Credit ratings for a customer'
        required=true
    )
    public List<ExternalService.MyBank_CreditRating> creditRatings;
}
}

```

The flow starts by calling the external `MyBank` service action `getCustomerById` to get the customer details given a customer ID. The Apex invokable flow action `getCreditRatings` gets the list of credit ratings from the customer details. Phone and email contact details are formatted as a list of customer contacts by looping through the `phones` and `emails` properties and assigning the formatted value to the `contacts` list.



The HTTP callout mock asserts an expected request and responds with a sample customer as

application/json:

```
public class MyBankGetCustomerCalloutMock implements HttpCalloutMock {
    public HttpResponse respond(HTTPRequest request) {
        // Assert expected request test data: customer ID in the request path
        System.assertEquals('GET', request.getMethod());
        System.assertEquals('callout:MyBank/v1/customers/42', request.getEndpoint());
        // Send response test data: customer details with sample ratings
        // as additional properties and sample contacts
        HttpResponse response = new HttpResponse();
        response.setHeader('Content-Type', 'application/json');
        response.setBody('{' +
            '"id": 42, "name": "Foo Bar", "phones": [' +
            ' {"primary": true, "timeOfDay": "Daytime", ' +
            '     "typeOfPhone": "Mobile", "phoneNumber": "555-5555"}, ' +
            ' {"primary": false, "timeOfDay": "Evening", ' +
            '     "typeOfPhone": "Landline", "phoneNumber": "222-5555"}' +
            '], "emails": [' +
            ' {"primary": true, "timeOfDay": "AllDay", "email": "fooBar@acme.or' +
            'g"}' +
            '], ' +
            '"rating1": {"rating": "Rating 1", "score": 0.95}, ' +
            '"rating2": {"rating": "Rating 2", "score": 0.78}' +
        '}');
        response.setStatusCode(200);
        return response;
    }
}
```

 **Tip** You can also use JSON to serialize a sample response matching the external service response type if the HTTP response media type is `application/json` and the JSON properties are compliant with Apex identifier characters:

```
ExternalService.MyBank_Customer customer = new ExternalService.MyBank_Customer();
customer.id = 42;
...
customer.properties = new Map<String, ExternalService.MyBank_CreditRating>();
ExternalService.MyBank_CreditRating creditRating = new ExternalService.MyBank_CreditRating();
creditRating.rating = 'Rarging 1';
creditRating.score = 0.95;
```

```
customer.properties.put('rating1', creditRating);
...
response.setBody(System.JSON.serialize(customer));
....
```

The Apex unit test sets up the HTTP callout mock and asserts the expected credit ratings and customer contacts:

```
@IsTest
public class MyBankFlowTest {
    @IsTest
    static public void testGetCustomer() {
        // Set HTTP callout mock to match flow's external service action invocation
        Test.setMock(HttpCalloutMock.class, new MyBankGetCustomerCalloutMock());

        // Set flow input variables and create the flow interview
        Map<String, Object> inputVariables = new Map<String, Object>();
        inputVariables.put('customerId', 42);
        Flow.Interview myBankFlow = Flow.Interview.createInterview('MyBank', inputVariables);

        // Start flow interview with set input variables
        myBankFlow.start();

        // Assert customer's expected credit ratings
        List<ExternalService.MyBank_CreditRating> creditRatings =
            (List<ExternalService.MyBank_CreditRating>)myBankFlow.
                getVariableValue('creditRatings');
        System.assertEquals(2, creditRatings == null ? 0 : creditRatings.size());
        ExternalService.MyBank_CreditRating actualRating = creditRatings.get(1);
        ExternalService.MyBank_CreditRating expectedRating =
            new ExternalService.MyBank_CreditRating();
        expectedRating.rating = 'Rating 2';
        expectedRating.score = 0.78;
        System.assertEquals(expectedRating.toString(), actualRating.toString());

        // Assert customer's contacts:
        List<String> contacts = (List<String>)myBankFlow.getVariableValue('*co
```

```
ntacts*');
    System.assertEquals(3, contacts == null ? 0 : contacts.size());
    System.assertEquals('Phone Number: 555-5555', contacts.get(0));
    System.assertEquals('Phone Number: 222-5555', contacts.get(1));
    System.assertEquals('Email: fooBar@acme.org', contacts.get(2));
}
}
```

For another example of a flow Apex unit test, see [Testing External Services](#).

For Examples 9: allOf Composition and additionalProperties Use in Apex with Apex Unit Tests

The External Services registration `MyBank` from Example 9 (from both OpenAPI 2.0 and 3.0 examples) is invoked by a sample Apex class that accesses the dictionary properties. An Apex unit test ties it all together.

The class `CustomerCreditRating` captures the customer's detail suitable for further processing. It's possible to directly use the external service's response output data structure. As a good practice, decouple your business relevant data structure from external dependencies:

```
public class CustomerCreditRating {
    public Integer Id {get; private set;}
    public String Name {get; private set;}

    private List<String> emails;
    private List<String> phoneNumbers;
    private Map<String, Integer> ratings;

    public CustomerCreditRating(Integer customerId, String name) {
        this.Id = customerId;
        this.Name = name;
        this.emails = new List<String>();
        this.phoneNumbers = new List<String>();
        this.ratings = new Map<String, Integer>();
    }

    public void addEmail(String email) {
        emails.add(email);
    }

    public void addPhoneNumber(String phoneNumber) {
        this.phoneNumbers.add(phoneNumber);
    }
}
```

```

}

public List<String> getContacts() {
    List<String> contacts = new List<String>();
    for (String phoneNumber: phoneNumbers) {
        contacts.add('Phone Number: ' + phoneNumber);
    }
    for (String email: emails) {
        contacts.add('Email: ' + email);
    }
    return contacts;
}

public void addRating(String ratingType, Integer ratingScore) {
    ratings.put(ratingType, ratingScore);
}

public Set<String> getRatingTypes() {
    return ratings.keySet();
}

public Integer getRatingScore(String ratingType) {
    return ratings.get(ratingType);
}
}

```

The Apex class `MyBankCustomerCreditRating` starts by calling the external `MyBank` service action `getCustomerById` to get the customer details given a customer ID. The Apex method `getCreditRating` gets the credit rating from the customer details. Phone and email contact details are formatted as a list of customer contacts by looping through the `phones` and `emails` properties and assigning the formatted value to the contacts list:

```

public class MyBankCustomerCreditRating {
    public class MyBankException extends Exception {}

    public CustomerCreditRating getCreditRating(Integer customerId) {
        // Get customer credit rating from an external bank rating service
        // Construct the external service registration MyBank
        ExternalService.MyBank myBank = new ExternalService.MyBank();

        // Make the callout to get the customer by ID.
        // The response is the customer detail for HTTP code 200
        ExternalService.MyBank_Customer customer;
    }
}

```

```

try {
    ExternalService.MyBank.getCustomersByCustomerId_Request request =
        new ExternalService.MyBank.getCustomersByCustomerId_Request();
    request.customerId = customerId;
    customer = myBank.getCustomersByCustomerId(request).Code200;
} catch (ExternalService.MyBank.getCustomersByCustomerId_ResponseException e) {
    // An HTTP failure code is thrown as exception -
    // captured and translated to a meaningful error
    throw new MyBankException(
        'Credit rating not available for customer ID: '
        + customerId);
}

// Gather the customer name, contacts and credit ratings
// from the callout's response data
CustomerCreditRating customerRating =
    new CustomerCreditRating(customerId, customer.name);
for (ExternalService.MyBank_Email email: customer.emails) {
    customerRating.addEmail(email.email);
}
for (ExternalService.MyBank_Phone phone: customer.phones) {
    customerRating.addPhoneNumber(phone.phoneNumber);
}
for (String ratingType: customer.properties.keySet()) {
    ExternalService.MyBank_CreditRating rating =
        customer.properties.get(ratingType);
    Integer ratingPercent = (Integer)(rating.score * 100.0);
    customerRating.addRating(ratingType, ratingPercent);
}

return customerRating;
}
}

```

You can share the same HTTP mock callout class for your Apex integration. The corresponding Apex unit test class tests the Apex credit rating logic:

```

@IsTest
public class MyBankCustomerRatingTest {
    @IsTest
    static public void testGetCustomerRating() {
        // Set HTTP callout mock to match Apex's external service callout
        Test.setMock(HttpCalloutMock.class, new MyBankGetCustomerCalloutMoc

```

```

k()));

        // Call the Apex MyBankCustomerRating class
        MyBankCustomerCreditRating myBankCreditRating = new MyBankCustomerCreditRating();
        CustomerCreditRating creditRating = myBankCreditRating.getCreditRating(42);

        // Assert customer's expected credit ratings
        System.assertEquals(2, creditRating.getRatingTypes().size());
        Integer actualRatingScore = creditRating.getRatingScore('rating2');
        Integer expectedRatingScore = 78;
        System.assertEquals(expectedRatingScore, actualRatingScore);

        // Assert customer's contacts:
        List<String> contacts = creditRating.getContacts();
        System.assertEquals(3, contacts.size());
        System.assertEquals('Phone Number: 555-5555', contacts.get(0));
        System.assertEquals('Phone Number: 222-5555', contacts.get(1));
        System.assertEquals('Email: fooBar@acme.org', contacts.get(2));
    }
}

```

For Examples 10: Polymorphism with allOf and Discriminator

The `discriminator` directive can be combined with `allOf` to define the composition's polymorphic type. Polymorphic types are marked with the suffix `_KT_PT` in the Apex object name. Work with polymorphic extension types through its polymorphic object type.

This example illustrates how to interact with polymorphic types in Apex with the OpenAPI spec from Example 10. `Contact` is the base type. `Phone` and `Email` are polymorphic extension types modeling contact types that can be assigned to a list of contacts for a customer:

```

// The customer
ExternalService.MyBank_Customer customer = new ExternalService.MyBank_Customer();

// The primary phone contact wrapped as polymorphic type Contact_KT_PT
ExternalService.MyBank_Phone mobile = new ExternalService.MyBank_Phone();
mobile.primary = true;
mobile.typeOfPhone = 'Mobile';
mobile.phoneNumber = '555-5555';
ExternalService.MyBank_Contact_KT_PT cMobile =new ExternalService.MyBank_Contact_K

```

```

T_PT();
cMobile.phone = mobile; // cMobile is a phone contact

// Customer's secondary home phone contact
ExternalService.MyBank_Phone home = new ExternalService.MyBank_Phone();
home.primary = false;
home.typeOfPhone = 'Home';
home.phoneNumber = '444-4444';
ExternalService.MyBank_Contact_KT_PT cHome = new ExternalService.MyBank_Contact_KT_PT();
cHome.phone = home; // cHome is a phone contact

// Customer's email
ExternalService.MyBank_Email email = new ExternalService.MyBank_Email();
email.primary = true;
email.email = 'someone@somewhere.org';
ExternalService.MyBank_KT_PT cEmail = new ExternalService.MyBank_Contact_KT_PT();
cEmail.email = email; // cEmail is an email contact

// Adding mobile, home phone and email as contacts to the customer contacts list
customer.contacts = new List<ExternalService.MyBank_Contact_KT_PT>();
customer.contacts.add(cMobile);
customer.contacts.add(cHome);
customer.contacts.add(cEmail);

// Send an email to a customer's primary email contacts
for (ExternalService.MyBank_Contact_KT_PT contact: customer.contacts) {
    if (contact.email != null && contact.email.primary) {
        String emailAddress = contact.email.email;
        // Sending email to email address
        ...
    }
}

```

For Example 11 (Open API 3.0): AnyOf, OneOf, and Discriminator

`oneOf` or `anyOf` define the composition's type - either one of the schema constructs can be used or any of them. Composition schema types can be accessed through the corresponding composition type's properties as follows:

- A named schema `Name` referenced as composition schema: `anyOfName` or `oneOfName`.
- An inline composition schema object: `anyOfObject` or `oneOfObject`. If more than one inline

object is declared in the composition, then the declaration order is added as a sequence number suffix. For example, `oneOfObject1`, `oneOfObject2` for first and second composition type respectively.

- An inline composition schema array: `anyOfArray` or `oneOfArray`. If more than one inline array is declared in the composition, then the declaration order is added as a sequence number suffix. For example `oneOfArray1`, `oneOfArray2`.
- An inline composition primitive type: `anyOfTypeName` or `oneOfTypeName`. If the same type is referenced in the inline composition, then like types are suffixed with their declaration order in the spec - for example `anyOfString1`, `anyOfString2`.

This example illustrates how to interact with `anyOf` and `oneOf` types in Apex with the OpenAPI spec from Example 11. `Contact` is the base type. `Phone` and `Email` are polymorphic extension types modeling contact types that can be assigned to a list of contacts for a customer. A customer can be identified by any one of social security number, driver's license or first and last name and optional middle name:

```
ExternalService.MyBank_Customer customer = new ExternalService.MyBank_Customer();  
  
// Setting the customer ID  
customer.id = new ExternalService.MyBank_Customer_id();  
// Setting the social security security number  
customer.id.anyOfSSN.ssn = '555-55-5555';  
// Setting the customer's first and last name without a middle name  
customer.id.anyOfFullName = new ExternalService.MyBank_FullName();  
customer.id.anyOfFullName.firstName = 'Somefirstname';  
customer.id.anyOfFullName.lastName = 'Somelastname';  
  
// Customer contacts  
ExternalService.MyBank_Phone mobile = new ExternalService.MyBank_Phone();  
mobile.typeOfPhone = 'Mobile';  
mobile.phoneNumber = '555-5555';  
ExternalService.MyBank_Contact cMobile = new ExternalService.MyBank_Contact();  
cMobile.oneOfPhone = mobile;  
  
ExternalService.MyBank_Phone home = new ExternalService.MyBank_Phone();  
home.typeOfPhone = 'Home';  
home.phoneNumber = '444-4444';  
ExternalService.MyBank_Contact cHome = new ExternalService.MyBank_Contact();  
cHome.oneOfPhone = home;  
  
ExternalService.MyBank_Email email = new ExternalService.MyBank_Email();  
email.email = 'someone@somewhere.org';  
ExternalService.MyBank_Contact cEmail = new ExternalService.MyBank_Contact();
```

```
cEmail.oneOfEmail = email;

customer.contacts = new List<ExternalService.MyBank_Contact>();
customer.contacts.add(cMobile);
customer.contacts.add(cHome);
customer.contacts.add(cEmail);

// Sending the customer's known identities to the customer's email address
String emailContact = null;
for (ExternalService.MyBank_Contact contact: customer.contacts) {
    if (contact.oneOfEmail != null) {
        emailContact = contact.oneOfEmail.email;
    }
}
if (emailContact != null) {
    String subject = 'Your identity';
    String body = 'These are the identities we\'ve found: \n';
    if (customer.id.anyOfSSN != null) {
        body += ' - Social Security Number: ' + customer.id.anyOfSSN.ssn +
    '\n';
    }
    if (customer.id.anyOfDriversLicense != null) {
        body += ' - Driver\'s License: ' + customer.id.anyOfDriversLicense.dl +
    '\n';
    }
    if (customer.id.anyOfFullName != null) {
        body += ' - First name: ' + customer.id.anyOfFullName.firstName +
    '\n';
        if (customer.id.anyOfFullName.middleName != null) {
            body += '     Middle name: ' + customer.id.anyOfFullName.middleName +
    '\n';
        }
        body += '     Last name: ' + customer.id.anyOfFullName.lastName + '\n';
    }
    ...
}
```

For Example 13 (Open API 3.0): Binary File Upload

The example Apex types in this section refer to [Example 13: File Upload and Download \(OAS 3.0\)](#). First, you register an external service with a PUT operation that contains the name of the file to upload and a binary `requestBody`. Here, the external service registration in the org is named `s3`. This example first creates an instance of the `s3` external service and then creates an instance of the `putObject` operation. It sets these input parameter values on the `request` object.

- `key` –The name of the file after it's uploaded to the external system.
- `Contentx2dType` –The content type.
- `body` –The ID of the file (stored as a ContentDocument) in the org.

You can test this snippet by running it in the Developer Console. The file hello.jpeg will be uploaded to the external location.

```
// Upload File
ExternalService.s3 fileService = new ExternalService.s3();
ExternalService.s3.putObject_Request request = new ExternalService.s3.putObject_Request();
request.key = 'hello.jpeg';
request.Contentx2dType = 'image/jpeg';
request.body = '123ABC00000123PXYZ';

try {
    ExternalService.s3.putObject_Response response = fileService.putObject(request);

    if (response.responseCode == 200) {
        System.debug('Success | ' + response);
    }
} catch (ExternalService.s3.putObject_ResponseException e) {
    if (e.responseCode == 400) {
        System.debug('Bad request: ' + e.Code400);
    } else if (e.responseCode == 404) {
        System.debug('Object not found: ' + e.Code404);
    } else if (e.responseCode == 500) {
        System.debug('Internal server error: ' + e.Code500);
    } else {
        System.debug('Unexpected error: ' + e.defaultResponse);
    }
}
```

For Example 13 (Open API 3.0): Binary File Download

The example Apex types in this section refer to [Example 13: File Upload and Download \(OAS 3.0\)](#). First, you register an external service with a GET operation that contains the name of the file to download and a binary `content` object in the response. Here, the external service registration in the org is named `s3`. This example first creates an instance of the `s3` external service and then creates an instance of the `getObject` operation. It sets these input parameter values on the `request` object.

- `key` –The name of the file to download from the external system.

You can test this snippet by running it in the Developer Console. The file test.jpeg will be downloaded from the external location.

```
// Download File
ExternalService.s3 fileService = new ExternalService.s3();
ExternalService.s3.GetObject__Request request = new ExternalService.s3.GetObject__Request();
request.key = 'test.jpeg';

try {
    ExternalService.s3.GetObject__Response response = fileService.GetObject(request);

    if (response.responseCode == 200) {
        System.debug('Success | ' + response);
    }
} catch (ExternalService.s3.GetObject__ResponseException e) {
    if (e.responseCode == 403) {
        System.debug('Access denied: ' + e.Code403);
    } else if (e.responseCode == 404) {
        System.debug('Object not found: ' + e.Code404);
    } else if (e.responseCode == 500) {
        System.debug('Internal server error: ' + e.Code500);
    } else {
        System.debug('Unexpected error: ' + e.defaultResponse);
    }
}
```

See Also

[External Services OpenAPI 2.0 Schema](#)

[External Services OpenAPI 3.0 Schema](#)

Access External Data With Salesforce Connect

Salesforce Connect lets your users view, search, and modify data that's stored outside your Salesforce org. Instead of copying the data into standard or custom objects, use external objects to access the data in real time via web service callouts.

[Salesforce Connect](#)

Salesforce Connect provides seamless integration of data across system boundaries by letting your users view, search, and modify data that's stored outside your Salesforce org. For example, perhaps you have data that's stored on premises in an enterprise resource planning (ERP) system. Instead of copying the data into your org, you can use external objects to access the data in real time via web service callouts.

External Data Sources With Salesforce Connect

Salesforce Connect uses external data sources to access data that's stored outside your Salesforce org. You must configure an external data source and synchronize it to map its tables with external objects in Salesforce.

Salesforce Platform Features Supported by Salesforce Connect

Salesforce Connect provides seamless integration with the Salesforce Platform and enables users to view, search, and modify external object data.

Access Data in Another Salesforce Org with the Cross-Org Adapter for Salesforce Connect

Connect your users to data that's stored in another Salesforce org.

Access External Data with OData Adapters for Salesforce Connect

Connect your users to data that's exposed via the Open Data Protocol.

Access External Data with a Custom Adapter for Salesforce Connect

Connect your users to any data anywhere by developing your own custom adapter with the Apex Connector Framework.

Access External Data with the Salesforce Connect Adapter for Amazon DynamoDB

Connect your users to data that's stored in Amazon DynamoDB.

Access External Data with the Salesforce Connect Adapter for SQL

Connect your users to external data sources that expose their capabilities via REST APIs and offer query and DML operations using SQL.

Access External Data with the Salesforce Connect Adapter for GraphQL

Connect your users to external data sources that expose their capabilities via GraphQL.

Access External Data with the Salesforce Connect Adapter for Salesforce Archive

Connect your users to data that's stored in a Salesforce Archive data store.

Access External Data with the Salesforce Connect Adapter for Privacy Center

Connect your users to data that's stored in a Privacy Center retention store.

Salesforce Connect

Salesforce Connect provides seamless integration of data across system boundaries by letting your users view, search, and modify data that's stored outside your Salesforce org. For example, perhaps you have data that's stored on premises in an enterprise resource planning (ERP) system. Instead of copying the data into your org, you can use external objects to access the data in real time via web service callouts.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer Edition**

Available for an extra cost in: **Enterprise, Performance, and Unlimited** Editions

Traditionally, we've recommended importing or copying data into your Salesforce org to let your users access that data. For example, extract, transform, and load (ETL) tools can integrate third-party systems with Salesforce. However, doing so copies data into your org that you don't need or that quickly becomes stale.

In contrast, Salesforce Connect maps Salesforce *external objects* to data tables in external systems. Instead of copying the data into your org, Salesforce Connect accesses the data on demand and in real time. The data is never stale, and we access only what you need. We recommend that you use Salesforce Connect when:

- You have a large amount of data that you don't want to copy into your Salesforce org.
- You need small amounts of data at any one time.
- You want real-time access to the latest data.

Even though the data is stored outside your org, Salesforce Connect provides seamless integration with the Lightning Platform. External objects are available to Salesforce tools, such as global search, lookup relationships, record feeds, and the Salesforce mobile app. External objects are also available to Apex, SOSL, SOQL queries, Salesforce APIs, and deployment via the Metadata API, change sets, and packages.

For example, you store product order information in a back-office ERP system. You want to view those orders as a related list on each customer record in your Salesforce org. Salesforce Connect enables you to set up a lookup relationship between the customer object (parent) and the external object (child) for orders. Then you can set up the page layouts for the parent object to include a related list that displays child records.

Going a step further, you can update the orders directly from the related list on the customer record. By default, external object records are read only. But you can define the external data source to enable writable external objects.

For information about using Apex DML write operations on external object records, see the [Apex Developer Guide](#).

 **Note** If transmitting potentially sensitive information, including sensitive or regulated data such as personal health data or financial data, Salesforce recommends that customers encrypt their external data sources at rest. Transmissions through the Salesforce Connect service are already encrypted using mTLS.

 **Example** This screenshot shows how Salesforce Connect can provide a seamless view of data across system boundaries. A record detail page for the Business_Partner external object includes two related lists of child objects. The external lookup relationships and page layouts enable users to view related data from inside and from outside the Salesforce org on a single page.

- Account standard object (1)
- Sales_Order external object (2)

The screenshot shows a Salesforce page for a Business Partner with ID 1000000. The top navigation includes links for 'Customize Page', 'Edit Layout', 'Help for this Page', 'Accounts [1]', and 'Sales_Orders [2]'. The main section, 'Business_Partner Detail', lists various fields: Business_Partner_ID (1,000,000), City (Munich), Company_Name (EcoTech), Country (Germany), Currency_Code (EUR), Email_Address (robert.stamps@trainingorg-echotech-germany.com), Fax (49-00-88766-0), and Website (http://www.Ecotech.net). Below this is a 'Accounts' section (labeled 1) with a table showing one record for 'Acme' with Billing City 'New York' and Phone '(212) 555-5555'. At the bottom is a 'Sales_Orders' section (labeled 2) with a table showing two records: one for External ID 500000 with Sales_Order_ID 50000000 and Customer_Name EcoTech, and another for External ID 500009 with Sales_Order_ID 50000009 and Customer_Name EcoTech.

Action	Account Name	Billing City	Phone
Edit Del	Acme	New York	(212) 555-5555

Action	External ID	Sales_Order_ID	Customer_Name	Total_Sum	Currency
	500000	50000000	EcoTech	26,581.03	EUR
	500009	50000009	EcoTech	3,972.22	EUR

Choose Which Salesforce Connect Adapter to Use

Salesforce Connect uses a protocol-specific adapter to connect to an external system and access its data. When you define an external data source in your organization, you specify the adapter in the **Type** field.

Salesforce Connect Adapters Included per Add-On License

Using Salesforce Connect to access external data in an org requires one or more Salesforce Connect add-on licenses.

General Limits for Salesforce Connect

Understand the limits that apply to all Salesforce Connect adapters.

Salesforce Connect Learning Map

Follow along the Salesforce Connect learning map to explore all the relevant trails, docs, and reference resources you need to integrate external data sources into your apps!

See Also

[External Data Sources With Salesforce Connect](#)

[External Object Relationships](#)

[Salesforce Platform Features Supported by Salesforce Connect](#)

Choose Which Salesforce Connect Adapter to Use

Salesforce Connect uses a protocol-specific adapter to connect to an external system and access its data. When you define an external data source in your organization, you specify the adapter in the **Type** field.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

These adapters are available for Salesforce Connect.

Salesforce Connect Adapter	Description	When to Use
Cross-org	Uses the Lightning Platform REST API to access data that's stored in other Salesforce orgs.	To connect data between your Salesforce orgs. For example, provide your service representatives a unified view of customer transactions by integrating data from different Salesforce orgs.
OData 2.0 OData 4.0 OData 4.01	Uses Open Data Protocol to access data that's stored outside Salesforce. The external data must be exposed via OData producers.	To integrate external data sources into your org that support the ODATA protocol and publish an OData provider. For example, give your account executives a unified data view by pulling data from legacy systems such as SAP and Microsoft in real time.
Custom adapter created via Apex	Uses the Apex Connector Framework to develop your own custom adapter when the other available adapters aren't suitable for your needs. A custom adapter can obtain data from anywhere. For example, some data can be retrieved from anywhere in the Internet via callouts, while other data can be manipulated or even generated programmatically.	To develop your own adapter with the Apex Connector Framework when the other available adapters aren't suitable for your needs. For example, when you want to retrieve data via callouts from a REST API.

Salesforce Connect Adapter	Description	When to Use
Salesforce Connect Adapter for Amazon DynamoDB	<p>Connects Amazon DynamoDB data sources to Salesforce through external objects.</p> <p>The data stored in Amazon DynamoDB can use the flexible data storage option of DynamoDB and Salesforce Platform capabilities.</p>	To integrate AWS data natively with Salesforce business applications.
Salesforce Connect Adapter for SQL	<p>Connect Salesforce to external data sources that expose their capabilities via REST APIs and offer query and DML operations using SQL.</p> <p>The Salesforce Connect Adapter for SQL supports Snowflake and Amazon Athena external data sources.</p>	To integrate external data natively with Salesforce and run interactive on demand queries using SQL.
Salesforce Connect Adapter for GraphQL	Uses GraphQL APIs to provide a modern way to integrate applications.	To access and integrate data from external sources that expose their capabilities via GraphQL, including Amazon RDS via AWS AppSync.

See Also

[Salesforce Connect Adapters Included per Add-On License](#)

[Salesforce Platform Features Supported by Salesforce Connect](#)

Salesforce Connect Adapters Included per Add-On License

Using Salesforce Connect to access external data in an org requires one or more Salesforce Connect add-on licenses.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Each Salesforce Connect add-on license includes a set number of connections per adapter type. A Salesforce Connect add-on license is associated with a single Salesforce org.

Salesforce Connect Adapter	Number of Connections per License
Cross-org	5
OData 2.0, OData 4.0, OData 4.01, a custom adapter, Salesforce Connect adapter for Amazon DynamoDB, Salesforce Connect adapter for SQL, or Salesforce Connect adapter for GraphQL.	1

The number of Salesforce Connect add-on licenses that you need depends on where the data is stored and which type of connections are required. Some common data integration scenarios and the number of licenses required:

- Your primary Salesforce org accesses data stored in six other Salesforce orgs. You need two Salesforce Connect add-on licenses. Both Salesforce Connect add-on licenses are assigned to the primary org.
- Your primary Salesforce org accesses data stored in one secondary Salesforce org. The secondary org accesses the primary org data. You need two Salesforce Connect add-on licenses, one for each org.
- Your primary Salesforce org accesses data stored in five other Salesforce orgs and one external data source. You need one Salesforce Connect add-on license, which is assigned to the primary org.

See Also

- [Choose Which Salesforce Connect Adapter to Use](#)
[Salesforce Platform Features Supported by Salesforce Connect](#)

General Limits for Salesforce Connect

Understand the limits that apply to all Salesforce Connect adapters.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Limits

Maximum new rows retrieved or created per hour.	100,000
---	---------

This limit doesn't apply to:	
<ul style="list-style-type: none"> Orgs hosted on Hyperforce¹ High data volume external data sources 	
Maximum external objects per org.	200
If the default value for external objects in your org is 100, create a support case to increase your org limit to 200.	
External objects and custom objects have different limits. External objects don't count towards custom object usage.	
Maximum joins per query across external objects and other types of objects. ²	4
Maximum length of the OAuth token issued by the external system.	4,000 characters
Maximum page size for server-driven paging.	2,000 rows

¹ Starting in the Spring '25 release, this functionality is available to orgs hosted on Hyperforce on a rolling basis.

² This can affect reports and dashboards.

Considerations

Salesforce Connect accesses the external data in real time via web service callouts to external data sources, and there are no callout limits. However, the external data source may be limited in how many callouts it can accept in a given time period. Keep in mind any API limits enforced by external systems.

- OData 2.0, OData 4.0, and OData 4.01 adapters: No callout limits. However, the external system may have its own limits on how much traffic it can accept.
- Custom adapter: Callouts are limited in Developer Edition orgs. See [Callout Limits and Limitations](#) and [Execution Governors and Limits](#) in the *Apex Developer Guide*.
- Amazon DynamoDB adapter, GraphQL adapter, and SQL adapter for both Amazon Athena and Snowflake: No callout limits. However, callouts can incur charges from the hosting provider of the external data source.
- Cross-org adapter: Each callout counts towards the API usage limits of the provider org. See [API Request Limits and Allocations](#).

When a Salesforce Connect adapter sends a query for a field value that's equal to an empty string, the query is converted to filter for a null value in the external system.

See Also

[Salesforce Connect](#)

[Salesforce Platform Features Supported by Salesforce Connect](#)

Salesforce Connect Learning Map

Follow along the Salesforce Connect learning map to explore all the relevant trails, docs, and reference resources you need to integrate external data sources into your apps!

The [learning map](#) guides you through every step of your journey.

- [Get Started](#): Get a 360-degree view of your customer data within Salesforce using Salesforce Connect and experience integration of data across system boundaries.
- [Configure & Connect](#): Connect to any database and give users of Salesforce applications seamless access to handle the data stored in external data sources.
- [Integrate & Build](#): Integrate with external systems to build a single view of customer data and give that data native platform ability inside Salesforce.

See Also

[Salesforce Connect](#)

[Salesforce Platform Features Supported by Salesforce Connect](#)

External Data Sources With Salesforce Connect

Salesforce Connect uses external data sources to access data that's stored outside your Salesforce org. You must configure an external data source and synchronize it to map its tables with external objects in Salesforce.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

[Identity Type for External Data Sources](#)

On the external data source configured for Salesforce Connect, the **Identity Type** field specifies whether your organization uses one set or multiple sets of credentials to access the external system. Each set of credentials corresponds to a login account on the external system.

[Sync an External Data Source for Salesforce Connect](#)

When you validate and sync an external data source, it creates or overwrites Salesforce external objects that map to the external system's schema. Syncing doesn't copy any data into your Salesforce org or write data from your org to the external system.

External Objects in Salesforce Connect

External objects behave similar to custom objects except that they map to data stored outside Salesforce in an external data source. Each external object maps to a data table, and the object fields map to accessible table columns.

Writable External Objects in Salesforce Connect

Select **Writable External Objects** when you define an external data source and use Salesforce Connect external objects to create, update, and delete data. External objects are read only by default.

See Also

[Salesforce Connect](#)

[Salesforce Platform Features Supported by Salesforce Connect](#)

[Work with External Data Sources](#)

Identity Type for External Data Sources

On the external data source configured for Salesforce Connect, the **Identity Type** field specifies whether your organization uses one set or multiple sets of credentials to access the external system. Each set of credentials corresponds to a login account on the external system.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

If you select **Named Principal**, your organization uses only one login account on the external system.

If you select **Per User**, your organization uses multiple login accounts on the external system. Each of your users can have a unique set of credentials, or you can group your users—for example, by function or business unit—and have each group share a set of credentials. After you grant user access to the external data source through permission sets or profiles, users can set up and manage their own authentication settings for the external system.

-  **Note** For an external data source with **Per User Identity Type**, site members in Experience Cloud can't set up their own credentials. However, as an admin you can set up and manage each user's authentication settings for external systems from Lightning Experience or Salesforce Classic.

To access the external system's...	The system uses the credentials that are defined in the...	
Data (search or view external objects)	External data source definition	User's personal authentication settings for the external system
Metadata (sync to create external objects)	External data source definition	External data source definition

 **Tip** Train your users on how to set up their authentication settings for external systems. Make sure that they know which credentials to enter for each external system. If you're using OAuth 2.0, test the OAuth flow for potentially confusing prompts or redirects, and train your users as needed. OAuth flows vary, depending on your external system, authentication provider, and specified scopes.

See Also

- [Define an External Data Source for Salesforce Connect Cross-Org Adapter](#)
- [Define an External Data Source for Salesforce Connect–OData 2.0 or 4.0 Adapter](#)
- [Define an External Data Source for Salesforce Connect–Custom Adapter](#)
- [Grant Access to Authentication Settings for External Data Sources](#)
- [Store Authentication Settings for External Systems](#)

Sync an External Data Source for Salesforce Connect

When you validate and sync an external data source, it creates or overwrites Salesforce external objects that map to the external system's schema. Syncing doesn't copy any data into your Salesforce org or write data from your org to the external system.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Syncing fails if it causes your org to exceed 200 external objects. Syncing also fails if it tries to create an external object with an API name that conflicts with an existing object in the org. In such cases, determine whether the existing object is needed.

- If the object isn't needed, delete that object, and sync again.
- If the object is needed, change the API name of the existing object to no longer conflict with the table

that you're trying to sync. However, if the existing object is an external object that was previously synced, you can't resync it. Manually update the external object and its fields as needed for schema changes on the external system.

 **Tip** We recommend that you create your external data sources and external objects in a Developer Edition org. Then use managed packages to deploy the external data sources and external objects to your other orgs. Doing so prevents your external object names from conflicting with other objects in your org by applying a namespace prefix.

When an external object is created via syncing, its Deployment Status is set to In Development. When you're ready to expose the external object to users, set the status to **Deployed**.

If the external system's schema changes, the modifications aren't automatically synced to your Salesforce org. To reflect the changes in the external system, resync the objects. After resyncing, all users who have the same profile as the user who initiated the resync are granted field-level access to the external objects. When you resync an external object:

- The **Display URL Reference Field** is set to None.
- If a custom field has the Is Name Field attribute, the attribute is removed. The External ID standard field is used as the name field of the external object.
- The Deployment Status doesn't change.

To understand how syncing affects relationship fields on external objects, see [Relationships on External Objects](#).

See Also

[Salesforce Connect](#)

[Salesforce Platform Features Supported by Salesforce Connect](#)

[Deployment Status for Custom Objects and External Objects](#)

[Validate and Sync an External Data Source](#)

External Objects in Salesforce Connect

External objects behave similar to custom objects except that they map to data stored outside Salesforce in an external data source. Each external object maps to a data table, and the object fields map to accessible table columns.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

As with custom objects, you must create custom tabs to display external object data in Salesforce. When you add a custom tab to an app in Salesforce Classic, it appears as a tab. When you add a custom tab to an app in Lightning Experience, it appears as an item in the app's navigation bar and in the App Launcher. See [Create a Custom Object Tab](#).

To know the field types that you can create for external objects, see [Custom Field Types](#). Some special behavior for text and number fields on external objects.

- Text fields: Make sure that the field length matches the length of the associated field on the external object to avoid any display or edit issues.
- Number fields: Ensure that the specified length can contain all digits to the left of the decimal point in external values. If the numeric value from an external system doesn't fit within the length of the associated number field on the external object, the value is blank in your Salesforce org. If you notice blank numeric field values, adjust **Length** and **Decimal Places** for the number field on the external object to accommodate more digits to the left of the decimal point. If the digits to the right of the decimal point in the external values don't fit within the specified decimal places, the value is truncated.

When a custom field on an external object record is displayed, leading and trailing spaces are removed from the field value. Keep this behavior in mind when you filter by a field that contains leading or trailing spaces. For example, include the leading and trailing spaces in the following SOQL query to match the values that are stored in the external system. However, the spaces aren't displayed in the query results.

```
SELECT FirstName__c FROM Buyers__x WHERE FirstName__c = ' Test '
```

 **Note** Formulas and roll-up summary fields can't reference fields on external objects.

Record IDs for Salesforce Connect External Objects

The first time a data row is retrieved from an external system, the external object record is assigned a Salesforce ID. Record IDs can change and aren't permanent identifiers. For a persistent ID, use an external ID. See [Understanding Changes to External Object ID \(x00\) Format and Recommended Best Practices](#).

 **Note** Salesforce IDs aren't assigned to external object records that are associated with high-data-volume external data sources.

Features Not Supported

- Set up data with external objects
 - Merge Fields
 - Schema Builder
 - Validation Rules
 - Custom fields

- Field types not supported
 - Auto-Number (available only with the cross-org adapter for Salesforce Connect)
 - Currency (available only with the cross-org adapter for Salesforce Connect)
 - Formula
 - Geolocation
 - Master-Detail Relationship
 - Picklist and Picklist (Multi-select) (available only with the cross-org adapter for Salesforce Connect)
 - Roll-up Summary
 - Text (Encrypted)
 - Text Area (Rich)
- Lookup filter on relationship fields
- Default field values
- Record-level security to manage data access for external objects
- Record types to customize external data
- Productivity Tools
 - Activities, Events, and Tasks
 - Attachments
 - Files (only for cross-org adapter)
 - Notes

To understand specific behavior and limitations, refer to considerations that apply to the corresponding Salesforce Connect adapter.

See Also

- [External Object Relationships](#)
- [Considerations for Object Relationships](#)
- [External Data Sources With Salesforce Connect](#)
- [Manage Custom Objects](#)

Writable External Objects in Salesforce Connect

Select **Writable External Objects** when you define an external data source and use Salesforce Connect external objects to create, update, and delete data. External objects are read only by default.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

 **Note** For external data sources of Amazon Athena type, writable external objects aren't supported.

External systems execute write operations initiated from Salesforce and also handle write conflicts, if any.

- It can take some time for changes to external object records to take effect. If you don't see recent changes when you view or query an external object record, try again later.
- It can't be guaranteed that all write operations that are initiated from Salesforce are applied in case of write conflicts.

If Salesforce attempts to save changes to an external object and a standard or custom object in the same transaction, an error occurs. As a best practice, we recommend you use a separate transaction to access or modify data in an external system.

Write operations on external object records initiated from different contexts can occur in varying order. When you create, update, or delete external object records via the user interface, processes or flows, operations occur synchronously. When Apex is used to perform external records operations, operations occur asynchronously and an active background queue minimizes potential save conflicts. To monitor an asynchronous job progress, use [BackgroundOperation](#).

 **Note** When a custom field on an external object record is edited, leading and trailing spaces are removed from the field value.

See Also

[Salesforce Connect](#)

[Salesforce Platform Features Supported by Salesforce Connect](#)

[External Data Sources With Salesforce Connect](#)

[Apex Developer Guide: Writable External Objects](#)

Salesforce Platform Features Supported by Salesforce Connect

Salesforce Connect provides seamless integration with the Salesforce Platform and enables users to view, search, and modify external object data.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Review the considerations for each Salesforce Connect adapter that you use to know more about Salesforce Platform feature support.

[Salesforce Connect Support for Reports](#)

Depending on network latency and the availability of the external system, reports that include an external object can take a long time to run.

Salesforce Connect Support for Record Feeds

View the Chatter feed associated with external object records you follow to see updates about the record. Following records helps keep you up to date on important changes to the external objects.

Salesforce Connect Support for Quick Actions

External objects support quick actions, except when the actions involve features or functionality that are incompatible with external objects.

Salesforce Connect Support for Flows and Processes

You can build flows and processes that include external objects and automate your organization's repetitive business tasks.

Salesforce Connect Support for the Salesforce App

You can view and search external objects from the Salesforce mobile app, Salesforce on the go!

Salesforce Connect Support for the Salesforce Console

You can access external objects from the Salesforce console only in Salesforce Classic. Other consoles, such as the Salesforce console in Lightning Experience, aren't supported.

More Features Supported by Salesforce Connect

External objects are available to Salesforce APIs, SOQL queries, SOSL and Salesforce searches, packages, Metadata API, change sets, and Lightning Experience app.

See Also

[Salesforce Connect](#)

[External Data Sources With Salesforce Connect](#)

Salesforce Connect Support for Reports

Depending on network latency and the availability of the external system, reports that include an external object can take a long time to run.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

When you run a report that includes external objects, your org performs a request callout for each external object in the report. And if it's a joined report, your org performs separate request callouts for each block. If the URL of a report callout approaches or exceeds 2 KB, the request is split into multiple HTTP calls, with each URL being less than 2 KB.

A report that includes an external object fetches up to 20,000 records for the primary object. If the report is customized to include child objects, the total number of rows can be greater or less than 10,000, depending on how many child records are fetched. To obtain more relevant external object rows,

try customizing the report.

For custom reports that include external objects as the primary object:

- If the deployment status of the external object changes, the custom report type's Deployment Status changes similarly from *Deployed* to *In Development*. See [Deployment Status for Custom Objects and External Objects](#).
- If the external object is deleted, the custom report type and reports created from it are deleted.

For large external data sources, report callouts typically access only a subset of the external data. If the report includes summary fields and formulas, those aggregate values likely reflect only a subset of your data. To improve the accuracy of the aggregate values and obtain more relevant data, try customizing the report.

As is true for all callouts for external objects, report callouts are limited by the Salesforce Connect adapters in use.

- Cross-org adapter: No callout limits. However, each callout counts toward the API usage limits of the provider org. See [API Request Limits and Allocations](#).
- OData 2.0 and OData 4.0 adapter
 - 20,000 OData callouts per hour for Enterprise, Performance, and Unlimited editions. If you require higher limits, create a support case.
 - 1,000 OData callouts per hour for Developer Edition.
- OData 4.01 adapter: No callout limits.
- Custom adapter: See [Callout Limits and Limitations](#) and [Execution Governors and Limits](#) in the *Apex Developer Guide*.

Features Not Supported

- Converted currency fields
- Cross filters
- Buckets and bucket fields
- Historical trend reporting
- Row-level formulas

See Also

[Reports and Dashboards Limits, Limitations, and Allocations](#)

[Troubleshoot Reports](#)

[External Object Relationships](#)

[Salesforce Platform Features Supported by Salesforce Connect](#)

Salesforce Connect Support for Record Feeds

View the Chatter feed associated with external object records you follow to see updates about the record. Following records helps keep you up to date on important changes to the external objects.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Features Not Supported

- Field history tracking.
- External objects that map to high-data-volume external data sources.

See Also

[Records and List Views](#)

[Salesforce Connect](#)

[Salesforce Platform Features Supported by Salesforce Connect](#)

Salesforce Connect Support for Quick Actions

External objects support quick actions, except when the actions involve features or functionality that are incompatible with external objects.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

With custom quick actions, you can make your users' navigation and workflow as smooth as possible by giving them convenient access to information that's most important. For example, you can let users quickly create or update records, send emails, and more in the context of the external object.

Features Not Supported

- Log a Call action: Creating tasks isn't available for external objects.
- Predefined Field Values for Quick Action Fields: Formulas can't reference external object fields.
- Lightning Web Component actions don't work on external objects because those objects are built on the Aura framework.

See Also

- [Action Types](#)
- [Salesforce Connect](#)
- [Salesforce Platform Features Supported by Salesforce Connect](#)

Salesforce Connect Support for Flows and Processes

You can build flows and processes that include external objects and automate your organization's repetitive business tasks.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

If a flow or a process accesses standard or custom objects along with external objects, you must commit the changes to the standard or custom object before accessing the external objects. We recommend using a separate transaction to access data from the external system.

- In Flow Builder, add a screen, local action, or Wait element that pauses until a flow-based time occurs. See [Flow Elements](#).
- In Process Builder, add a scheduled action. See [Add Actions to Your Process](#).

To understand how processes interact with external objects, see [Compatibility Considerations for Processes](#).

To know the limits while building flows with external objects, see [External Object Considerations for Flows](#).

Features Not Supported

These automation tool features aren't available for external objects in Salesforce Connect.

- Approval Processes
- Workflow Rules

See Also

- [Flow Builder Tour](#)
- [Process Builder](#)
- [Salesforce Connect](#)
- [Salesforce Platform Features Supported by Salesforce Connect](#)

Salesforce Connect Support for the Salesforce App

You can view and search external objects from the Salesforce mobile app, Salesforce on the go!

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

As with custom objects, external objects must be assigned to tabs that users can access, and object permissions must be granted via profiles or permission sets. When you search for an external object, click **More** in the Recent section to view the search results.

When external objects are used with Salesforce for iOS and Salesforce mobile web on iOS devices, you must select **Enable Search** in the associated external data sources. Only then external objects appear in these apps. This requirement doesn't apply to custom adapters for Salesforce Connect.

See Also

[Salesforce Mobile App](#)

[Salesforce Connect](#)

[Salesforce Platform Features Supported by Salesforce Connect](#)

Salesforce Connect Support for the Salesforce Console

You can access external objects from the Salesforce console only in Salesforce Classic. Other consoles, such as the Salesforce console in Lightning Experience, aren't supported.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

External objects haven't been fully adapted to a console and can cause unexpected behaviors. Unlike other objects that haven't been fully adapted to a console, external objects aren't marked with asterisks in the console setup area.

See Also

[Salesforce Console in Salesforce Classic](#)

[Salesforce Classic Console Limitations](#)

[Salesforce Connect](#)

[Salesforce Platform Features Supported by Salesforce Connect](#)

More Features Supported by Salesforce Connect

External objects are available to Salesforce APIs, SOQL queries, SOSL and Salesforce searches, packages, Metadata API, change sets, and Lightning Experience app.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

API Query

- To know how Salesforce Connect accesses the external data in real time via Web service callouts, see `query()`, `queryAll()`, and `queryMore()`.
- To understand the limitations that apply to `queryAll()` and `queryMore()` calls on external data, see [External Objects](#) in the SOAP API Developer Guide.
- To implement server-driven or client-driven paging for external object data that's obtained by a custom adapter, see [Paging with the Apex Connector Framework](#).

SOQL

- To know about the specific limits SOQL applies to external objects, see [External objects in SOQL Limits on Objects](#).
- To understand the limitations for external objects when you design SOQL relationship queries, see [Understanding Relationship Query Limitations](#).

SOSL and Salesforce Searches

- To know about the specific limits SOSL applies to external objects in search results, see [SOSL Limits on External Object Search Results](#).
- To learn about how to add `RETURNING` clause to a SOSL query with external objects, see [RETURNING FieldSpec](#).
- To understand how search works with external objects, look for external objects in [Searchable Fields by Object in Lightning Experience](#) and [Searchable Fields by Object in Salesforce Classic](#).

- (Salesforce Classic only) Search results for external object display only the top 25 rows.

Packaging

- To understand how packaging affects external data sources and external objects, see [Special Behavior of Components in Packages](#).
- To know what components are automatically included in the package when you add external data sources and external objects, see [Components Automatically Added to Packages](#).
- To learn about the behavior of external data sources and external objects with permission sets or profile settings, see [About Permission Sets and Profile Settings](#).

Deployment via the Metadata API

In Metadata API, external objects are represented as [CustomObject](#).

Change Sets

External objects are included in the Custom Object component.

Lightning Experience App

When you access external objects from the Lightning Experience app, the associated external data sources must have the High Data Volume option deselected. This requirement doesn't apply to the cross-org adapter for Salesforce Connect.

See Also

[Salesforce Connect](#)

[Salesforce Platform Features Supported by Salesforce Connect](#)

[Writable External Objects in Salesforce Connect](#)

Access Data in Another Salesforce Org with the Cross-Org Adapter for Salesforce Connect

Connect your users to data that's stored in another Salesforce org.

[Cross-Org Adapter for Salesforce Connect](#)

Collaborate more effectively and improve processes by connecting the data across your Salesforce orgs. With the cross-org adapter, Salesforce Connect uses Lightning Platform REST API calls to access records in other Salesforce orgs. Nevertheless, setup is quick and easy with point-and-click tools.

[Set Up Salesforce Connect to Access Data in Another Org with the Cross-Org Adapter](#)

Provide users with seamless access to data in your other Salesforce orgs so that they have a complete view of the business. Setting up the cross-org adapter for Salesforce Connect is quick and easy with point-and-click tools.

Considerations for Salesforce Connect–Cross-Org Adapter

Understand the special behaviors, limits, and recommendations for using the cross-org adapter for Salesforce Connect.

Cross-Org Adapter for Salesforce Connect

Collaborate more effectively and improve processes by connecting the data across your Salesforce orgs. With the cross-org adapter, Salesforce Connect uses Lightning Platform REST API calls to access records in other Salesforce orgs. Nevertheless, setup is quick and easy with point-and-click tools.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Your users and the Lightning Platform interact with other orgs' data via external objects. The cross-org adapter for Salesforce Connect converts each of those interactions into a Lightning Platform REST API call.

Suppose that you store your inventory of products in one Salesforce org. You want your regional and local branch offices, who have their own orgs, to see the latest information about your stock. With the cross-org adapter for Salesforce Connect, those other organizations can easily access your data while respecting access restrictions that you control.

The cross-org adapter makes a Lightning Platform REST API call each time that:

- A user clicks an external object tab for a list view.
- A user views a record detail page of an external object.
- A user views a record detail page of a parent object that displays a related list of child external object records.
- A user performs a Salesforce global search.
- A user creates, edits, or deletes an external object record.
- A user runs a report.
- The preview loads in the report builder.
- An external object is accessed via flows, processes, APIs, Apex, SOQL, or SOSL.
- You validate or sync an external data source.

To set up Salesforce Connect with the cross-org adapter, you use only point-and-click tools.

Subscriber and Provider Orgs in Salesforce Connect–Cross-Org Adapter

The *provider org* stores the data that the *subscriber org* accesses.

API Names for External Objects and Custom Fields in Salesforce Connect–Cross-Org Adapter

If external objects and custom fields are created in the subscriber org via syncing, their API names are derived from the corresponding API names in the provider org.

[Record IDs and External IDs for External Objects in Salesforce Connect–Cross-Org Adapter](#)

External object record IDs are derived from the corresponding record IDs in the provider organization. External ID values in external object records match the record IDs in the provider organization.

[User Access to External Data in Salesforce Connect–Cross-Org Adapter](#)

A user's access to external data is determined by settings on both subscriber and provider orgs.

See Also

[Salesforce Connect](#)

[Set Up Salesforce Connect to Access Data in Another Org with the Cross-Org Adapter](#)

[Considerations for Salesforce Connect–Cross-Org Adapter](#)

Subscriber and Provider Orgs in Salesforce Connect–Cross-Org Adapter

The *provider org* stores the data that the *subscriber org* accesses.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

You define the external data source and external objects in the subscriber org. Manually create the external objects and their fields, or automatically create them by syncing the provider org's metadata. When users view or search those external objects in the subscriber org, the data is obtained from the provider org and displayed in the subscriber org. When users create or edit external object records in the subscriber org, the data is saved in the provider org.

- An org can serve as both a subscriber and a provider.
- A subscriber org can access data from multiple provider orgs.
- A provider org can let multiple subscriber orgs access its data.

See Also

[Cross-Org Adapter for Salesforce Connect](#)

API Names for External Objects and Custom Fields in Salesforce Connect–Cross-Org Adapter

If external objects and custom fields are created in the subscriber org via syncing, their API names are derived from the corresponding API names in the provider org.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Each external object's API name ends with `__x`. Custom fields on external objects use the traditional `__c` suffix in the API name. Specifically for objects and custom fields that are synced with the cross-org adapter for Salesforce Connect:

- For an API name with no suffix in the provider org, the API name is reused in the subscriber org, but with an applied `__x` suffix for an object or `__c` suffix for a field.
- For an API name with a suffix in the provider org, the API name is reused in the subscriber org. But one of the underscores (`_`) from the original suffix is removed, and a new `__x` or `__c` suffix is applied.

 **Example** If you sync the provider org's Account object, the subscriber org creates:

- An external object with the API name `Account__x`
- Custom fields including one with the API name `Account__x.Name__c`

If you sync the provider org's `CustObj__c` object, the subscriber org creates:

- An external object with the API name `CustObj_c__x`
- Custom fields including one with the API name `CustObj_c__x.Name__c`

If the provider org's object has a custom field, the subscriber org creates the custom field on the equivalent external object, for example:

- `Account__x.MyCustField_c__c`
- `CustObj_c__x.MyOtherCustField_c__c`

If you sync the provider org's `Account__x` external object, the subscriber org creates:

- An external object with the API name `Account_x__x`
- Custom fields including one with API name `Account_x__x.Name_c__c`

See Also

[Cross-Org Adapter for Salesforce Connect](#)

Record IDs and External IDs for External Objects in Salesforce Connect—Cross-Org Adapter

External object record IDs are derived from the corresponding record IDs in the provider organization. External ID values in external object records match the record IDs in the provider organization.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Each object in Salesforce has an object ID with a key prefix as the first three characters. When an external object is created, it's assigned a unique key prefix.

Each external object record has a record ID that uses the same key prefix as the external object ID. The rest of the external object record ID matches the original record ID that's in the provider organization, excluding its original key prefix.

Each record ID that comes from the provider organization becomes a case-insensitive 18-character alphanumeric string in the subscriber organization.

The original record ID is available in the subscriber organization as the value of the External ID standard field on the external object record.

Each external object has an **External ID** standard field. Its values uniquely identify each external object record in your org. When the external object is the parent in an external lookup relationship, the External ID standard field is used to identify the child records.

 **Example** You sync the provider organization's Account object, and the subscriber organization's `Account__x` object is assigned the key prefix `x00`. An account in the provider organization with the ID 001B0000003SVC7IAO appears in the subscriber organization with the ID `x00B0000003SVC7IAO` and the external ID `001B0000003SVC7IAO`.

See Also

[Cross-Org Adapter for Salesforce Connect](#)

User Access to External Data in Salesforce Connect—Cross-Org Adapter

A user's access to external data is determined by settings on both subscriber and provider orgs.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

The credentials that are used by the subscriber org to connect to the provider org are associated with a user in the provider org. We refer to this user as the *connected user*.

A user in the subscriber org can access only data that the connected user can access within the provider org. In other words, the subscriber org's user access respects the connected user's access restrictions, which are determined by these settings in the provider org.

- Object-level security—permission sets and profiles
- Field-level security—permission sets and profiles
- Record-level security—organization-wide sharing settings, role hierarchies, and sharing rules

In the subscriber org, grant users access to external objects via permission sets and profiles.

See Also

[Cross-Org Adapter for Salesforce Connect](#)

[Set Up Salesforce Connect to Access Data in Another Org with the Cross-Org Adapter](#)

Set Up Salesforce Connect to Access Data in Another Org with the Cross-Org Adapter

Provide users with seamless access to data in your other Salesforce orgs so that they have a complete view of the business. Setting up the cross-org adapter for Salesforce Connect is quick and easy with point-and-click tools.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit external data sources:	Customize Application
To create and edit external objects:	Customize Application
To define or change object-level help:	Customize Application
To create and edit custom fields:	Customize Application
To edit permission sets and user profiles:	Manage Profiles and Permission Sets
To edit another user's authentication settings for external systems:	Manage Users

Setting up Salesforce Connect with the cross-org adapter involves these high-level steps.

1. Define an external data source of type **Salesforce Connect: Cross-Org**.

Create an external data source for each provider org.

2. [Create the external objects.](#)

Perform this task only if you don't sync to automatically create the external objects. In the subscriber org, create an external object for each object in the provider org that you want to access.

3. [Create help content for the external objects.](#)

Help your users distinguish between external objects and the other objects in the subscriber org, which can have similar names and types of data. On the subscriber org, create Visualforce pages to describe the external objects. When your users click **Help for this Page** on an external object, they read your custom help content.

4. [Add custom fields and relationships to the external objects.](#)

Create relationships between objects. If you didn't sync to automatically create the external objects and their fields on the subscriber org, create a custom field for each of the provider org's fields that you want to access.

5. [Enable user access to external objects.](#)

Grant object permissions through permission sets or profiles.

6. [Enable user access to the fields on the external objects.](#)

Grant field permissions through permission sets or profiles.

7. If the external data source uses per-user authentication:

a. [Let users authenticate to the external system.](#)

Grant users access to authentication settings for the external data source through permission sets or profiles.

b. [Set up each user's authentication settings.](#)

You or your users can perform this task.

c. [Configure OAuth settings for the connected app.](#)

If the Require Proof Key for Code Exchange (PKCE) Extension for Supported Authorization

Flows setting is visible and selected in the OAuth connected app's API settings, be sure to deselect it.



Tip Train your users on how to set up their authentication settings for external systems. Make sure that they know which credentials to enter for the provider org. If you're using OAuth 2.0, the OAuth flow displays the Salesforce login page twice: first to log in to the provider org to obtain an access token, and then to log back in to the subscriber org. Test the OAuth flow for potentially confusing prompts or redirects, and train your users as needed. OAuth flows vary, depending on your external system, authentication provider, and specified scopes.

See Also

[Cross-Org Adapter for Salesforce Connect](#)

[Considerations for Salesforce Connect–Cross-Org Adapter](#)

[Developer Guide: *Visualforce Developer Guide*](#)

[External Object Relationships](#)

[Subscriber and Provider Orgs in Salesforce Connect–Cross-Org Adapter](#)

Define an External Data Source for Salesforce Connect Cross-Org Adapter

Give your users seamless access to data across your Salesforce orgs.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit external data sources: Customize Application

1. From Setup, enter *External Data Sources* in the Quick Find box, then select **External Data Sources**.
2. Click **New External Data Source**, or click **Edit** to modify an existing external data source.
3. Complete the fields.

Field	Description
Label	A user-friendly name for the external data source. The label appears in the Salesforce user interface, such as in list views. If you set Identity Type to Per User , this label appears when your users view or edit their authentication settings for external systems.
Name	A unique identifier that's used to refer to this external data source definition through the API. The name can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
Type	Select Salesforce Connect: Cross-Org .
Connect to	Determines which URL is used to connect to the provider org.
URL	If you selected Connect to Custom URL , enter the login URL for the provider org.
API Version	Select an API version that the provider org supports. The API version determines which of the provider org's objects, fields, and types you can access from the subscriber org.
Connection Timeout	Number of seconds to wait for a response from the provider org before timing out. By default, the value is set to the maximum of 120 seconds.
Writable External	The Lightning Platform and users in this org can create, update, and delete

Field	Description
Objects	records for external objects associated with the external data source. The external object data is stored outside the org. By default, external objects are read only.
Enable Search	<p>Determines whether global searches in the subscriber org also search the external objects' data, which is stored in the provider org.</p> <p>When selected, you can control which external objects are searchable by selecting or deselecting Allow Search on each external object.</p> <p>Only text, text area, and long text area fields on external objects can be searched. If an external object has no searchable fields, searches on that object return no records.</p>
Identity Type	Determines whether the subscriber org uses one set or multiple sets of credentials to access the provider org. See Identity Type for External Data Sources .

4. Select the authentication protocol.

- If you select **Password Authentication**, enter the username and password to access the external system.
- If you select **OAuth 2.0**, complete these fields.

Field	Description
Authentication Provider	Select a Salesforce authentication provider. See Configure a Salesforce Authentication Provider . On a production org, an external data source can't use an authentication provider that directs authorization or token requests to a sandbox org. Similarly, on a sandbox org, an external data source can't use an authentication provider that directs authorization or token requests to a production org.
Scope	<p>Specifies the scope of permissions to request for the access token. Your authentication provider determines the allowed values. See Use the Scope Parameter.</p> <p>Keep these considerations in mind when you set a scope.</p> <ul style="list-style-type: none"> • The value that you enter replaces the Default Scopes value that's defined in the specified authentication provider. • Whether scopes are defined can affect whether each OAuth flow prompts the user with a consent screen. • We recommend that you request a refresh token or offline access. Otherwise, when the token expires, you lose access to the external system.

Field	Description
Start Authentication Flow on Save	<p>To authenticate to the external system and obtain an OAuth token, select this checkbox. This authentication process is called an OAuth flow.</p> <p>When you click Save, the external system prompts you to log in. After successful login, the external system grants you an OAuth token for accessing its data from this org.</p> <p>Redo the OAuth flow when you need a new token—for example, if the token expires—or if you edit the Scope or Authentication Provider fields. When the token expires, the external system returns a 401 HTTP error status.</p>

5. Click **Save**.
6. Click **Validate and Sync**, and confirm that the connection is successful.
 - If you receive an error message, refer to these resources: “Status Codes and Error Responses” in the *REST API Developer Guide*, and the “API Fault Element,” “ExceptionCode,” “Error,” and “StatusCode” sections of “Core Data Types Used in API Calls” in the *SOAP API Developer Guide*.
 - Starting in Winter ’26 (API version 65.0), the SOAP API login() method is disabled by default in new orgs. If you encounter an INVALID_OPERATION error while using password authentication, use API version 64.0 or earlier and manually enable the SOAP API login() in Setup to resolve it.
7. Optionally, select tables and click **Sync** to do the following for each selected table.
 - Automatically create a Salesforce external object.
 - Automatically create a custom field for each table column that’s compatible with a Salesforce metadata field type.

 **Note** Before you sync, make sure that you understand the considerations that are described in these topics.

- [Sync an External Data Source for Salesforce Connect](#)
- [Sync Considerations for Salesforce Connect–Cross-Org Adapter](#)

You can instead choose to manually create the external objects and custom fields that map to the external data. Doing so lets you customize the external object names, decide which table columns to create custom fields for, and customize the custom field names. However, this approach takes longer and requires manual maintenance.

See Also

[Set Up Salesforce Connect to Access Data in Another Org with the Cross-Org Adapter](#)

[Store Authentication Settings for External Systems](#)

[API Names for External Objects and Custom Fields in Salesforce Connect–Cross-Org Adapter](#)

[Subscriber and Provider Orgs in Salesforce Connect–Cross-Org Adapter](#)

[Developer Guide: REST API Developer Guide](#)

Considerations for Salesforce Connect–Cross-Org Adapter

Understand the special behaviors, limits, and recommendations for using the cross-org adapter for Salesforce Connect.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Also review the considerations that apply to all Salesforce Connect adapters.

[Salesforce Compatibility Considerations for Salesforce Connect–Cross-Org Adapter](#)

Some Salesforce features and functionality have special behaviors or aren't available for external objects that are associated with an external data source of type `Salesforce Connect: Cross-Org`.

[Sync Considerations for Salesforce Connect–Cross-Org Adapter](#)

When you validate and sync an external data source of type `Salesforce Connect: Cross-Org`, some special behaviors and limitations apply.

[Writable External Objects Considerations for Salesforce Connect–Cross-Org Adapter](#)

Some behaviors and limitations affect writable external objects that are associated with the cross-org adapter for Salesforce Connect.

[API Usage Considerations for Salesforce Connect–Cross-Org Adapter](#)

With the cross-org adapter, Salesforce Connect uses Lightning Platform REST API calls to access records in other Salesforce orgs. Depending on how the external object is accessed, each call counts toward the API usage limits of only the provider org or of both provider and subscriber orgs.

[Picklist Considerations for Salesforce Connect–Cross-Org Adapter](#)

Special behaviors and limitations apply to picklist and multi-select picklist fields on external objects.

[Currency Considerations for Salesforce Connect–Cross-Org Adapter](#)

Special behaviors and limitations apply to currency fields on external objects.

See Also

[Salesforce Platform Features Supported by Salesforce Connect](#)

[Salesforce Compatibility Considerations for Salesforce Connect–Cross-Org Adapter](#)

Some Salesforce features and functionality have special behaviors or aren't available for external objects that are associated with an external data source of type `Salesforce Connect: Cross-Org`.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- The cross-org adapter for Salesforce Connect can access only queryable objects in the provider org. If you define an external object whose table name specifies an object that can't be queried, your users and the Lightning Platform can't access that external object.
- You can't use Salesforce Connect external objects to access big objects in another org.
- When you deploy an external data source that uses OAuth 2.0 from a sandbox org to a production org, you must update the authentication provider. On a production org, an external data source can't use an authentication provider that directs authorization or token requests to a sandbox org. Similarly, on a sandbox org, an external data source can't use an authentication provider that directs authorization or token requests to a production org.

Also review the considerations that apply to all Salesforce Connect adapters.

Sync Considerations for Salesforce Connect–Cross-Org Adapter

When you validate and sync an external data source of type [Salesforce Connect: Cross-Org](#), some special behaviors and limitations apply.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- Which objects and fields can be synced is determined by the object permissions and field permissions of the provider org's user whose credentials are defined in the external data source definition. See [User Access to External Data in Salesforce Connect–Cross-Org Adapter](#).
- Only queryable objects can be synced.
- You can't sync an object whose API name contains 38 or more characters. An external object's name can't exceed 40 characters, including the automatically appended `_x` suffix. See [API Names for External Objects and Custom Fields in Salesforce Connect–Cross-Org Adapter](#).
- These field types aren't synced.
 - Encrypted text
 - Rich text area
- Global picklist value sets aren't synced. If a provider org's picklist field uses a global picklist value set,

- syncing creates a local picklist field on the subscriber org. A local picklist field has its own set of values.
- Inactive picklist values aren't synced. If the subscriber org accesses an external object record that contains an inactive picklist value, the inactive value is added to the picklist field on the external object.
 - Syncing converts restricted picklists on the provider org into unrestricted picklists on the subscriber org's external objects.

We recommend having only unrestricted picklists on external objects, even when they're associated with restricted picklists on the provider org. Restricted picklists on the provider org block unapproved values from the subscriber org, eliminating the need to restrict picklists on external objects.

- To enable users to change the currency when editing an external object record, add the Currency field to the page layouts. All other synced fields are automatically added to the default page layout.
- Syncing always enables search on the external object when search is enabled on the external data source, and vice versa.
- Hierarchical, lookup, and master-detail relationship fields are synced. However, they become text fields in the subscriber org, and their values appear as IDs, not as record names.

For example, suppose that we sync the Account object. When we view the Acme Wireless account in the provider org, the value of the Account Owner (`Account.OwnerId`) field appears as John Smith.

When we view the equivalent account in the subscriber org, the value for the (`Account_x.OwnerId_c`) field appears as 005B00000019eapIAA.

- You can use external lookup relationships in the subscriber org to mirror lookup relationships in the provider org. For each lookup relationship that you want to bring into the subscriber org, sync the parent and child objects. Each lookup relationship field becomes a text field on the subscriber org. Change the field type of the sync-created text field to External Lookup Relationship. When specifying the parent of the external lookup relationship, select the external object that corresponds to the parent object in the provider org.
- The names and labels of synced fields on the subscriber org are derived from the API names—not the labels—of the fields on the provider org. For example, the Account object in the provider org has the Account Owner (`OwnerId`) standard field. If we sync the Account object, the `Account_x.OwnerId_c` field in the subscriber org has the label “Owner ID” and the name “OwnerId.”

! **Important** When you select the provider org objects to sync, determine whether check marks appear in the Synced column. If a Synced check mark appears, the subscriber org has an external object whose object name (for example, `Account_x`) associates it with the object in the provider org (for example, `Account`). If you select the object and click **Sync**:

- The external object is overwritten.
- Any custom field on the external object is overwritten if its API name (for example, `Email_c_c`) associates it with a field on the provider org (for example, `Email_c`).
- Any other custom fields on the external object remain as they are, including:
 - Previously synced custom fields whose API names were changed by editing their **Field Name** values.
 - Manually added custom fields whose API names aren't associated with fields on the provider org's object.

If no Synced check mark appears, and you sync the object, a new external object is created in the subscriber org. For example, the object name is changed on the provider org to no longer be associated with the object name of the external object on the subscriber org. Syncing that object creates a new external object on the subscriber org. We recommend that you change the object name of the existing external object to match the updated object name on the provider org before you sync.

Also review the considerations that apply to all Salesforce Connect adapters.

See Also

- [API Names for External Objects and Custom Fields in Salesforce Connect–Cross-Org Adapter](#)
- [Subscriber and Provider Orgs in Salesforce Connect–Cross-Org Adapter](#)
- [Sync an External Data Source for Salesforce Connect](#)
- [Cross-Org Adapter for Salesforce Connect](#)

Writable External Objects Considerations for Salesforce Connect–Cross-Org Adapter

Some behaviors and limitations affect writable external objects that are associated with the cross-org adapter for Salesforce Connect.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- Some fields remain read only on writable external objects, such as fields that are creatable but not editable and fields that have derived or calculated values. For example:
 - Address fields such as **Billing Address** and **Shipping Address**, which are derived from writable fields, such as **Street Address**, **State**, and **Zip**)
 - Auto-number fields
 - **Created by** fields
 - Formula fields
 - **Last Modified** fields
 - Roll-up summaries
- Picklist values on external objects can get out of sync with picklist values on the provider org. If picklist values are changed on the provider org, resync or manually delete and recreate the associated picklist fields on the subscriber org.

Also review the considerations that apply to all Salesforce Connect adapters.

See Also

- [Writable External Objects in Salesforce Connect](#)
- [Sync an External Data Source for Salesforce Connect](#)
- [Cross-Org Adapter for Salesforce Connect](#)

API Usage Considerations for Salesforce Connect–Cross-Org Adapter

With the cross-org adapter, Salesforce Connect uses Lightning Platform REST API calls to access records in other Salesforce orgs. Depending on how the external object is accessed, each call counts toward the API usage limits of only the provider org or of both provider and subscriber orgs.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

When a user accesses an external object in one of the following ways, the Lightning Platform REST API call counts toward the API usage limits of the provider org.

- Opening a list view of external object records
- Viewing an external object record detail page
- Viewing a parent object record that contains an external object related list
- Viewing a child object record that contains an external lookup field
- Executing a search that also searches external objects
- Accessing an external object from a flow, Visualforce page, Apex class, or Apex trigger
- Creating, editing, or deleting an external object record
- Running a report
- Editing a report and causing the preview to load in the report builder

If a user or system accesses an external object through the SOAP API, Bulk API, or Lightning Platform REST API, that access counts toward the API usage limits of both the subscriber org and the provider org.

See Also

- [Salesforce Limits Quick Reference Guide: API Request Limits and Allocations](#)
- [Considerations for Salesforce Connect–Cross-Org Adapter](#)
- [Subscriber and Provider Orgs in Salesforce Connect–Cross-Org Adapter](#)

Picklist Considerations for Salesforce Connect–Cross-Org Adapter

Special behaviors and limitations apply to picklist and multi-select picklist fields on external objects.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- You can edit picklist values on external objects, but changes to fields on the provider org aren't automatically reflected in the subscriber org. To reflect changes on the subscriber org, resync the external object or manually update the picklist values on the external object.
If you don't resync or update the picklist values on the external object:
 - When an active picklist value is added to the provider org, the subscriber org doesn't display it as an available picklist value on external object records.
 - When an active picklist value is deleted from or made inactive on a restricted picklist on the provider org, the subscriber org can't create or edit external object records with that value.
- Global picklist value sets aren't synced. If a provider org's picklist field uses a global picklist value set, syncing creates a local picklist field on the subscriber org. A local picklist field has its own set of values.
- Inactive picklist values aren't synced. If the subscriber org accesses an external object record that contains an inactive picklist value, the inactive value is added to the picklist field on the external object.
- Syncing converts restricted picklists on the provider org into unrestricted picklists on the subscriber org's external objects.
- We recommend having only unrestricted picklists on external objects, even when they're associated with restricted picklists on the provider org. Restricted picklists on the provider org block unapproved values from the subscriber org, eliminating the need to restrict picklists on external objects.

See Also

- [Sync Considerations for Salesforce Connect–Cross-Org Adapter](#)
- [Subscriber and Provider Orgs in Salesforce Connect–Cross-Org Adapter](#)
- [Considerations for Salesforce Connect–Cross-Org Adapter](#)
- [Picklist Considerations for Salesforce Connect OData Adapters](#)

Currency Considerations for Salesforce Connect–Cross-Org Adapter

Special behaviors and limitations apply to currency fields on external objects.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- Enable multiple currencies in the subscriber org, and make sure that the subscriber org has all the currencies that the provider orgs use. If a currency is later added to the provider org, add the currency to the subscriber org.^{[1][2]}
If you can't enable multiple currencies in the subscriber org, make sure that all provider orgs are single-currency and use the same currency as the subscriber org.
- To enable users to change the currency when editing an external object record, add the **Currency** field to the page layouts.
- If a user tries to change the currency in an external object record, the list of currency options isn't limited to active currencies in the provider org. If the user selects a currency that's inactive or unavailable in the provider org, the user gets an error and can't save the record.
- The `convertCurrency()` function is ignored in SOQL queries of external objects.

See Also

- [Considerations for Enabling Multiple Currencies](#)
- [Considerations for Salesforce Connect–Cross-Org Adapter](#)
- [Subscriber and Provider Orgs in Salesforce Connect–Cross-Org Adapter](#)

Access External Data with OData Adapters for Salesforce Connect

Connect your users to data that's exposed via the Open Data Protocol.

[OData Adapters for Salesforce Connect](#)

Connect to your back office for a complete view of your business. With the OData 2.0, 4.0, or 4.01 adapter, Salesforce Connect uses Open Data Protocol Version 2.0, Version 4.0, or Version 4.01 to access data that's stored outside Salesforce.

[Set Up Salesforce Connect to Access External Data with OData Adapters](#)

Let users view and search data that's stored outside your Salesforce org, such as data in an enterprise resource planning (ERP) system.

[Considerations for Salesforce Connect OData Adapters](#)

Understand the special behaviors, limits, and recommendations for using the OData 2.0, 4.0, and 4.01 adapters for Salesforce Connect.

[Picklist Considerations for Salesforce Connect OData Adapters](#)

Special behaviors and limitations apply to picklist fields on external objects for OData 2.0, 4.0, and 4.01 adapters.

[OData Reference for Salesforce Connect OData Adapters](#)

Get to know the Salesforce implementation of the OData protocol for accessing external systems with Salesforce Connect.

[Use External Change Data Capture to Track Data Changes on External Objects](#)

With External Change Data Capture, you can track changes to data that is stored outside your Salesforce org when using the OData 4.0 and 4.01 adapters. You can then build automation in response to the changes to increase productivity or provide a better customer experience.

OData Adapters for Salesforce Connect

Connect to your back office for a complete view of your business. With the OData 2.0, 4.0, or 4.01 adapter, Salesforce Connect uses Open Data Protocol Version 2.0, Version 4.0, or Version 4.01 to access data that's stored outside Salesforce.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Your users and the Lightning Platform interact with the external data via external objects. Salesforce Connect converts each of those interactions into an OData query that contains the relevant parameters to filter the results. Salesforce performs an OData callout each time that:

- A user clicks an external object tab for a list view.
- A user views a record detail page of an external object.
- A user views a record detail page of a parent object that displays a related list of child external object records.
- A user performs a Salesforce global search.
- A user creates, edits, or deletes an external object record.
- A user runs a report.
- The preview loads in the report builder.
- An external object is accessed via flows, processes, APIs, Apex, SOQL, or SOSL.
- You validate or sync an external data source.

The OData adapters for Salesforce Connect can access external data that's exposed via services called *OData producers*. Learn more about OData producers at www.odata.org.

To understand the limitations on Apex code accessing external objects via the OData adapters, see [Apex Considerations for Salesforce Connect External Objects](#).

[External IDs and OData Entity Keys](#)

When you access external data with the OData 2.0, 4.0, or 4.01 adapter for Salesforce Connect, the values of the External ID standard field on an external object are derived according to the entity key that's defined in the OData service metadata document.

[Client-Driven and Server-Driven Paging for Salesforce Connect OData Adapters](#)

It's common for Salesforce Connect queries of external data to have a large result set that's broken into smaller batches or pages. You can decide whether to have the paging behavior controlled by the external system (server-driven) or by the OData adapter for Salesforce Connect (client-driven).

[General Limits for Salesforce Connect OData Adapters](#)

Understand the limits for the OData 2.0, 4.0, and 4.01 adapters for Salesforce Connect.

See Also

[Salesforce Connect](#)

[Set Up Salesforce Connect to Access External Data with OData Adapters](#)

[Considerations for Salesforce Connect OData Adapters](#)

[OData Reference for Salesforce Connect OData Adapters](#)

External IDs and OData Entity Keys

When you access external data with the OData 2.0, 4.0, or 4.01 adapter for Salesforce Connect, the values of the External ID standard field on an external object are derived according to the entity key that's defined in the OData service metadata document.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Each external object has an External ID standard field. Its values uniquely identify each external object record in your org. When the external object is the parent in an external lookup relationship, the External ID standard field is used to identify the child records.

 **Important** Don't use sensitive data as the values of the External ID standard field or fields designated as name fields, because Salesforce sometimes stores those values.

- External lookup relationship fields on child records store and display the External ID values of the parent records.
- For internal use only, Salesforce stores the External ID value of each row that's retrieved from the external system. This behavior doesn't apply to external objects that are associated with high-data-volume external data sources.

This list view for the Order_Detail external object displays External ID values.

External ID	Display URL
10248-11	http://service...
10250-51	http://service...
10464-56	http://service...

Each External ID value is derived according to the entity key that's defined in the OData service metadata document of the remote data service (OData producer). The entity key is formed from a subset of the entity type's properties.

This excerpt from an OData service metadata document shows that the External ID values for the Order_Detail external object are derived from the OrderID and ProductID properties.

```

<EntityType Name="Order_Detail">
  <Key>
    <PropertyRef Name="OrderID"/>
    <PropertyRef Name="ProductID"/>
  </Key>
  <Property Name="OrderID" Type="Edm.Int32" Nullable="false"/>
  <Property Name="ProductID" Type="Edm.Int32" Nullable="false"/>
  <Property Name="UnitPrice" Type="Edm.Decimal" Nullable="false" Precision="19" Scale="4"/>
  <Property Name="Quantity" Type="Edm.Int16" Nullable="false"/>
  <Property Name="Discount" Type="Edm.Single" Nullable="false"/>
  ...

```

This record detail page displays the OrderID and ProductID fields. Their values are combined to create the value of the External ID standard field.

Discount	0.15000001
OrderID	10250
ProductID	51
Quantity	35
UnitPrice	42.4000

If you enable writable external objects, determine whether the external system requires write operations to specify values for the entity keys. For example, many external systems generate values for entity keys when new external object records are created in Salesforce. If your external system requires write operations to specify values for entity keys, ensure that External ID standard field values and entity key values don't contradict each other. For each write operation, include either the External ID standard field value or the custom field values that form the entity key, but never both.

See Also

- [OData Adapters for Salesforce Connect](#)
- [Writable External Objects in Salesforce Connect](#)

Client-Driven and Server-Driven Paging for Salesforce Connect OData Adapters

It's common for Salesforce Connect queries of external data to have a large result set that's broken into smaller batches or pages. You can decide whether to have the paging behavior controlled by the external system (server-driven) or by the OData adapter for Salesforce Connect (client-driven).

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

By default, the OData adapters for Salesforce Connect use client-driven paging. Specifically, the OData requests use the `$top` and `$skip` system query options to page through the result set.

With server-driven paging, the external system determines the page sizes and batch boundaries. The

external system's paging settings can optimize the external system's performance and improve the load times for external objects in your org. Also, the external data can change while your users or the Lightning Platform are paging through the result set. Typically, server-driven paging adjusts batch boundaries to accommodate changing external data more effectively than client-driven paging.

The **Server Driven Pagination** field on the external data source specifies whether to use client-driven or server-driven paging. If you enable server-driven paging on an external data source, Salesforce ignores the requested page sizes, including the default `queryMore()` batch size of 500 rows. The pages returned by the external system determine the batches, but each page can't exceed 2,000 rows. However, the limits for the OData adapters for Salesforce Connect still apply.

See Also

[Define an External Data Source for Salesforce Connect–OData 2.0 or 4.0 Adapter](#)

[Define an External Data Source for Salesforce Connect OData 4.01 Adapter](#)

[General Limits for Salesforce Connect OData Adapters](#)

[OData Query String Options](#)

General Limits for Salesforce Connect OData Adapters

Understand the limits for the OData 2.0, 4.0, and 4.01 adapters for Salesforce Connect.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

If your users or applications encounter rate limit errors for OData callouts, try one or more of the following.

- Select **High Data Volume** in the external data source definition. Doing so bypasses most rate limits, but some special behaviors and limitations apply.
- If you have Apex code that invokes the external system, modify that code to cache frequently accessed external data that seldom changes.
- Contact Salesforce to request a higher limit.

Maximum HTTP request size for OData	8 MB
Maximum HTTP response size for OData	8 MB
Maximum result set size for an OData query	16 MB
Maximum result set size for an OData subquery	1,000 rows

For OData 2.0 and 4.0, an org is limited to:

- 100k new external object record IDs per hour for long-term ID mapping
- 100k new external object record IDs per hour for short-term ID mapping

Starting in the Spring '25 release, these limits are removed for OData 4.01. This functionality is available to orgs hosted on Hyperforce on a rolling basis.

 **Note** To view external object records in Lightning Experience, Salesforce Connect maps the external IDs of the record to Salesforce IDs. Mappings classified as short-term refer to records retrieved via Search results only. Mappings classified as long-term refer to records retrieved by all other Lightning Experience components, such as List Views. Record IDs not viewed in 365 days are subject to deletion unless the mappings have customer data attached to them.

See Also

- [General Limits for Salesforce Connect](#)
[OData Adapters for Salesforce Connect](#)

Set Up Salesforce Connect to Access External Data with OData Adapters

Let users view and search data that's stored outside your Salesforce org, such as data in an enterprise resource planning (ERP) system.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit external data sources:	Customize Application
To create and edit external objects:	Customize Application
To define or change object-level help:	Customize Application
To create and edit custom fields:	Customize Application
To edit permission sets and user profiles:	Manage Profiles and Permission Sets
To edit another user's authentication settings for external systems:	Manage Users

Setting up Salesforce Connect with an OData 2.0, 4.0, or 4.01 adapter involves these high-level steps.

1. Define an external data source of type `Salesforce Connect: OData 2.0` or `Salesforce Connect: OData 4.0`, or Define an external data source of type `Salesforce Connect: OData 4.01`.

If your external system hosts multiple services, create an external data source for each service endpoint. Each service endpoint points to an OData service root URL and can expose collections of entities. For example, you'd create a separate external data source for each of these service endpoints.

- `http://services.example.org/Warehouse.svc`
- `https://services.example.org/Payroll.svc`

2. Create the external objects.

Perform this task only if you don't sync to automatically create the external objects. Create an external object for each external data table that you want to access from your Salesforce org.

3. Add custom fields and relationships to the external objects.

Create relationships between objects. If you didn't sync to automatically create the external objects and their fields, create a custom field for each external table column that you want to access from your Salesforce org.

4. Verify access to external object data.

Check that expected user and code interactions with the external objects work, including sorting and filtering search and query results.

 **Tip** After you configure an external data source, run the validator tool on each external object to test and troubleshoot its connections. The tool tests for ID uniqueness and the ability to sort and filter results.

5. Enable user access to external objects.

Grant object permissions through permission sets or profiles.

6. Enable user access to the fields on the external objects.

Grant field permissions through permission sets or profiles.

7. If the external data source uses per-user authentication:

- a. Let users authenticate to the external system.

Grant users access to authentication settings for the external data source through permission sets or profiles.

- b. Set up each user's authentication settings.

You or your users can perform this task.

 **Tip** Train your users on how to set up their authentication settings for external systems. Make sure that they know which credentials to enter for each external system. If you're using OAuth 2.0, test the OAuth flow for potentially confusing prompts or redirects, and train your users as needed. OAuth flows vary, depending on your external system, authentication provider, and specified scopes.

Define an External Data Source for Salesforce Connect OData 4.01 Adapter

Connect your Salesforce org to data that's stored in an external system, such as SAP® NetWeaver Gateway, Microsoft Dynamics® NAV, or IBM WebSphere®.

See Also

[OData Adapters for Salesforce Connect](#)

[Salesforce Platform Features Supported by Salesforce Connect](#)

[Developer Guide: Visualforce Developer Guide](#)
[External Object Relationships](#)

Define an External Data Source for Salesforce Connect–OData 2.0 or 4.0 Adapter

Connect your Salesforce org to data that's stored in an external system, such as SAP® NetWeaver Gateway, Microsoft Dynamics® NAV, or IBM WebSphere®.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit external data sources: Customize Application

 **Note**

- The external data must be exposed by a service that uses Open Data Protocol (OData) Version 2.0 or 4.0. Such a service is called an *OData producer*.
- The URL for reaching the OData producer must be accessible by Salesforce application servers through the Internet. You can access by adding Salesforce server IP addresses on your corporate network firewall to the allow list or by setting up a reverse-proxy XML Gateway.
- The extent to which you can customize data visibility depends on the external system. To determine the optimal settings for integration with Salesforce, consult the external system's documentation.

1. From Setup, enter *External Data Sources* in the Quick Find box, then select **External Data Sources**.
2. Click **New External Data Source**, or click **Edit** to modify an existing external data source.
3. Complete the fields.

Field	Description
External Data Source	A user-friendly name for the external data source. The label is displayed in the Salesforce user interface, such as in list views. If you set Identity Type to Per User, this label appears when your users view or edit their authentication settings for external systems.

Field	Description
Name	<p>A unique identifier that's used to refer to this external data source definition through the API.</p> <p>The name can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.</p>
Type	Select Salesforce Connect: OData 2.0 or Salesforce Connect: OData 4.0 .
URL	<p>The OData service root URL. Make sure that you escape all special characters.</p> <p>Each service endpoint requires its own external data source definition, but you can have multiple entities under one service root URL. For more information about the service root URL and other URL conventions, go to www.odata.org.</p> <p>Examples:</p> <ul style="list-style-type: none"> • <code>http://services.example.org/Warehouse.svc</code> • <code>https://services.example.org/Payroll.svc</code> <p>If the endpoint is defined in a named credential, enter the named credential URL. A named credential URL contains the scheme <code>callout:</code>, the name of the named credential, and an optional path. For example: <code>callout: My_Named_Credential / some_path</code>.</p> <p>You can append a query string to a named credential URL. Use a question mark (?) as the separator between the named credential URL and the query string. For example: <code>callout: My_Named_Credential / some_path ?format=json</code>.</p> <p>If you enter a named credential URL, skip the Authentication section for the external data source. To access the external system, Salesforce Connect uses the authentication settings that are defined in the named credential.</p>
Connection Timeout	<p>Number of seconds to wait for a response from the external system before timing out. By default, the value is set to the maximum of 120 seconds.</p> <p>Depending on the availability of and the connection to the external system, it can take a long time to retrieve external data. Use this field to limit how long to wait for external data to load into your org.</p>
Writable External	The Lightning Platform and users in this org can create, update, and delete

Field	Description
Objects	records for external objects associated with the external data source. The external object data is stored outside the org. By default, external objects are read only.
High Data Volume	<p>Salesforce enforces rate limits for retrieving and viewing data from external systems. If your org hits rate limits when accessing external objects, consider selecting the High Data Volume option on the associated external data sources. Doing so bypasses most rate limits, but some special behaviors and limitations apply. See High Data Volume Considerations for Salesforce Connect–OData 2.0 and 4.0 Adapters.</p> <p>High-data-volume external data sources are still limited to 20,000 OData queries per hour for Enterprise, Performance, and Unlimited Editions. If you require higher limits, create a support case. See callout limits in General Limits for Salesforce Connect OData Adapters.</p>
Server Driven Pagination	It's common for Salesforce Connect queries of external data to have a large result set that's broken into smaller batches or pages. To have the external system control the paging behavior, select this option. See Client-Driven and Server-Driven Paging for Salesforce Connect OData Adapters .
Request Row Counts	<p>Includes one of the following system query options in each OData query so that the response includes the total row count of the result set.</p> <ul style="list-style-type: none"> • <code>\$inlinecount=allpages</code> for the OData 2.0 adapter • <code>\$count=true</code> for the OData 4.0 adapter <p>Some external systems don't support these system query options. If you receive errors or notice long load times when you try to access their data, deselect Request Row Counts on the external data source. If you do so, however, the external data source and its associated external objects can't support the following functionality, which requires the total row count.</p> <ul style="list-style-type: none"> • SOQL <code>COUNT()</code> aggregate function • Batch Apex with <code>Database.QueryLocator</code>
Compress Requests	When selected, Salesforce sends compressed HTTP requests to the external system. Make sure that the external system is set up to receive gzip-compressed data. Salesforce automatically accepts gzip-compressed responses.
Enable Search	Determines whether SOSL and Salesforce global searches also query the external objects that are associated with this external data source.

Field	Description
	<p>When selected, you can control which external objects are searchable by selecting or deselecting Allow Search on each external object.</p> <p>Only text, text area, and long text area fields on external objects can be searched. If an external object has no searchable fields, searches on that object return no records.</p> <p>To allow the external data source's associated external objects to appear in Salesforce for iOS and Salesforce mobile web when used on iOS devices, select this option.</p>
Custom Query Option for Salesforce Search	<p>Available only for the OData 2.0 adapter for Salesforce Connect. If the OData producer has implemented and exposed a free-text-search custom query option, enter the name of that query string parameter.</p> <p>Learn more about OData custom query options and other URI conventions at www.odata.org.</p> <p>This field has no effect when Enable Search is deselected or when the OData producer isn't set up to correctly handle the custom query option.</p> <p>See OData 2.0 Query Options.</p>
Use Free-Text Search Expressions	<p>Available for the OData 4.0 and 4.01 adapters for Salesforce Connect. To use the <code>\$search</code> system query option instead of <code>\$filter</code> in search requests that are sent to the external system, select this option. Make sure that the OData producer is set up to support the <code>\$search</code> system query option.</p> <p>This field has no effect when Enable Search is deselected.</p> <p>See OData 4.0 and 4.01 Query Options.</p>
Format	<p>The format that the OData producer uses to represent resources, such as collections of data.</p> <p>Make sure that the OData producer is set up to support the selected format. Learn more about representation formats and operations at www.odata.org.</p> <p>If your external data source uses the OData 4.0 adapter and JSON format, make sure that the OData producer accepts headers that contain the <code>odata.metadata=full</code> format parameter. Other variations, including</p>

Field	Description
	<code>odata.metadata=minimal</code> , aren't supported for the OData 4.0 adapter.
Special Compatibility	Select Socrata only if the URL specifies a Socrata open data endpoint. See Socrata™ Considerations for Salesforce Connect–OData 2.0 and 4.0 Adapters .
Display Server Errors	Specify whether to display error messages from an external system in the user interface.
Eligible for External Change Data Capture	Select to track changes to external data. See Use External Change Data Capture to Track Data Changes on External Objects .
CSRF Protection	<p>If the external system requires Cross-Site Request Forgery (CSRF) protection in requests to create, edit or delete its data, select this option. If you do so, your org obtains an anti-CSRF token and cookie from the external system and includes them in each create, edit, and delete request. See CSRF Considerations for Salesforce Connect–OData 2.0 and 4.0 Adapters.</p> <p>Available only when Writable External Objects is selected.</p>
Anti-CSRF Token Name	<p>HTTP header field that contains the anti-CSRF token. The external system determines the field name. Default: X-CSRF-Token</p> <p>Available only when CSRF Protection is selected.</p>
Certificate	If you specify a certificate, your Salesforce org supplies it when establishing each two-way SSL connection with the external system. The certificate is used for digital signatures, which verify that requests are coming from your Salesforce org.
Identity Type	<p>Determines whether you're using one set or multiple sets of credentials to access the external system. See Identity Type for External Data Sources.</p> <p>Select Anonymous only if the external system doesn't require authentication.</p>

4. Select the authentication protocol.

- If you select **Password Authentication**, enter the username and password for accessing the external system.
- If you select **OAuth 2.0**, complete these fields.

Field	Description
Authentication	Choose the provider. See Authentication Providers .

Field	Description
Provider	
Scope	<p>Specifies the scope of permissions to request for the access token. Your authentication provider determines the allowed values. See Use the Scope Parameter.</p> <p>Keep these considerations in mind when you set a scope.</p> <ul style="list-style-type: none"> The value that you enter replaces the Default Scopes value that's defined in the specified authentication provider. Whether scopes are defined can affect whether each OAuth flow prompts the user with a consent screen. We recommend that you request a refresh token or offline access. Otherwise, when the token expires, you lose access to the external system.
Start Authentication Flow on Save	<p>To authenticate to the external system and obtain an OAuth token, select this checkbox. This authentication process is called an OAuth flow.</p> <p>When you click Save, the external system prompts you to log in. After successful login, the external system grants you an OAuth token for accessing its data from this org.</p> <p>Redo the OAuth flow when you need a new token—for example, if the token expires—or if you edit the Scope or Authentication Provider fields. When the token expires, the external system returns a 401 HTTP error status.</p>

- Click **Save**.
 - Click **Validate and Sync**, and confirm that the connection is successful.
 - Optionally, select tables and click **Sync** to do the following for each selected table.
 - Automatically create a Salesforce external object.
 - Automatically create a custom field for each table column that's compatible with a Salesforce metadata field type.
-  **Note** Before you sync, make sure that you understand the considerations that are described in these topics.
- [Sync an External Data Source for Salesforce Connect](#)
 - [Sync Considerations for Salesforce Connect OData Adapters](#)

You can instead choose to manually create the external objects and custom fields that map to the external data. Doing so lets you customize the external object names, decide which table columns to create custom fields for, and customize the custom field names. However, this approach takes longer and requires manual maintenance.

Optionally, you can associate custom HTTP headers to the external data source to retrieve or request additional data.

See Also

- [Set Up Salesforce Connect to Access External Data with OData Adapters](#)
- [Store Authentication Settings for External Systems](#)
- [OData Query String Options](#)
- [OData Type Mapping](#)
- [Named Credentials](#)

Define Custom HTTP Headers for OData Connectors

Define and activate custom HTTP headers to pass more request information to the external data source for processing.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit external data sources: Customize Application

Custom HTTP headers provide context information from Salesforce such as region, org details, or the role of the person viewing the external object. For each OData external data source, define up to 10 HTTP headers to request or data. For example, to see who's making a callout to the external data source, use a formula that resolves to the name of the user.

When naming the custom HTTP header, don't override the following existing standard header names.

- Content-Type
- Accept
- maxVersionHeader
- versionHeader
- X-HTTP-METHOD
- Content-Length
- X-CSRF-Token (when CSRF-enabled)
- Prefer (when a trigger on the external object is enabled)
- Cookie

1. From Setup, enter *External Data Sources* in the Quick Find box, then select **External Data**

Sources.

2. Click the name of an OData 2.0 or 4.0 data source.
3. In the External Custom HTTP Headers related list, click **New** to create a custom HTTP header or **Edit** to change an existing one.
4. Complete the fields.

Field	Description
Header Field Name	Enter a name that contains at least one alphanumeric character or underscore. It can also include: ! # \$ % & ' * + - . ^ _ ` ~
Header Field Value	Create a formula for the Header Field Value using the formula editor. The values in the formula must evaluate to a string. If the formula resolves to null and an empty string, the header isn't sent.
Active	To start using the header field right away, select the Active checkbox.
Description	A text description of the header field's purpose.
Parent	The name of the entity that the custom HTTP header is related to. Using a named credential as the parent entity for the custom HTTP header isn't supported.

5. Click **Save**.



Note You can't add or delete Custom HTTP headers that are included in a managed package. However, you can edit custom HTTP headers that are part of a managed package. The Header Name and Description fields are developer editable. The Active and Header Field Value fields are both subscriber and developer editable.

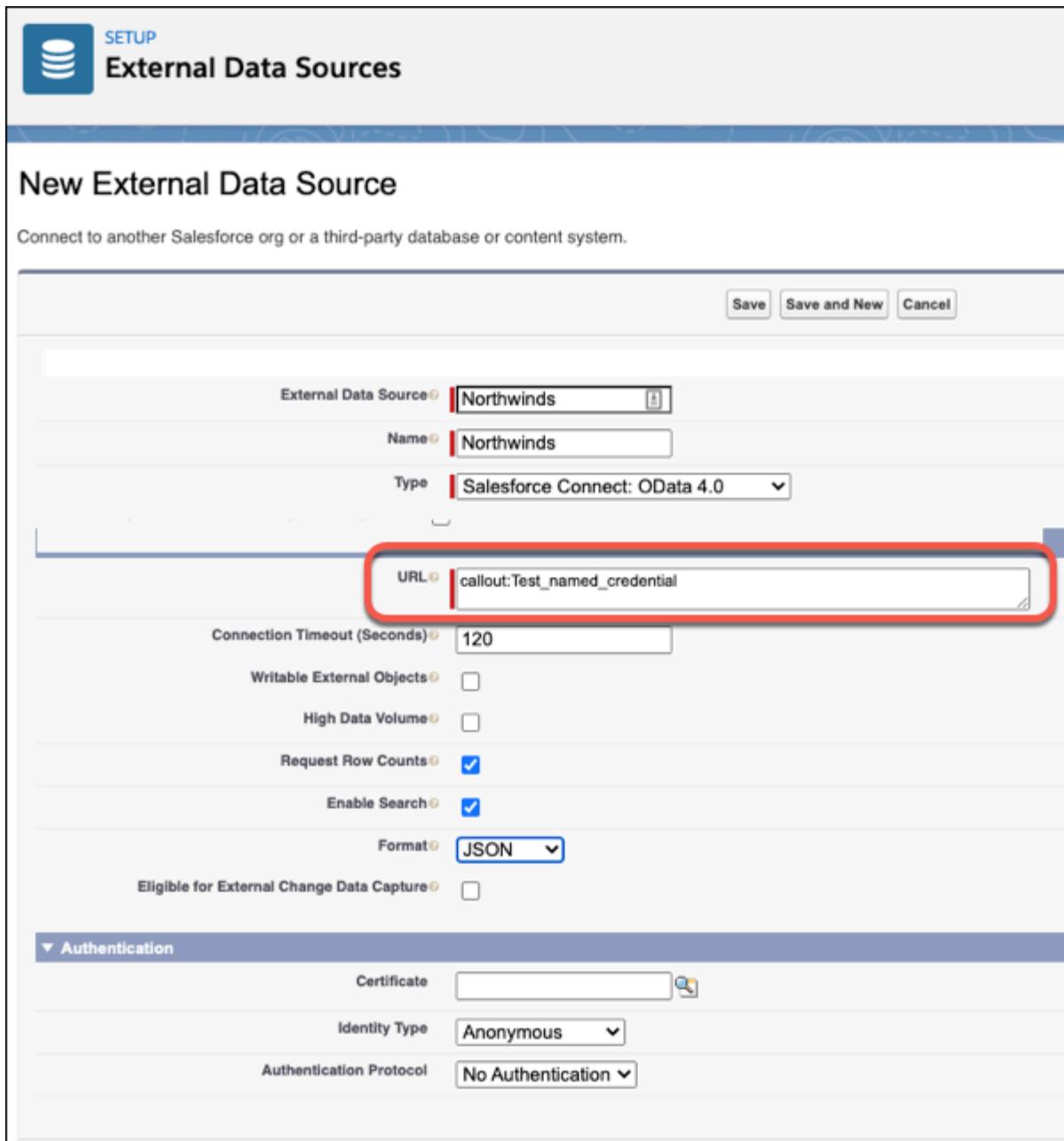
Named Credential as Callouts for Salesforce Connect OData 2.0 or 4.0 Adapters

When you define an external data source for the Salesforce Connect OData 2.0 or 4.0 adapters, you can enter a named credential URL as the URL for reaching the OData producer. This example shows how Salesforce Connect OData 2.0 or 4.0 adapters can make callouts using the authentication settings defined in a named credential.

Define a named credential that specifies the endpoint URL and the JWT authentication settings.

The screenshot shows the 'Named Credentials' setup page in Salesforce. At the top, there's a 'SETUP' button and a 'Named Credentials' section. Below that, a heading says 'New Named Credential'. A sub-instruction says 'Specify the callout endpoint's URL and the authentication settings that are required for Salesforce to make callouts to the remote system.' On the right, there are 'Save' and 'Cancel' buttons. The main form has three fields: 'Label' (containing 'Test named credential'), 'Name' (containing 'Test_named_credential'), and 'URL' (containing 'https://test_endpoint.example.com'). Below this is a section titled 'Authentication' with three dropdowns: 'Certificate' (empty), 'Identity Type' (set to 'Named Principal'), and 'Authentication Protocol' (set to 'JWT').

When you're defining an external data source with an Odata 2.0 or Odata 4.0 adapter, specify the named credential you defined as the OData service root URL. In this example, the **URL** is `callout:Test_named_credential`. And skip the **Authentication** section for the external data source. To access the external system, Salesforce Connect uses the authentication settings defined in the named credential.



The screenshot shows the 'External Data Sources' setup page in Salesforce. A new external data source is being created for 'Northwinds'. The 'Type' is set to 'Salesforce Connect: OData 4.0'. The 'URL' field contains 'callout:Test_named_credential' and is highlighted with a red box. Other settings include 'Connection Timeout (Seconds)' at 120, 'Request Row Counts' checked, 'Enable Search' checked, 'Format' set to JSON, and 'Authentication' configured with 'Identity Type' as 'Anonymous' and 'Authentication Protocol' as 'No Authentication'.

See Also

[Named Credentials](#)

[Store Authentication Settings for External Systems](#)

[Grant Access to Authentication Settings for Legacy Named Credentials](#)

Define an External Data Source for Salesforce Connect OData 4.01 Adapter

Connect your Salesforce org to data that's stored in an external system, such as SAP® NetWeaver Gateway, Microsoft Dynamics® NAV, or IBM WebSphere®.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit external data sources: Customize Application

1. From Setup, enter *External Data Sources* in the Quick Find box, then select **External Data Sources**.
2. Click **New External Data Source**, or click **Edit** to modify an existing external data source.
3. Complete the fields.

Field	Description
External Data Source	A user-friendly name for the external data source. The label is displayed in the Salesforce user interface, such as in list views.
Name	A unique identifier that's used to refer to this external data source definition through the API. The name can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
Type	Select Salesforce Connect: OData 4.01 .
Named Credential	Enter the named credential URL defined for the data source exposed via OData 4.01. To know how to create an external credential and named credential, see Named Credentials .
Connection Timeout (Seconds)	Number of seconds to wait for a response from the external system before timing out. By default, the value is set to the maximum of 120 seconds.
Writable External Objects	Lets the Lightning Platform and users in this org create, update, and delete records for external objects associated with the external data source. The external object data is stored outside the org. By default, external objects are read only.
Server Driven Pagination	Select this option to have the external system control the paging behavior. See Client-Driven and Server-Driven Paging for Salesforce Connect OData Adapters .

Field	Description
Enable Search	<p>Determines whether SOSL and Salesforce global searches also query the external objects that are associated with this external data source.</p> <p>When selected, you can control which external objects are searchable by selecting or deselecting Allow Search on each external object.</p> <p>Only text, text area, and long text area fields on external objects can be searched. If an external object has no searchable fields, searches on that object return no records.</p> <p>Select this option to allow the external data source's associated external objects to appear in Salesforce for iOS and Salesforce mobile web when used on iOS devices.</p>
Use Free-Text Search Expressions	<p>Available for the OData 4.0 and 4.01 adapters for Salesforce Connect. Select this option to use the <code>\$search</code> system query option instead of <code>\$filter</code> in search requests that are sent to the external system. Make sure that the OData producer is set up to support the <code>\$search</code> system query option.</p> <p>This field has no effect when Enable Search is deselected.</p> <p>See OData 4.0 and 4.01 Query Options.</p>
Eligible for External Change Data Capture	Select to track changes to external data. See Use External Change Data Capture to Track Data Changes on External Objects .
Batch DML	Select this option to be able to send multiple DML requests to the external data source in JSON batch format. See batch requests .
Metadata Control Information	Select <i>Full</i> or <i>Minimal</i> as per your client application and device requirements.
Display Server Errors	Select to specify whether to display error messages from an external system in the user interface.

4. Click **Save**.
5. Click **Validate and Sync**, select the tables, and click **Sync**. Before you sync, make sure that you understand the considerations described in [Sync an External Data Source for Salesforce Connect](#) and [Sync Considerations for Salesforce Connect OData Adapters](#)

 **Note** To leverage [Named Credentials](#) and [Private Connect](#) with OData 4.0 data sources, use the OData 4.01 adapter for Salesforce Connect.

Considerations for Salesforce Connect OData Adapters

Understand the special behaviors, limits, and recommendations for using the OData 2.0, 4.0, and 4.01 adapters for Salesforce Connect.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Also review the considerations that apply to all Salesforce Connect adapters.

[Lightning Experience Considerations for Salesforce Connect OData Adapters](#)

Users can access external objects from the Lightning Experience app. But some requirements and special behaviors apply when the external data is accessed via the OData 2.0, 4.0, or 4.01 adapter for Salesforce Connect.

[Writable External Objects Considerations for Salesforce Connect OData Adapters](#)

Some special behaviors and limitations affect writable external objects that are associated with OData 2.0, 4.0, and 4.01 adapters for Salesforce Connect.

[Sync Considerations for Salesforce Connect OData Adapters](#)

When you validate and sync an external data source of type OData 2.0, 4.0, or 4.01, some special behaviors and limitations apply.

[OData Producer Considerations for Salesforce Connect OData Adapters](#)

Understand the limits and recommendations for the remote data service that exposes the external data via OData 2.0, 4.0, and 4.01 to your Salesforce org.

[High Data Volume Considerations for Salesforce Connect–OData 2.0 and 4.0 Adapters](#)

If your org hits rate limits when accessing external objects, consider selecting the **High Data Volume** option on the associated external data sources. Doing so bypasses most rate limits, but some special behaviors and limitations apply.

[Socrata™ Considerations for Salesforce Connect–OData 2.0 and 4.0 Adapters](#)

Socrata Open Data Protocol™ is commonly used for health data and for collaboration between governments and their citizens. Salesforce Connect can access data from endpoints that are backed by Socrata Open Data Portal. To accommodate Socrata-specific requirements, set the **Special Compatibility** field on the external data source to Socrata.

[CSRF Considerations for Salesforce Connect–OData 2.0 and 4.0 Adapters](#)

Understand the special behaviors, limitations, and recommendations for Cross-Site Request Forgery (CSRF) on OData external data sources.

See Also

[Salesforce Platform Features Supported by Salesforce Connect](#)

Lightning Experience Considerations for Salesforce Connect OData Adapters

Users can access external objects from the Lightning Experience app. But some requirements and special behaviors apply when the external data is accessed via the OData 2.0, 4.0, or 4.01 adapter for Salesforce Connect.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- If external object records don't appear in your org, make sure that the OData producer doesn't change the values specified in the OData query filters. When your org sends OData queries that specify field values with the `$filter equals (eq)` operator, the OData producer must return those same field values in the resulting data rows.
- If external object records don't appear in relationship fields or related lists, check for case-sensitive values. For example, suppose that you set up an indirect lookup relationship. If the external system uses case-sensitive values in the specified External Column Name, make sure that the parent object field is also case-sensitive. When you define the parent object's custom field, select **External ID**, **Unique**, and **Treat "ABC" and "abc" as different values (case sensitive)**.

See Also

[External Object Relationships](#)

[OData Query String Options](#)

[OData Adapters for Salesforce Connect](#)

[Define an External Data Source for Salesforce Connect–OData 2.0 or 4.0 Adapter](#)

[Define an External Data Source for Salesforce Connect OData 4.01 Adapter](#)

Writable External Objects Considerations for Salesforce Connect OData Adapters

Some special behaviors and limitations affect writable external objects that are associated with OData 2.0, 4.0, and 4.01 adapters for Salesforce Connect.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- An external object custom field associated with an OData complex type on the external system is always read only, even if the external object is writable.
- Writable external objects aren't available for high-data-volume external data sources.
- If your external system requires write operations to specify values for entity keys, ensure that **External ID** standard field values and entity key values don't contradict each other. For each write operation, include either the External ID standard field value or the custom field values that form the entity key, but never both.
- When an external object record is edited, Salesforce Connect sends an HTTP request to the external system.
 - If the record is edited from the Salesforce user interface, the HTTP request includes all fields, including the fields that weren't changed.
 - If the record is edited from the API, only the specified fields are included in the HTTP request.
- Make sure that the OData producer supports the HTTP methods that are used by your Salesforce Connect adapter.

To Do This	OData 4.01 Adapter Uses This HTTP Method	OData 4.0 Adapter Uses This HTTP Method	OData 2.0 Adapter Uses This HTTP Method
Create record	POST	POST	POST
Edit record	PATCH	POST with X-HTTP-METHOD header set to PATCH	POST with X-HTTP-METHOD header set to MERGE
Delete record	DELETE	DELETE	DELETE
View record	GET	GET	GET

Also review the considerations that apply to all Salesforce Connect adapters.

See Also

[Writable External Objects in Salesforce Connect](#)

[External IDs and OData Entity Keys](#)

[OData Adapters for Salesforce Connect](#)

Sync Considerations for Salesforce Connect OData Adapters

When you validate and sync an external data source of type OData 2.0, 4.0, or 4.01, some special behaviors and limitations apply.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer Edition**

Available for an extra cost in: **Enterprise, Performance, and Unlimited** Editions

Syncing always enables search on the external object when search is enabled on the external data source, and vice versa.

 **Important** When you select the tables to sync, determine whether check marks appear in the Synced column. If a Synced check mark appears, your org has an external object whose object name matches the table name. If you select the table and click **Sync**:

- The external object is overwritten.
- Any custom field on the external object is overwritten if its API name (for example, `Email__c`) associates it with a table column name (for example, `Email`).
- Any other custom fields on the external object remain as they are, including:
 - Previously synced custom fields whose API names were changed by editing their **Field Name** values.
 - Manually added custom fields whose API names aren't associated with table column names.

If no Synced check mark appears, and you sync the table, a new external object is created in your org. The new external object's object name matches the table name. For example, if the table name is changed on the external system to no longer match the object name of the external object, syncing that table creates a new external object in Salesforce. We recommend that you change the object name of the existing external object to match the new table name on the external system before you sync that table.

Also review the considerations that apply to all Salesforce Connect adapters.

See Also

[Sync an External Data Source for Salesforce Connect](#)

[Define an External Data Source for Salesforce Connect—OData 2.0 or 4.0 Adapter](#)

[Define an External Data Source for Salesforce Connect OData 4.01 Adapter](#)

[OData Adapters for Salesforce Connect](#)

[OData Type Mapping](#)

OData Producer Considerations for Salesforce Connect OData Adapters

Understand the limits and recommendations for the remote data service that exposes the external data via OData 2.0, 4.0, and 4.01 to your Salesforce org.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer Edition**

Available for an extra cost in: **Enterprise, Performance, and Unlimited** Editions

- Validate your OData producer by using the Open Data Protocol Service Validation Tool at services.odata.org/validation. Doing so checks your implementation against the OData specification and identifies potential issues.
- To improve performance over low-bandwidth connections, set up your OData producer to receive gzip-compressed data. Then, in the external data source definition for OData 2.0 or 4.0, select **Compress Requests**. You can also set up the OData producer to send gzip-compressed data to Salesforce, which automatically accepts gzip-compressed responses.
- By default, Salesforce sends each OData request with one of the following system query options so that the response includes the total row count of the result set.
 - `$inlinecount=allpages` for the OData 2.0 adapter
 - `$count=true` for the OData 4.0 adapter
 - `$count=true` hard coded for 4.01 adapter

Some external systems don't support these system query options. If you receive errors or notice long load times when you try to access their data, deselect **Request Row Counts** for OData 2.0 or 4.0, on the external data source. If you do so, the external data source and its associated external objects can't support the following functionality, which requires the total row count.

- SOQL `COUNT()` aggregate function
- Batch Apex with [Database.QueryLocator](#)

For details about OData URI conventions, go to www.odata.org.

- Configure your OData producer to use a page size that's large enough to avoid excessive round trips. Querying a large set of data with a small page size can take a long time because of network latency. Salesforce pages that display external data can take a long time to load.

For example, if the query results include 100 records, and the page size holds only 5 records, it takes 20 round trips to retrieve the results. If the network latency is 100 ms per round trip, it takes 2 seconds (20×100 ms) to retrieve the results.

In contrast, if the page size holds 20 records, it takes only five round trips to retrieve the 100 records. With the same network latency of 100 ms per round trip, it takes 0.5 seconds (5×100 ms) to retrieve the results.

- If your external data source uses the OData 4.0 adapter and JSON format, make sure that the OData producer accepts headers that contain the `odata.metadata=full` format parameter. Other variations, including `odata.metadata=minimal`, aren't supported for OData 4.0 adapter.
- If external object records don't appear in your org, make sure that the OData producer doesn't change the values specified in the OData query filters. When your org sends OData queries that specify field values with the `$filter equals (eq)` operator, the OData producer must return those same field values in the resulting data rows.

See Also

[OData Adapters for Salesforce Connect](#)

[OData Query String Options](#)

[OData Type Mapping](#)

High Data Volume Considerations for Salesforce Connect–OData 2.0 and 4.0 Adapters

If your org hits rate limits when accessing external objects, consider selecting the **High Data Volume** option on the associated external data sources. Doing so bypasses most rate limits, but some special behaviors and limitations apply.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- The following features aren't available for external objects that are associated with high-data-volume external data sources.
 - Access via Lightning Experience
 - Access via the Salesforce mobile app
 - Appearance in Recent Items lists
 - Record feeds
 - Reports and dashboards
 - Writable external objects
- Salesforce IDs aren't assigned to external object records that are associated with high-data-volume external data sources.
- On record detail pages for external objects that are associated with high-data-volume external data sources, custom buttons, and links that call JavaScript aren't supported.
- Salesforce Console in Salesforce Classic doesn't support external objects that are associated with high-data-volume external data sources.
- CSRF protection for writable external objects isn't available for high-data-volume external data sources.

See Also

[REST API Developer Guide : List Organization Limits](#)

[CSRF Considerations for Salesforce Connect–OData 2.0 and 4.0 Adapters](#)

[Define an External Data Source for Salesforce Connect–OData 2.0 or 4.0 Adapter](#)

Socrata™ Considerations for Salesforce Connect–OData 2.0 and 4.0 Adapters

Socrata Open Data Protocol™ is commonly used for health data and for collaboration between governments and their citizens. Salesforce Connect can access data from endpoints that are backed by Socrata Open Data Portal. To accommodate Socrata-specific requirements, set the **Special Compatibility** field on the external data source to Socrata.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Socrata doesn't support the row identifier (`_id`) column in `$select` or `$orderby` clauses in OData queries. When Socrata is selected in the **Special Compatibility** field:

- OData queries don't include the `_id` column in `$select` clauses.
- If an `_id` column is synced from a Socrata endpoint, the resulting custom field on the external object isn't sortable.
- If you manually define an external object's custom field with `_id` as the External Column Name, make sure that you select the **Sorting Disabled** attribute for that custom field.

If you modify the Special Compatibility field on an external data source, we recommend that you resync its external objects. Or instead, you can test whether queries or user access to the external objects result in errors, and resync only the problematic external objects.

See Also

[Define an External Data Source for Salesforce Connect–OData 2.0 or 4.0 Adapter](#)
[Considerations for Salesforce Connect OData Adapters](#)

[CSRF Considerations for Salesforce Connect–OData 2.0 and 4.0 Adapters](#)

Understand the special behaviors, limitations, and recommendations for Cross-Site Request Forgery (CSRF) on OData external data sources.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- CSRF protection isn't available for high-data-volume external data sources.
- Make sure that the URL of the external data source starts with `https://` so that secure HTTP can prevent unauthorized access to the anti-CSRF token and cookie.
- In addition to enabling CSRF protection on the external data source, we recommend keeping CSRF protection enabled in your org's session security settings. These session settings are enabled by

default, and keeping them enabled protects your Salesforce data and your external data from CSRF attacks.

- **Enable CSRF protection on GET requests on non-setup pages**
- **Enable CSRF protection on POST requests on non-setup pages**

See Also

[Define an External Data Source for Salesforce Connect–OData 2.0 or 4.0 Adapter](#)

[Modify Session Security Settings](#)

Picklist Considerations for Salesforce Connect OData Adapters

Special behaviors and limitations apply to picklist fields on external objects for OData 2.0, 4.0, and 4.01 adapters.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- Global and local single-select picklists are supported.
- You must manually update picklist values to stay in sync between your org and the external system. If values aren't in sync, your users could see an error message when viewing the field.
 - Add values to the picklist as you would a picklist on a custom object.
 - Remove a value from a picklist by deactivating the value rather than deleting it.
 - Replace a value by deactivating it and then adding the new value. Update the external system's record values to the new value on the external system.
- Convert existing external object text fields to picklists by deleting and recreating. Delete the text field and create a picklist pointing to the text field's existing External Column Name.
- If you select to delete, you may be prompted to replace the value with another value. No replace occurs for external object record values. You must manually change the values on the external system.
- We recommend having only unrestricted picklists on external objects, even when they're associated with restricted picklists on the provider org. Restricted picklists on the provider org block unapproved values from the subscriber org, eliminating the need to restrict picklists on external objects.

See Also

[Picklist Considerations for Salesforce Connect–Cross-Org Adapter](#)

OData Reference for Salesforce Connect OData Adapters

Get to know the Salesforce implementation of the OData protocol for accessing external systems with Salesforce Connect.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

[OData Type Mapping](#)

Salesforce Connect maps OData types to Salesforce metadata field types when syncing metadata and converting values between Salesforce and external systems.

[OData Query String Options](#)

The OData adapters for Salesforce Connect use a subset of the OData 2.0 and 4.0 and 4.01 system functions and filter expression constructs to query external systems.

See Also

[OData Adapters for Salesforce Connect](#)

OData Type Mapping

Salesforce Connect maps OData types to Salesforce metadata field types when syncing metadata and converting values between Salesforce and external systems.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

[OData 2.0 Type Mapping](#)

Understand how the OData 2.0 adapter for Salesforce Connect maps OData types to Salesforce metadata field types.

[OData 4.0 and 4.01 Type Mapping](#)

Understand how the OData 4.0 and 4.01 adapters for Salesforce Connect maps OData types to Salesforce metadata field types.

OData 2.0 Type Mapping

Understand how the OData 2.0 adapter for Salesforce Connect maps OData types to Salesforce metadata field types.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Salesforce Connect supports only the following types when syncing metadata and converting values between Salesforce and an external system.

OData 2.0 Primitive Types

OData 2.0 Type	Salesforce Metadata Field Type
Binary	TextArea
Boolean	Checkbox
Byte	Number
DateTime	DateTime
DateTimeOffset	DateTime
Decimal	Number
Double	Number
Guid	Text
Int16	Number
Int32	Number
Int64	Number
SByte	Number
Single	Number
String	<p>Depends on the declared maximum length of the OData string column.</p> <ul style="list-style-type: none"> • OData declared maximum length: not declared <ul style="list-style-type: none"> - Salesforce field type: Text - Salesforce field length attribute: 128 characters • OData declared maximum length: 255 or fewer characters <ul style="list-style-type: none"> - Salesforce field type: Text

OData 2.0 Type	Salesforce Metadata Field Type
	<ul style="list-style-type: none"> - Salesforce field length attribute: Same as declared • OData declared maximum length: More than 255 characters <ul style="list-style-type: none"> - Salesforce field type: LongTextArea - Salesforce field length attribute: Same as declared
Time	Text

 **Tip** A binary value from an external system is represented in Salesforce as a base64-encoded string. You can convert it to a value of type `Blob` by using the `EncodingUtil.base64Decode(inputString)` Apex method.

OData 2.0 Complex Types

Salesforce Connect supports OData complex types as follows.

See Also

- [Sync an External Data Source for Salesforce Connect](#)
- [Custom Field Types](#)
- [Custom Field Attributes](#)
- [OData Reference for Salesforce Connect OData Adapters](#)
- [Apex Developer Guide : EncodingUtil Class: `base64Decode\(inputString\)`](#)
- [Visualforce Developer Guide : `apex:image`](#)

OData 4.0 and 4.01 Type Mapping

Understand how the OData 4.0 and 4.01 adapters for Salesforce Connect maps OData types to Salesforce metadata field types.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Salesforce Connect supports only the following types when syncing metadata and converting values between Salesforce and an external system.

OData 4.0 and 4.01 Primitive Types

OData 4.0 and 4.01 Type	Salesforce Metadata Field Type
Binary	TextArea
Boolean	Checkbox
Byte	Number
Date	DateTime with time properties set to 0
DateTimeOffset	DateTime
Decimal	Number
Double	Number
Guid	Text
Int16	Number
Int32	Number
Int64	Number
Byte	Number
Single	Number
String	<p>Depends on the declared maximum length of the OData string column.</p> <ul style="list-style-type: none"> • OData declared maximum length: not declared <ul style="list-style-type: none"> - Salesforce field type: Text - Salesforce field length attribute: 128 characters • OData declared maximum length: 255 or fewer characters <ul style="list-style-type: none"> - Salesforce field type: Text - Salesforce field length attribute: Same as declared • OData declared maximum length: More than 255 characters <ul style="list-style-type: none"> - Salesforce field type: LongTextArea - Salesforce field length attribute: Same as declared



Tip A binary value from an external system is represented in Salesforce as a base64-encoded string. You can convert it to a value of type `Blob` by using the `EncodingUtil.base64Decode(inputString)` Apex method.

OData 4.0 and 4.01 Complex Types

Salesforce Connect supports OData complex types as follows.

See Also

[Sync an External Data Source for Salesforce Connect](#)

[Custom Field Types](#)

[Custom Field Attributes](#)

[OData Reference for Salesforce Connect OData Adapters](#)

[Apex Developer Guide : EncodingUtil Class: `base64Decode \(inputString\)`](#)

[Visualforce Developer Guide : `apex:image`](#)

OData Query String Options

The OData adapters for Salesforce Connect use a subset of the OData 2.0 and 4.0 and 4.01 system functions and filter expression constructs to query external systems.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Salesforce automatically creates the OData queries so that you, as an administrator or developer, don't have to. However, understanding how OData queries are generated—or even attempting manual OData queries—can help you troubleshoot issues with the external system's OData producer. For details about each system query option, go to www.odata.org.

Salesforce Connect supports only the following OData system query options. All other options in the OData 2.0, 4.0, and 4.01 specifications are unused.

- `$count` (OData 4.0 and 4.01)
- `$filter`
- `$inlinecount` (OData 2.0 only)
- `$orderby`
- `$search` (OData 4.0 and 4.01)
- `$select`
- `$skip`
- `$top`

A query search string to an external system is sent as a case-sensitive single phrase after removing all ASCII punctuation characters except hyphens (-). For example, if the search string is *Sales &*

Marketing, the external system receives *Sales Marketing*.

\$count (OData 4.0 and 4.01)

Specifies that the response must include the number of rows that the URI identifies after `$filter` system query options are applied, but before `$top` and `$skip` system query options are applied.

For OData 4.01, `$count=true` is hard coded and for OData 4.0, `$count=true` when **Request Row Counts** is enabled on the external data source. The total items in the result set is the same as the LIMIT value in the query for values less than 202. If the LIMIT value is greater than 202, then 202 items are returned to indicate that more records exist in the next batch. For OData 4.0, if **Request Row Counts** is disabled, Salesforce includes `$count=false` in all OData 4.0 queries of the external data source, and 2000 items are returned in each result set.

Examples	
User action in Salesforce	View or access an external object.
SOQL query	Any SOQL query of an external object
Resulting OData query	<code>http://services.example.org/my.svc/Suppliers?\$select=SupplierID&\$count=true&\$top=26</code>

\$filter

Filters the collection of resources that's addressed by a request URL. The response contains the results that evaluate to `true`.

Examples	
User action in Salesforce	Open a list view of cities from supplier records that are filtered so that the country is USA.
SOQL query	<code>SELECT City__c FROM Suppliers__x WHERE Country__c = 'USA' ORDER BY City__c ASC LIMIT 26</code>
Resulting OData query	<code>http://services.example.org/my.svc/Suppliers?\$orderby=City&\$select=City,SupplierID&\$inlinecount=allpages&\$filter=Country+eq+'USA'&\$top=26</code>

\$inlinecount (OData 2.0 only)

Specifies that the response must include a count of the number of rows that the URI identifies after `$filter` system query options are applied but before `$top` and `$skip` system query options are

applied.

When **Request Row Counts** is enabled on the external data source, Salesforce uses `$inlinecount` in all OData 2.0 queries of that external data source to determine the total number of items in each result set. If **Request Row Counts** is disabled, `$inlinecount` is excluded from all OData 2.0 queries of the external data source.

Examples	
User action in Salesforce	View or access an external object.
SOQL query	Any SOQL query of an external object.
Resulting OData query	<code>http://services.example.org/my.svc/Suppliers?\$select=SupplierID&\$inlinecount=allpages&\$top=26</code>

\$orderby

Sorts the result set in ascending or descending order. The fields in the ORDER BY clause of the SOQL query don't always match the properties used by the `$orderby` option in the resulting OData query. If you use the OFFSET clause in the SOQL query, the entity key property is added in the resulting OData query.

Examples	
User action in Salesforce	Open a list view of supplier records that are ordered by company name.
SOQL query	<code>SELECT CompanyName__c, ContactName__c FROM Suppliers__x ORDER BY CompanyName__c ASC LIMIT 26</code>
Resulting OData query	<code>http://services.example.org/my.svc/Suppliers?\$orderby=CompanyName&\$select=CompanyName,ContactName,SupplierID&\$inlinecount=allpages&\$top=26</code>

\$search (OData 4.0 and 4.01)

Requests entities that match the search query string as a free-text search expression. For OData 4.0, enable this option by selecting **Use Free-Text Search Expressions** on the external data source. By default, **Use Free-Text Search Expressions** isn't enabled. The search query string is used as the `contains` value in the `$filter` system query option.

Examples	
User action in Salesforce	View or access an external object.
SOQL query	Any SOQL query of an external object
Resulting OData query	<pre>http://services.example.org/my.svc/Shippers? \$select=CompanyName,Phone,ShipperID& \$count=true&\$search=Acme&\$top=25</pre>

\$select

Requests a limited set of properties for each entity.

Examples	
User action in Salesforce	Open a list view of supplier records where the page layout displays the company name and contact name.
SOQL query	<pre>SELECT CompanyName__c,ContactName__c FROM Suppliers__x ORDER BY CompanyName__c ASC LIMIT 26</pre>
Resulting OData query	<pre>http://services.example.org/my.svc/ Suppliers?\$orderby=CompanyName& \$select=CompanyName,ContactName,SupplierID&\$inlinecount=allpages&\$top =26</pre>

\$skip

Specifies the number of items in the queried collection to skip in the result set.

Examples	
User action in Salesforce	Click to view the second page of a list view of supplier records that are ordered by city.
SOQL query	<pre>SELECT City__c,CompanyName__c FROM Suppliers__x ORDER BY City__c ASC OFFSET 25</pre>
Resulting OData query	<pre>http://services.example.org/my.svc/Suppliers?\$orderby=City& \$select=City,CompanyName,SupplierID&\$inlinecount=allpages&\$top=25& \$skip=25</pre>

\$top

Specifies the number of items in the queried collection to include in the result. The value in the `LIMIT` clause of a SOQL query doesn't always match the requested `$top` value, because the latter is modified as needed for client-driven paging and `queryMore()` calls.

Examples	
User action in Salesforce	Open a list view of the top 25 supplier records.
SOQL query	<code>SELECT SupplierID__c FROM Suppliers__x LIMIT 25</code>
Resulting OData query	<code>http://services.example.org/my.svc/Suppliers?\$select=SupplierID&\$inlinecount=allpages&\$top=25</code>

OData 2.0 Query Options

By default, the search query string is used as the `substringof` value in the `$filter` system query option.

OData 4.0 and 4.01 Query Options

By default, the search query string is used as the `contains` value in the `$filter` system query option.

See Also

[OData Producer Considerations for Salesforce Connect OData Adapters](#)

[OData Reference for Salesforce Connect OData Adapters](#)

[Client-Driven and Server-Driven Paging for Salesforce Connect OData Adapters](#)

OData 2.0 Query Options

By default, the search query string is used as the `substringof` value in the `$filter` system query option.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

In this example, the search query string is `Acme`.

```
http://services.example.org/my.svc/Shippers?  
$select=CompanyName,Phone,ShipperID&$inlinecount=allpages&  
$filter=substringof('Acme',CompanyName)+eq+true+  
or+substringof('Acme',Phone)+eq+true&$top=25
```

OData 2.0 Custom Query Option

We recommend that you set up the OData producer to support free text search expressions with the custom query option. You can then specify the name of the query string parameter in the **Custom Query Option for Salesforce Search** field on the external data source. In this example, the custom query parameter is `doSearch` and the search query string is `Acme`.

```
http://services.example.org/my.svc/Shippers?  
$select=CompanyName,Phone,ShipperID&  
$inlinecount=allpages&doSearch=Acme&$top=25
```

Learn more about OData URI conventions at www.odata.org.

OData 4.0 and 4.01 Query Options

By default, the search query string is used as the `contains` value in the `$filter` system query option.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

In the following example, the search query string is `Acme`.

```
http://services.example.org/v4.svc/Shippers?  
$select=CompanyName,Phone,ShipperID&$count=true&  
$filter=contains(CompanyName,'Acme') eq true  
or contains(Phone,'Acme') eq true&$top=25&
```

OData 4.0 and 4.01 Custom Query Option

We recommend that you set up the OData producer to support free text search expressions with the

`$search` system query option. You can then select **Use Free-Text Search Expressions** on the external data source.

```
http://services.example.org/v4.svc/Shippers?  
$select=CompanyName,Phone,ShipperID&$count=true&  
$search=Acme&$top=25
```

<http://www.odata.org>

Use External Change Data Capture to Track Data Changes on External Objects

With External Change Data Capture, you can track changes to data that is stored outside your Salesforce org when using the OData 4.0 and 4.01 adapters. You can then build automation in response to the changes to increase productivity or provide a better customer experience.

The external data change tracking feature polls the external system at configurable intervals (5–30 minutes) for tracked changes. For each external object that has the feature enabled, a topic channel and an associate change event entity are created where change event notifications are published. You add a subscriber to each topic channel and then process the data changes through Streaming API. You can also add an Apex trigger that is called when change event notifications are published.

[External Change Data Capture Considerations](#)

Keep these considerations in mind when working with External Change Data Capture.

[Enable External Change Data Capture and Tracking](#)

Enable external change data capture for the data source and each external object that you want to monitor.

[Subscribe to Change Events](#)

You can use Apex triggers to subscribe to change events. Or you can use a Bayeux client to subscribe to Streaming API on the publication channel.

[Check the External Change Data Capture Status for an External Object](#)

You can quickly check the change-tracking status from an external object's detail page. More detailed monitoring is also available.

[Change the Polling Interval for External Change Data Capture](#)

By default, the external change data capture feature polls the external system every 30 minutes for tracked changes. You can change the interval to poll the external system as frequently as every 5 minutes.

[Monitor and Troubleshoot External Change Data Capture](#)

Check these tips when troubleshooting change tracking on external objects.

[Example: How Codey Outfitters Uses External Change Data Capture](#)

The fictitious company, Codey Outfitters, distributes outdoor equipment from local small businesses to outdoor enthusiasts around the world. The company handles all logistics, such as complying with import laws for each state and country.

External Change Data Capture Considerations

Keep these considerations in mind when working with External Change Data Capture.

- Each polling request counts toward your Salesforce org's OData callout rate allocation.
- OData doesn't differentiate between created and updated change events. Both events are published with update change type.
- When you enable change tracking, a \$Stop=0 query for zero rows is sent to the external data source to tell it to track changes, and a delta link is returned.
- If an error occurs, Salesforce stops sending polling requests. You can check for an error status by querying the BackgroundOperation object or viewing the latest error on the object's setup page.
- If you create an Apex trigger to respond to change events, these requirements and behaviors apply.
 - Create the Apex trigger with development tools, such as the Developer Console or Metadata API or Salesforce UI. You can't create the trigger from the Salesforce UI.
 - Because the Apex trigger runs after an event occurs, only the after-insert trigger event is supported.
 - The trigger passes the associated change event, not the object that caused the change.
 - Changed records can be committed in the same transaction that started the trigger. However, keep in mind that this transaction is separate from the transaction that created the original event.

Enable External Change Data Capture and Tracking

Enable external change data capture for the data source and each external object that you want to monitor.

REQUIRED EDITIONS

USER PERMISSIONS NEEDED

To create or edit external objects:	Customize Application
-------------------------------------	-----------------------

1. From Setup, enter *External Data Sources* in the Quick Find box, then select **External Data Sources**.
2. Click **Edit**, and select **Eligible for External Change Data Capture**.
3. Click **Save**.
4. From the External Objects list, select the external object that you want to track.
5. Click **Edit**, and select **Track Data Changes**.
6. Click **Save**.

After you enable change tracking, a publication channel is created under the topic for the external object. For example, `/data/Object_Name__ChangeEvent`s appears as `/data/Products__ChangeEvent`s, where *Product* is the object name.

Subscribe to Change Events

You can use Apex triggers to subscribe to change events. Or you can use a Bayeux client to subscribe to Streaming API on the publication channel.

After subscribing, you can observe change event notifications when you perform a DML operation on an external object. You can also see change events on the tracked data stored on the external system.

Check the External Change Data Capture Status for an External Object

You can quickly check the change-tracking status from an external object's detail page. More detailed monitoring is also available.

REQUIRED EDITIONS

USER PERMISSIONS NEEDED

To create or edit external objects:	Customize Application
-------------------------------------	-----------------------

1. From Setup, in the Quick Find box, enter *External Objects*, and then select **External Objects**.
2. Click the label of the external object.

The External Change Data Capture Status field shows one of the following statuses.

- Disabled—Either the data source or the external object doesn't have external change data capture enabled.
- Error—External change data capture encountered an error in the last poll for changes and stopped. After resolving the error, restart external change data capture.
- Running—External change data capture is running and polls the external system in the configured time interval for changes.
- Stopped—External change data capture isn't scheduled to poll for changes.

Change the Polling Interval for External Change Data Capture

By default, the external change data capture feature polls the external system every 30 minutes for tracked changes. You can change the interval to poll the external system as frequently as every 5 minutes.

1. From Setup, enter *External Data Sources* in the Quick Find box, then select **External Data Sources**.
2. Next to the name of the external data source, click **Edit**.
3. In the Polling Interval for External Change Data Capture field, enter the number of seconds between polling.
4. Click **Save**.

Monitor and Troubleshoot External Change Data Capture

Check these tips when troubleshooting change tracking on external objects.

- To check the status of external data change tracking, query the `BackgroundOperation` object.
- Each `BackgroundOperation` record is available for inspection for one day. You can include the `ExpiresAt` field in your SOQL query of the `BackgroundOperation` object to check when the record expires.

- Set up and view debug logs.

Example: How Codey Outfitters Uses External Change Data Capture

The fictitious company, Codey Outfitters, distributes outdoor equipment from local small businesses to outdoor enthusiasts around the world. The company handles all logistics, such as complying with import laws for each state and country.

Over the years, Codey Outfitters has acquired different IT systems to manage orders, shipments, inventory, and billing. It uses Salesforce CRM to manage its customer base and the suppliers whose items it sells. Codey Outfitters wants to provide customers with a one-stop shopping portal where they can see price changes on the gear.

When the inventory is updated, Codey Outfitters wants the change notifications sent to its Salesforce org. Salesforce then performs business logic to determine whether the updates include price changes that are valuable to customers based on their order history. Customers can see or be notified about relevant deals on their Codey Outfitters portal.

Let's see how Codey Outfitters sets up and uses external change data capture to accomplish these goals.

Set Up the External Data Source and the External Object

Codey Outfitters has an OData 4.0 service for its inventory. The company connects its Salesforce org to the inventory by creating an external data source called Gear Inventory and creating an external object called Products.

Monitor External Change Data Capture

With external change data capture enabled, Codey Outfitters' org polls the external system every 30 minutes. To monitor the tracking status, it queries the BackgroundOperation object. Codey Outfitters prefers to run SOQL queries in the Query Editor of the Developer Console, but you can use your preferred tool.

Subscribe to Changes with Streaming API

When Codey Outfitters runs its Bayeux client, the client subscribes to Streaming API on the products publication channel.

React to Changes with Apex Triggers

To automate your business with external change events, use Apex triggers.

Set Up the External Data Source and the External Object

Codey Outfitters has an OData 4.0 service for its inventory. The company connects its Salesforce org to the inventory by creating an external data source called Gear Inventory and creating an external object called Products.

The Products external object provides the following data.

- Name (product name)
- UnitPrice (current price)
- Stock (number of items in stock)

- OrderLimit (limit for each order)

To track changes, Codey Outfitters selects **Eligible for External Change Data Capture** for the Gear Inventory external data source. It then enables **Track Data Changes** on the Products external object.

Monitor External Change Data Capture

With external change data capture enabled, Codey Outfitters' org polls the external system every 30 minutes. To monitor the tracking status, it queries the BackgroundOperation object. Codey Outfitters prefers to run SOQL queries in the Query Editor of the Developer Console, but you can use your preferred tool.

Here's the SOQL query that Codey Outfitters uses to monitor the tracking status.

```
SELECT Id, Name, SubmittedAt, StartedAt, FinishedAt, ProcessAfter, Status
FROM BackgroundOperation WHERE WorkerUri='v45.0/xds/data-changes'
```

In the query results, the first row shows the initial callout to the external system. The initial request tells the OData producer to start tracking changes on the Products__x object. In the second row, the org schedules the next callout to request the tracked changes. Notice that the callout is scheduled to occur after 30 minutes have passed.

ID	Name	Submitted	Started	Finished	Process After	Status
08Pxx0000000019EAA	Products__x	2017-10-06T01:11:04.000Z	2017-10-06T01:11:08.000Z	2017-10-06T01:11:08.000Z		Complete
08Pxx000000000qEAA	Products__x	2017-10-06T01:11:08.000Z			2017-10-06T01:41:08.000Z	Waiting

After waiting 30 minutes, querying the BackgroundOperation object shows that the second callout finishes, and the status is Complete. In the third row, a new change request is scheduled to run after 30 more minutes have passed.

ID	Name	Submitted	Started	Finished	Process After	Status
08Pxx0000000019EAA	Products__x	2017-10-06T01:11:04.000Z	2017-10-06T01:11:08.000Z	2017-10-06T01:11:08.000Z		Complete
08Pxx000000000qEAA	Products__x	2017-10-06T01:41:09.000Z	2017-10-06T01:41:09.000Z	2017-10-06T01:41:09.000Z	2017-10-06T01:41:08.000Z	Complete
08Pxx0000000002MEAQ	Products__x	2017-10-06T01:41:09.000Z			2017-10-06T02:11:09.000Z	Waiting

Suppose that the Codey Outfitters inventory has a service outage, preventing the scheduled poll request from succeeding. Errors cause change tracking to stop, so querying the BackgroundOperation object

shows the error status and no new rows.

ID	Name	Submitted	Started	Finished	Process After	Status
08Pxx000000 0019EAA	Products __x	2017-10-06T0 1:11:04.000Z	2017-10-06T0 1:11:08.000Z	2017-10-06T0 1:11:08.000Z		Complete
08Pxx000000 000qEAA	Products __x	2017-10-06T0 1:41:09.000Z	2017-10-06T0 1:41:09.000Z	2017-10-06T0 1:41:09.000Z	2017-10-06T0 1:41:09.000Z	Complete
08Pxx000000 002MEAQ	Products __x	2017-10-06T0 1:41:09.000Z	2017-10-06T0 2:12:07.000Z	2017-10-06T0 2:12:07.000Z	2017-10-06T0 2:11:09.000Z	Error

When query results show no rows in Waiting status, external change data capture is disabled. When the query results show an error, you can get error details by including the Error field in your SOQL query of the BackgroundOperation object. Here's how Codey Outfitters investigates the error.

```
SELECT Error FROM BackgroundOperation WHERE Id='08Pxx000000002MEAQ'
```

Because a service outage caused the problem, the error message says:

The external system is unreachable. Try again later, or contact your admin, who can verify the external data source settings and the external system's availability. Attempted to reach this URL: <https://codey-outfitters.example.com/inventory/Products>

Codey Outfitters fixes the service outage and restarts change tracking on the Products__x object.

Subscribe to Changes with Streaming API

When Codey Outfitters runs its Bayeux client, the client subscribes to Streaming API on the products publication channel.

Codey Outfitters uses a Visualforce page to show its customers updates to prices and inventory and new and discontinued outdoor gear. This sample source code for the Visualforce page is on [Github](#), and you can modify it to work with your OData 4.0 producer.

The heart of the subscription mechanism is in the _subscribeChannels method in src/js/channel/ServerConnection.js.

```
cometd.subscribe(channelName, function(event) {
    if (event && event.data && event.data.payload &&
        event.data.payload.ChangeEventHeader) {
        var header = event.data.payload.ChangeEventHeader;
        var entityName = header.entityName;
```

```
var recordId = header.recordIds[0];
var timestamp = header.commitTimestamp;
var changeType = header.changeType;

switch (changeType) {
  case "UPDATE":
    // Update logic ...
    break;
  case "CREATE":
    // Create logic
    break;
  case "DELETE":
    // Delete logic
    break;
}
}
```

The channelName for Codey Outfitters' products is /data/Products__ChangeEvent. Each change event notification that's published on the channel is in the Bayeux format and represents an updated record in the product inventory. The data.payload contains the change event data and includes a header that describes the nature of the change event. This sample event notification describes a change to a Codey Outfitters product record.

```
{
  schema="lswW55LyUCYeU7raSrmZ5A",
  payload={
    ChangeEventHeader={
      commitUser="005xx000001SvAn",
      sequenceNumber=15,
      entityName="Products__x",
      changeType="UPDATE",
      changeOrigin="Products?$deltaToken=3779",
      transactionKey="f158f145-a...",
      commitTimestamp=1507759119826,
      recordIds=["x03xx0000000008"]
    },
    Name__c="Camping Stove",
    ExternalId=7801,
    UnitPrice__c=10.0,
    Stock__c=1500.0,
    OrderLimit__c=12.0,
    ProductId__c=7801.0
  },
}
```

```
event={replayId=15}
}
```

Each change event notification has this structure.

Field	Description
schema	Versioned schema reference that defines the change event.
payload	<p>Details about the change event.</p> <ul style="list-style-type: none"> ChangeEventHeader—Describes the nature of the data change event. Record fields and values—The OData producer can also include fields that have no changed values. To see the current values, query the external object using the record ID in the change event header.
event	The replay ID that refers to the position of the event notification in the event stream on a channel.

Each change event header contains these fields.

Field	Description
commitUser	User who publishes the event.
sequenceNumber	Change event sequence number. Each change occurs in the order in which it's received.
entityName	API name of the external object.
changeType	Specifies whether a record was updated, created, or deleted.
changeOrigin	Delta link that's used to request the change. The OData producer calculates the delta token while executing the previous change request.
transactionKey	Unique identifier for the transaction.
commitTimestamp	Specifies when the change event notification is published in milliseconds since January 1, 1970.
recordIds	Salesforce ID of the external object record that's

Field	Description
	updated, created, or deleted. To see the current values of the record's fields, query the external object using this record ID.
nulledFields	List of fields set to null.

React to Changes with Apex Triggers

To automate your business with external change events, use Apex triggers.

Codey Outfitters can customize Salesforce to send email or SMS notifications on items that are low on stock or have been updated to an attractive price. Gear and other items that are in high demand can be monitored for low stock and automatically ordered. Before customizing Salesforce, make sure that you're tracking data changes on the external object that you want to track. To track items that are running low on stock, Codey Outfitters creates an Apex trigger named LowOnStock. Next, the company selects the trigger's sObject and the change event that publishes changes for the external object Products__x: Products__ChangeEvent.

Here's an example of a simple rule that sends a notification when an item is low on stock.

```
trigger LowOnStock on Products__ChangeEvent (after insert) {
    // Get the change event's product ID for added and updated products
    Set<Id> productIds = new Set<Id>();
    for (Products__ChangeEvent event: Trigger.new) {
        if (event.ChangeEventHeader.getChangeType() != 'DELETE') {
            String productId = event.ChangeEventHeader.getRecordIds()[0];
            productIds.add(productId);
        }
    }
    if (productIds.size() > 0) {
        ProductNotifications.notifyOnLowOnStockProductUpdates(productIds);
    }
}
```

The product notifications class queries the external Product objects for the given changed product IDs and LowOnStock threshold value. Looking up the current Product records on the external system for qualifying products is a callout and is scheduled as an Apex future.

```
public class ProductNotifications {
    // Notify subscribers on product updates that are low on stock.
    // This method is run asynchronously as it is performing a callout to the
    external
```

```

// system to get the latest product details
@future (callout=true)
public static void notifyOnLowOnStockProductUpdates(Set<Id> productIds) {
    // Look up the current stock threshold to filter the products with
    Integer productStockThreshold = ...;
    List<Products__x> lowOnStockProducts =
        getProductsLowOnStock(productIds, productStockThreshold);
    // Notify subscribers
    notifySubscribers(lowOnStockProducts,
        'These items are running out of stock. Act soon!');
}

// Get the external Product objects for each Product ID
public static List<Products__x>
getProductsLowOnStock(Set<Id> productIds, Integer productStockThreshold) {
    return [SELECT Id, ExternalId, Name__c, ProductId__c, Stock__c, UnitPrice__c
            FROM Products__x
            WHERE Stock__c < :productStockThreshold AND Stock__c > 0 AND
                  Id IN :productIds];
}

// Notify subscribers by sending an email or SMS
public static void notifySubscribers(String message, List<Products__x> products) {
    ...
}
}

```

You can discover the metadata for change events through the sObject describe API. For example, the sObject describe for Products__ChangeEvent shows the following standard change event fields.

Field	Type	Description
Id	String	The change event's Salesforce ID. This ID isn't the Salesforce ID of the record that changed, but the event that carries the record's changes.
ReplayId	String	The position of the event notification in the event stream on a channel.
ChangeEventHeader	Complex Type	The type of data change event.

Each change event object defines fields that hold the updated value at the time the record was updated. The OData producer can also include fields that haven't changed. To see the current values, query the external object using the record ID in the change event header.

The following table shows sample Products__x fields. Event change objects are tailored to their tracked external object. For a change event, not all fields are necessarily present.

Field	Type	Description
Id	String	Product record's Salesforce ID in your org
ExternalId	String	Product record's external ID
DisplayUrl	String	URL representing the record in the external system
CreatedOn__c	DateTime	When the record was created in the external system
UpdatedOn__c	DateTime	When the record was last updated on the external system
Name__c	String	Product name in Product__x
OrderLimit__c	Integer	Number of items that can be ordered at a time
Stock__c	Integer	Number of items in stock
UnitPrice__c	Decimal(8,2)	Item's unit price

The following fields are change event header complex type fields. To access ChangeEventHeader fields from Apex, use its getter method. For example, to access the field ChangeType, use

```
eventObject.ChangeEventHeader.getChangeType()
```

Field	Type	Description
commitUser	String	User who published the event.
sequenceNumber	Integer	Change event sequence number. Each change is listed in the order in which it's received.
entityName	String	API name of the external object.
changeType	ChangeType enum	Specifies whether a record was updated, created, or deleted. The enum values for external change events are Create , Update , and Delete .

Field	Type	Description
changeOrigin	String	Delta link that's used to request the changes made since the previous or initial change request. The OData producer calculates the delta token while executing the previous change request.
transactionKey	String	Unique identifier for the transaction.
commitTimestamp	Long	Specifies when the change event notification is published in milliseconds since January 1, 1970.
recordIds	Array of type String	Salesforce ID of the external object record that is updated, created, or deleted. To see the current values of the record fields, query the external object using this record ID. This array always has exactly one Salesforce ID.
nulledFields	Array of type String	List of field names set to null.

Access External Data with a Custom Adapter for Salesforce Connect

Connect your users to any data anywhere by developing your own custom adapter with the Apex Connector Framework.

[Custom Adapter for Salesforce Connect](#)

Connect to any data anywhere for a complete view of your business. Use the Apex Connector Framework to develop a custom adapter for Salesforce Connect.

[Set Up Salesforce Connect to Access External Data with a Custom Adapter](#)

Let users view, search, and modify any data anywhere from within their Salesforce org.

[Considerations for Salesforce Connect—Custom Adapter](#)

Understand the special behaviors, limits, and recommendations for using a Salesforce Connect custom adapter built with the Apex Connector Framework.

Custom Adapter for Salesforce Connect

Connect to any data anywhere for a complete view of your business. Use the Apex Connector Framework

to develop a custom adapter for Salesforce Connect.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Your users and the Lightning Platform interact with the external data via external objects. For each of those interactions, Salesforce Connect invokes methods in the Apex classes that compose the custom adapter. Salesforce invokes the custom adapter's Apex code each time that:

- A user clicks an external object tab for a list view.
- A user views a record detail page of an external object.
- A user views a record detail page of a parent object that displays a related list of child external object records.
- A user performs a Salesforce global search.
- A user creates, edits, or deletes an external object record.
- A user runs a report.
- The preview loads in the report builder.
- An external object is accessed via flows, processes, APIs, Apex, SOQL, or SOSL.
- You validate or sync an external data source.

Apex code can access external objects, but some limitations and requirements apply.

- To understand the limitations on Apex code accessing external objects, see [Apex Considerations for Salesforce Connect External Objects](#).
- To know how to use a custom adapter for Salesforce Connect, see [Get Started with the Apex Connector Framework](#).

[External IDs for External Objects in Salesforce Connect–Custom Adapter](#)

When you access external data with a custom adapter for Salesforce Connect, the values of the External ID standard field on an external object come from the `DataSource.Column` named `ExternalId`.

See Also

[Salesforce Connect](#)

[Set Up Salesforce Connect to Access External Data with a Custom Adapter](#)

[Considerations for Salesforce Connect–Custom Adapter](#)

External IDs for External Objects in Salesforce Connect–Custom Adapter

When you access external data with a custom adapter for Salesforce Connect, the values of the External

ID standard field on an external object come from the `DataSource.Column` named `ExternalId`.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Each external object has an **External ID** standard field. Its values uniquely identify each external object record in your org. When the external object is the parent in an external lookup relationship, the External ID standard field is used to identify the child records.

Important

- The custom adapter's Apex code must declare the `DataSource.Column` named `ExternalId` and provide its values.
- Don't use sensitive data as the values of the External ID standard field or fields designated as name fields, because Salesforce sometimes stores those values.
 - External lookup relationship fields on child records store and display the External ID values of the parent records.
 - For internal use only, Salesforce stores the External ID value of each row that's retrieved from the external system. This behavior doesn't apply to external objects that are associated with high-data-volume external data sources.



Example This excerpt from a sample `DataSource.Connection` class shows the `DataSource.Column` named `ExternalId`.

```
override global List<DataSource.Table> sync() {  
    List<DataSource.Table> tables =  
        new List<DataSource.Table>();  
    List<DataSource.Column> columns;  
    columns = new List<DataSource.Column>();  
    columns.add(DataSource.Column.text('title', 255));  
    columns.add(DataSource.Column.text('description', 255));  
    columns.add(DataSource.Column.text('createdDate', 255));  
    columns.add(DataSource.Column.text('modifiedDate', 255));  
    columns.add(DataSource.Column.url('selfLink'));  
    columns.add(DataSource.Column.url('DisplayUrl'));  
    columns.add(DataSource.Column.text('ExternalId', 255));  
    tables.add(DataSource.Table.get('googleDrive', 'title',
```

```
    columns));
    return tables;
}
```

See Also

[Custom Adapter for Salesforce Connect](#)
[Apex Developer Guide](#)

Set Up Salesforce Connect to Access External Data with a Custom Adapter

Let users view, search, and modify any data anywhere from within their Salesforce org.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create Apex classes:	Author Apex
To configure remote settings:	Modify All Data
To create and edit external data sources:	Customize Application
To create and edit external objects:	Customize Application
To define or change object-level help:	Customize Application
To create and edit custom fields:	Customize Application
To edit permission sets and user profiles:	Manage Profiles and Permission Sets
To edit another user's authentication settings for external systems:	Manage Users

Setting up Salesforce Connect with a custom adapter involves these high-level steps.

1. [Develop the custom adapter for Salesforce Connect.](#)

Using the Apex Connector Framework, create the `DataSource.Connection` and `DataSource.Provider` classes that comprise the custom adapter.

2. [Define remote sites for Apex callouts.](#)

If the custom adapter involves any Apex callouts, define each callout endpoint as a remote site in your

organization. However, you don't need to define a remote site for a callout whose endpoint is specified as a [named credential](#) instead of a URL.

3. [Define an external data source of type `Salesforce Connect: Custom`.](#)

If you created multiple custom adapters, make sure that the external data source's **Type** field specifies the correct `DataSource.Provider` class.

4. [Create the external objects.](#)

Perform this task only if you don't sync to automatically create the external objects. Create an external object for each external data table that you want to access from your Salesforce org.

5. [Create help content for the external objects.](#)

Create Visualforce pages that describe the external data. When your users click **Help for this Page** on an external object, they read your custom help content. Remember, your users can't find information about the external data in Salesforce Help.

6. [Add custom fields and relationships to the external objects.](#)

Create relationships between objects. If you didn't sync to automatically create the external objects and their fields, create a custom field for each external table column that you want to access from your Salesforce org.

7. [Verify access to external object data.](#)

Check that expected user and code interactions with the external objects work, including sorting and filtering search and query results.

 **Tip** After you configure an external data source, run the validator tool on each external object to test and troubleshoot its connections. The tool tests for ID uniqueness and the ability to sort and filter results.

8. [Enable user access to external objects.](#)

Grant object permissions through permission sets or profiles.

9. [Enable user access to the fields on the external objects.](#)

Grant field permissions through permission sets or profiles.

10. If the external data source uses per-user authentication:

a. [Let users authenticate to the external system.](#)

Grant users access to authentication settings for the external data source through permission sets or profiles.

b. [Set up each user's authentication settings.](#)

You or your users can perform this task.

 **Tip** Train your users on how to set up their authentication settings for external systems. Make sure that they know which credentials to enter for each external system. If you're using OAuth 2.0, test the OAuth flow for potentially confusing prompts or redirects, and train your users as needed. OAuth flows vary, depending on your external system, authentication provider, and specified scopes.

See Also

[Custom Adapter for Salesforce Connect](#)

[Salesforce Platform Features Supported by Salesforce Connect](#)

[Developer Guide: *Visualforce Developer's Guide*](#)

[External Object Relationships](#)

Define an External Data Source for Salesforce Connect–Custom Adapter

Connect your Salesforce org to any data anywhere via a Salesforce Connect custom adapter that you create with the Apex Connector Framework.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit external data sources: Customize Application

Before you begin, develop the custom adapter for Salesforce Connect. If the custom adapter uses Apex callouts, also define remote sites for the callout endpoints. See [Set Up Salesforce Connect to Access External Data with a Custom Adapter](#).

1. From Setup, enter *External Data Sources* in the **Quick Find** box, then select **External Data Sources**.
2. Click **New External Data Source**, or click **Edit** to modify an existing external data source.
3. Complete the fields.

Field	Description
Label	A user-friendly name for the external data source. The label is displayed in the Salesforce user interface, such as in list views. If you set Identity Type to Per User, this label appears when your users view or edit their authentication settings for external systems.
Name	A unique identifier that's used to refer to this external data source definition through the API. The name can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
Type	Select Salesforce Connect: Custom-<your DataSource.Provider class name> . If you change and save a DataSource.Connection class, resave the

Field	Description
	<p>corresponding <code>DataSource.Provider</code> class. Otherwise, when you define the external data source, the custom adapter doesn't appear as an option for the Type field.</p>
URL	<p>URL of the external system.</p> <p>Available only when the <code>DataSource.Provider</code> class declares the <code>REQUIRE_ENDPOINT</code> capability. If the class also declares the <code>REQUIRE_HTTPS</code> capability, the URL must begin with <code>https://</code>.</p> <p>If the endpoint is defined in a named credential, enter the named credential URL. A named credential URL contains the scheme <code>callout:</code>, the name of the named credential, and an optional path. For example: <code>callout: My_Named_Credential / some_path</code>.</p> <p>You can append a query string to a named credential URL. Use a question mark (?) as the separator between the named credential URL and the query string. For example: <code>callout: My_Named_Credential / some_path ?format=json</code>.</p>
Writable External Objects	<p>The Lightning Platform and users in this org can create, update, and delete records for external objects associated with the external data source. The external object data is stored outside the org. By default, external objects are read only.</p>
High Data Volume	<p>Salesforce enforces rate limits for retrieving and viewing data from external systems. If your org hits rate limits when accessing external objects, consider selecting the High Data Volume option on the associated external data sources. Doing so bypasses most rate limits, but some special behaviors and limitations apply. See High Data Volume Considerations for Salesforce Connect–Custom Adapters.</p>
Certificate	<p>If you specify a certificate, your Salesforce org supplies it when establishing each two-way SSL connection with the external system. The certificate is used for digital signatures, which verify that requests are coming from your Salesforce org.</p> <p>Available only when the <code>DataSource.Provider</code> class declares the <code>CERTIFICATE</code> authentication capability.</p>
Identity Type	<p>Determines whether you're using one set or multiple sets of credentials to access the external system. See Identity Type for External Data Sources.</p>

Field	Description
	Available only when the <code>DataSource.Provider</code> class declares the <code>BASIC</code> or <code>OAUT</code> authentication capability.

4. Select the authentication protocol.

- If you select **Password Authentication**, enter the username and password for accessing the external system.
Password authentication is available only when the `DataSource.Provider` class declares the `BASIC` authentication capability.
- If you select **OAuth 2.0**, complete the following fields.
OAuth 2.0 is available only when the `DataSource.Provider` class declares the `OAUT` authentication capability.

Field	Description
Authentication Provider	Choose the provider. See Authentication Providers .
Scope	<p>Specifies the scope of permissions to request for the access token. Your authentication provider determines the allowed values. See Use the Scope Parameter.</p> <p>Keep these considerations in mind when you set a scope.</p> <ul style="list-style-type: none"> The value that you enter replaces the Default Scopes value that's defined in the specified authentication provider. Whether scopes are defined can affect whether each OAuth flow prompts the user with a consent screen. We recommend that you request a refresh token or offline access. Otherwise, when the token expires, you lose access to the external system.
Start Authentication Flow on Save	<p>To authenticate to the external system and obtain an OAuth token, select this checkbox. This authentication process is called an OAuth flow.</p> <p>When you click Save, the external system prompts you to log in. After successful login, the external system grants you an OAuth token for accessing its data from this org.</p> <p>Redo the OAuth flow when you need a new token—for example, if the token expires—or if you edit the Scope or Authentication Provider fields. When the token expires, the external system returns a 401 HTTP error status.</p>

5. Click **Save**.

6. Click **Validate and Sync**, and confirm that the connection is successful.

This step also invokes the `sync()` method on the `DataSource.Connection` class to obtain the list of tables that you can sync to create external objects and their fields.

7. Optionally, select tables and click **Sync** to do the following for each selected table.

- Automatically create a Salesforce external object.
- Automatically create a custom field for each table column that's compatible with a Salesforce metadata field type.



Note Before you sync, make sure that you understand the considerations that are described in these topics.

- [Sync an External Data Source for Salesforce Connect](#)
- [Sync Considerations for Salesforce Connect–Custom Adapter](#)

You can instead choose to manually create the external objects and custom fields that map to the external data. Doing so lets you customize the external object names, decide which table columns to create custom fields for, and customize the custom field names. However, this approach takes longer and requires manual maintenance.

See Also

[Set Up Salesforce Connect to Access External Data with a Custom Adapter](#)

[Store Authentication Settings for External Systems](#)

[Named Credentials](#)

Considerations for Salesforce Connect–Custom Adapter

Understand the special behaviors, limits, and recommendations for using a Salesforce Connect custom adapter built with the Apex Connector Framework.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

[Apex Connector Framework Considerations for Salesforce Connect–Custom Adapter](#)

Understand the limits and considerations for creating a Salesforce Connect custom adapter with the Apex Connector Framework.

[High Data Volume Considerations for Salesforce Connect–Custom Adapters](#)

If your org hits rate limits when accessing external objects, consider selecting the **High Data Volume** option on the associated external data sources. Doing so bypasses most rate limits, but some special behaviors and limitations apply.

[Writable External Objects Considerations for Salesforce Connect–Custom Adapters](#)

Some special behaviors and limitations affect writable external objects that are associated with

custom adapters for Salesforce Connect.

Sync Considerations for Salesforce Connect–Custom Adapter

When you validate and sync an external data source of type `Salesforce Connect: Custom`, some special behaviors and limitations apply.

See Also

[Salesforce Platform Features Supported by Salesforce Connect](#)

Apex Connector Framework Considerations for Salesforce Connect–Custom Adapter

Understand the limits and considerations for creating a Salesforce Connect custom adapter with the Apex Connector Framework.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- If you change and save a `DataSource.Connection` class, resave the corresponding `DataSource.Provider` class. Otherwise, when you define the external data source, the custom adapter doesn't appear as an option for the **Type** field. Also, the associated external objects' custom tabs no longer appear in the Salesforce UI.
- DML operations aren't allowed in the Apex code that comprises the custom adapter.
- Apex methods with the `future` annotation aren't supported in an Apex custom adapter. When a future method is invoked, the external callout could cause an error in Salesforce.
- Make sure that you understand the limits of the external system's APIs. For example, some external systems accept only requests for up to 40 rows.
- Apex data type limitations:
 - Double—The value loses precision beyond 18 significant digits. For higher precision, use decimals instead of doubles.
 - String—if the length is greater than 255 characters, the string is mapped to a long text area field in Salesforce.
- Custom adapters for Salesforce Connect are subject to the same limitations as any other Apex code. For example:
 - All Apex governor limits apply.
 - Test methods don't support web service callouts. Tests that perform web service callouts fail. For an example that shows how to avoid these failing tests by returning mock responses, see [Google Drive™ Custom Adapter for Salesforce Connect](#).
- In Apex tests, use dynamic SOQL to query external objects. Tests that perform static SOQL queries of external objects fail.

See Also

- [Custom Adapter for Salesforce Connect](#)
- [External Objects in Salesforce Connect](#)
- [Apex Developer Guide: Primitive Data Types](#)
- [Apex Developer Guide: Execution Governors and Limits](#)
- [Apex Developer Guide: Callout Limits and Limitations](#)
- [Apex Developer Guide: Google Drive™ Custom Adapter for Salesforce Connect](#)
- [Apex Developer Guide: Dynamic SOQL](#)

High Data Volume Considerations for Salesforce Connect–Custom Adapters

If your org hits rate limits when accessing external objects, consider selecting the **High Data Volume** option on the associated external data sources. Doing so bypasses most rate limits, but some special behaviors and limitations apply.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- The following features aren't available for external objects that are associated with high-data-volume external data sources.
 - Access via Lightning Experience
 - Access via the Salesforce mobile app
 - Appearance in Recent Items lists
 - Record feeds
 - Reports and dashboards
 - Writable external objects
- Salesforce IDs aren't assigned to external object records that are associated with high-data-volume external data sources.
- On record detail pages for external objects that are associated with high-data-volume external data sources, custom buttons, and links that call JavaScript aren't supported.
- Salesforce Console in Salesforce Classic doesn't support external objects that are associated with high-data-volume external data sources.
- CSRF protection for writable external objects isn't available for high-data-volume external data sources.

See Also

- [Custom Adapter for Salesforce Connect](#)
- [Considerations for Salesforce Connect–Custom Adapter](#)

Writable External Objects Considerations for Salesforce Connect–Custom Adapters

Some special behaviors and limitations affect writable external objects that are associated with custom adapters for Salesforce Connect.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- Writable external objects aren't available for high-data-volume external data sources.
- Queued changes to the external data execute over time, so external object records that are read successively can contain different data.

Also review the considerations that apply to all Salesforce Connect adapters.

See Also

[Writable External Objects in Salesforce Connect](#)

[Custom Adapter for Salesforce Connect](#)

[Apex Developer Guide : Writable External Objects](#)

Sync Considerations for Salesforce Connect–Custom Adapter

When you validate and sync an external data source of type `Salesforce Connect: Custom`, some special behaviors and limitations apply.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- Syncing always enables search on the external object when search is enabled on the external data source, and vice versa. For an external data source of type `Salesforce Connect: Custom`, search is enabled when the custom adapter's `DataSource.Provider` class declares the `DataSource.Capability.SEARCH` capability.

! **Important** When you select the tables to sync, determine whether check marks appear in the

Synced column. If a Synced check mark appears, your org has an external object whose object name matches the table name. If you select the table and click **Sync**:

- The external object is overwritten.
- Any custom field on the external object is overwritten if its API name (for example, `Email__c`) associates it with a table column name (for example, `Email`).
- Any other custom fields on the external object remain as they are, including:
 - Previously synced custom fields whose API names were changed by editing their **Field Name** values.
 - Manually added custom fields whose API names aren't associated with table column names.

If no Synced check mark appears, and you sync the table, a new external object is created in your org. The new external object's object name matches the table name. For example, suppose you change the table name in the custom adapter's `DataSource.Connection` class to no longer match the object name of the external object. Syncing that table creates a new external object in Salesforce. We recommend that you change the object name of the existing external object to match the new table name in the `DataSource.Connection` class before you sync that table.

Also review the considerations that apply to all Salesforce Connect adapters.

See Also

- [Sync an External Data Source for Salesforce Connect](#)
- [Define an External Data Source for Salesforce Connect—Custom Adapter](#)
- [Custom Adapter for Salesforce Connect](#)

Access External Data with the Salesforce Connect Adapter for Amazon DynamoDB

Connect your users to data that's stored in Amazon DynamoDB.

[Salesforce Connect Adapter for Amazon DynamoDB](#)

Connect to Amazon DynamoDB to access and integrate DynamoDB data into Salesforce applications. With this adapter, take advantage of the flexible data storage capabilities of Amazon DynamoDB that provides fast and predictable performance combined with customer 360-degree view captured in Salesforce.

[Set Up Salesforce Connect to Access External Data in Amazon DynamoDB](#)

Provide users with access to data stored in DynamoDB data source in Salesforce so that they have a complete view of the business.

[Manage Qualifiers for Salesforce Connect Adapter for Amazon DynamoDB](#)

Amazon DynamoDB uses unique design patterns to model data efficiently that support multiple table and single table designs. To make data stored in Amazon DynamoDB table accessible to external objects, use qualifiers to reinterpret external data. The qualifiers describe the item attributes and indicates whether an item belongs to the schema represented by the external object. For multiple-table schema designs, the table qualifier is optional.

Considerations for Salesforce Connect Adapters for Amazon DynamoDB

Understand the special behaviors, limits, and recommendations for using the Salesforce Connect adapter for Amazon DynamoDB.

Salesforce Connect Adapter for Amazon DynamoDB

Connect to Amazon DynamoDB to access and integrate DynamoDB data into Salesforce applications. With this adapter, take advantage of the flexible data storage capabilities of Amazon DynamoDB that provides fast and predictable performance combined with customer 360-degree view captured in Salesforce.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Amazon DynamoDB achieves its goals of consistent fast performance at very high scale through a series of trade-offs different from those of an RDBMS. A sophisticated partitioning scheme manages the storage of very large tables, though at that size JOINs become impractical and aren't supported. Querying is limited in other ways as well. To understand more, see [Amazon DynamoDB Developer Guide: Modeling Relational Data in DynamoDB](#).

Salesforce Connect Adapter for Amazon DynamoDB maps single table and multiple table designs of DynamoDB as external objects on the Salesforce Platform. You can define external object relationships to integrate DynamoDB data into your Salesforce org, based on your business requirements.

Salesforce Connect adapter for Amazon DynamoDB uses PartiQL to perform data manipulation operations, such as SELECT or UPDATE on the external objects. To know about how the Amazon DynamoDB source tables are manipulated, see [Amazon DynamoDB Developer Guide: PartiQL - A SQL-Compatible Query Language for Amazon DynamoDB](#).

[Map Salesforce External Object Field to DynamoDB Attributes](#)

External object fields of Salesforce Connect Adapter for Amazon DynamoDB can be mapped to attributes of string, number, and boolean scalar types. If an attribute of string type that represents a date or a timestamp is mapped to an external object field, then date, dateTIme, or time must be stored in ISO 8601 format.

See Also

[Salesforce Connect](#)

[Set Up Salesforce Connect to Access External Data in Amazon DynamoDB](#)

[Manage Qualifiers for Salesforce Connect Adapter for Amazon DynamoDB](#)

[Considerations for Salesforce Connect Adapters for Amazon DynamoDB](#)

Map Salesforce External Object Field to DynamoDB Attributes

External object fields of Salesforce Connect Adapter for Amazon DynamoDB can be mapped to attributes of string, number, and boolean scalar types. If an attribute of string type that represents a date or a timestamp is mapped to an external object field, then date, dateTIme, or time must be stored in ISO 8601 format.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

This table lists Amazon DynamoDB attribute types supported for field types of external objects in Salesforce.

Amazon DynamoDB Shorthand Data Type	Amazon DynamoDB Data Type	Salesforce External Object Field Type
S	String	<p>Text</p> <p>Mapped as is.</p>
N	Number	<p>Number</p> <p>Mapped to default precision and scale, which you can override.</p>
BOOL	Boolean	<p>Checkbox</p> <p>Mapped as is.</p>

Following is the mapping strategy of Amazon DynamoDB attribute types to some of the common field types of external objects in Salesforce:

- An attribute of string type that represents a date or dateTIme is stored in ISO 8601 format.
- An attribute of string type that represents a time is stored in a format used by TIMEVALUE() or TEXT() function.
- An attribute of number type that represents currency is stored as a numeric value using the currency symbol configured for the Salesforce org.
- An attribute of string type that represents a picklist is rendered as Salesforce defined enumeration

- values. For picklist (multi-select), it is stored as comma-separated values of the selected choices.
- An attribute of number type that represents a percent is stored as a numeric value of percent type.
 - An attribute of string type that represents a phone, URL, textarea, or email is stored as the corresponding Salesforce field type.

External object fields cannot be mapped to these attributes types:

- Scalar Types: Binary, Null
- Document Types: Map, List
- Sets of number, string, or binary values

 **Note** When you manually create fields for an external object, make sure the mapped attributes are of a compatible type.

See Also

- [Amazon DynamoDB Developer Guide: Data Types](#)
- [Custom Field Types](#)
- [Salesforce Connect Adapter for Amazon DynamoDB](#)

Set Up Salesforce Connect to Access External Data in Amazon DynamoDB

Provide users with access to data stored in DynamoDB data source in Salesforce so that they have a complete view of the business.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit external data sources:	Customize Application
To create and edit external objects:	Customize Application
To view named credentials:	View Setup and Configuration
To create, edit, or delete named credentials:	Customize Application
To create and edit custom fields:	Customize Application
To edit permission sets and user profiles:	Manage Profiles and Permission Sets
To edit another user's authentication settings for external systems:	Manage Users

Before you begin, review Amazon DynamoDB setup and access permissions in [Amazon DynamoDB Developer Guide](#).

- For information about setup and access instructions, see [Setting Up DynamoDB and Accessing DynamoDB](#).
- To know about identity-based AWS Identity and Access Management (IAM) policies with Amazon DynamoDB, see [Managing Access](#).

The AWS user that the Salesforce Connect adapter for Amazon DynamoDB uses to make API calls must have the required permissions to access the DynamoDB data source. You can use [AWS Managed \(Predefined\) IAM Policies](#) to allow read-only or edit access to users. To be able to use PartiQL statements, you must add [IAM Security Policies with PartiQL for DynamoDB](#) to user's access permissions.

[Configure an Amazon DynamoDB External Data Source with Setup Wizard](#)

Define an external data source for an Amazon DynamoDB with fewer clicks using the setup wizard. The guided interface walks you through connecting an Amazon DynamoDB data source and integrating the data in Salesforce.

[Define a Named Credential for Salesforce Connect Adapter for Amazon DynamoDB](#)

Create a named credential that specifies the endpoint URL for Amazon DynamoDB and an external credential to provide the required authentication parameters.

[Define a Legacy Named Credential for Salesforce Connect Adapter for Amazon DynamoDB](#)

Create a legacy named credential that specifies the URL of a callout endpoint as Amazon DynamoDB endpoint and provides the required authentication parameters.

[Create an External Data Source for Salesforce Connect Adapter for Amazon DynamoDB](#)

Connect your Salesforce org to data that's stored in Amazon DynamoDB.

[Create External Objects for Salesforce Connect Adapter for Amazon DynamoDB](#)

Tables in Amazon DynamoDB map to one or more external objects in Salesforce, combining all your data and content for users in your org. The external objects associated with Amazon DynamoDB data source are searchable objects. Use the search box at the top of every page to search by a specific external object or for a global search across Salesforce. You can also use other search tools available in the Salesforce Platform to find external object records mapped to an Amazon DynamoDB table.

[Sync an External Data Source for Salesforce Connect Adapter for Amazon DynamoDB](#)

After you create the external objects, sync the corresponding external data source to get the metadata, such as the primary key schema and indexes on the DynamoDB table.

See Also

[Salesforce Connect](#)

[Salesforce Connect Adapter for Amazon DynamoDB](#)

[Manage Qualifiers for Salesforce Connect Adapter for Amazon DynamoDB](#)

[Considerations for Salesforce Connect Adapters for Amazon DynamoDB](#)

[Configure an Amazon DynamoDB External Data Source with Setup Wizard](#)

Define an external data source for an Amazon DynamoDB with fewer clicks using the setup wizard. The

guided interface walks you through connecting an Amazon DynamoDB data source and integrating the data in Salesforce.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit external data sources: Customize Application

1. From Setup, enter *External Data Sources* in the Quick Find box, then select **External Data Sources**.
2. Click **Connect to Amazon DynamoDB** to launch the wizard.

The wizard lets you create a new external data source or use an existing one, specify the named credential required to authenticate, select the DynamoDB table to map, and create an external object based on representative sample data. You can review the summary of the newly created external object and proceed to connect to another Amazon DynamoDB table, as required.

Define a Named Credential for Salesforce Connect Adapter for Amazon DynamoDB

Create a named credential that specifies the endpoint URL for Amazon DynamoDB and an external credential to provide the required authentication parameters.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To view named credentials: View Setup and Configuration

To create, edit, or delete named credentials: Customize Applications

1. From Setup, enter *Named Credentials* in the Quick Find box, then select **Named Credentials**.
2. Click **External Credentials**.
3. To create a new external credential, click **New**. To edit an existing external credential, click its link in the list of external credentials and then click **Edit**.
4. Complete the fields.

Field	Description
Label	A user-friendly name for the named credential that's displayed in the Salesforce user interface.
Name	A unique identifier that's used to refer to this external credential from callout definitions while creating an external data source for Amazon DynamoDB.
Authentication Protocol	Select AWS Signature Version 4 .
Service	The name of an AWS service, such as <i>dynamodb</i> .
Region	The AWS region for the named credential's endpoint. For example, <i>us-west-2</i> .
AWS Account ID	Optional. The 12-digit number that uniquely identifies your AWS account.
Use STS for Temporary Access	Select the checkbox to provide limited access and specify STS access key, access secret, external ID, and duration. For details, see Create and Edit an External Credential .

5. Click **Save**. You're taken to the Named Credentials screen.
6. To further edit the new external credential, click **External Credential**.
7. Select the external credential you created.
8. Scroll to Permission Set Mappings.
9. Click **New** to create a permission set mapping for this external credential.
10. Complete the following fields. When you use STS for temporary access, the Access Key and Secret fields are disabled and display the temporary credentials, if any.

Field	Description
Permission Set	<p>Select an available permission set. This enables different groups of Salesforce users to mirror access permissions of IAM roles.</p> <p>Make sure you define the required access to the User External Credentials object (use permission sets or profiles to configure object access). Only users with access to the User External Credentials object can make callouts to the external source. For details, see Named Credentials and External Credentials.</p>
Sequence Number	Assign a sequence number. A sequence number specifies the order of principals to apply when a user participates in more than one principal. For example, a user could be part of multiple permission sets that are applicable for a credential provider. Priority is from lower to higher numbers.

Field	Description
IAM Role ARN	<p>The Amazon Resource Name (ARN) of the role that the credential assumes.</p> <p>To get the ARN for an IAM role:</p> <ol style="list-style-type: none"> In the navigation pane of the IAM console, choose Roles. In the list of roles, choose the role you want to map to the permission set. In the Summary section, copy the ARN value. <p>For details, see AWS Identity and Access Management User Guide: Tutorials.</p>

11. Create a named credential that uses the authentication configuration you defined in the external credential. From Setup, enter *Named Credentials* in the Quick Find box, select **Named Credentials**, and then click **Named Credentials**.
12. To create a new named credential, click **New** and complete the fields.
 - Fill out the **Label**, **Name**, and **URL** fields.
Sample format for the URL would be `https://dynamodb.REGION.amazonaws.com` For example, if the AWS region is `us-west-2`, the URL would be `https://dynamodb.us-west-2.amazonaws.com`
For details, see [Create and Edit a Named Credential](#).
 - For the **External Credential** field, select the external credential you created that uses the AWS Signature Version 4 authentication protocol.
 - Select **Generate Authorization Header**.

Use the named credential that captures the authentication configuration in external credentials to authenticate Salesforce users against Amazon DynamoDB, provide limited access to AWS resources, and periodically refresh access tokens.

See Also

- [Amazon DynamoDB Developer Guide: Getting an AWS Access Key](#)
- [Named Credentials](#)
- [Salesforce Connect Adapter for Amazon DynamoDB](#)

Define a Legacy Named Credential for Salesforce Connect Adapter for Amazon DynamoDB

Create a legacy named credential that specifies the URL of a callout endpoint as Amazon DynamoDB endpoint and provides the required authentication parameters.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: All Editions

USER PERMISSIONS NEEDED

To view named credentials:

View Setup and Configuration

To create, edit, or delete named credentials:

Customize Applications

! **Important** In Winter '23, Salesforce introduced an improved named credential that is extensible, customizable, and more secure. We strongly recommend that you use this preferred credential instead of legacy named credentials, which are no longer updated or enhanced. For information on extensible, customizable named credentials, see [Named Credentials Schema](#).

1. From Setup, enter *Named Credentials* in the Quick Find box, then select **Named Credentials**.
2. To create a new legacy named credential, click **New Legacy** from the dropdown menu.
3. Complete the fields.

Field	Description
Label	A user-friendly name for the named credential that's displayed in the Salesforce user interface.
Name	A unique identifier that's used to refer to this named credential from callout definition while creating an external data source for Amazon DynamoDB.
URL	<p>The URL of the Amazon DynamoDB endpoint.</p> <p>Sample format for the URL would be <code>https://dynamodb.REGION.amazonaws.com</code></p> <p>For example, if the AWS region is <i>us-east-1</i>, the URL would be <code>https://dynamodb.us-east-1.amazonaws.com</code></p>
Identity Type	Select Named Principal to designate one user account on the Amazon DynamoDB external system for all your Salesforce org users.
Authentication Protocol	<p>Select AWS Signature Version 4 and complete the following fields.</p> <ol style="list-style-type: none"> a. AWS Access Key ID: First part of the access key used to sign programmatic requests to AWS. b. AWS Secret Access Key : Second part of the access key used to sign programmatic requests to AWS. c. AWS Region : The AWS region name for the named credential's endpoint. For example, <i>us-east-1</i>. d. AWS Service: <i>dynamodb</i>, which is the AWS utility to access.

See Also

- [Amazon DynamoDB Developer Guide: Getting an AWS Access Key](#)
- [Named Credentials](#)
- [Salesforce Connect Adapter for Amazon DynamoDB](#)

Create an External Data Source for Salesforce Connect Adapter for Amazon DynamoDB

Connect your Salesforce org to data that's stored in Amazon DynamoDB.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit external data sources: Customize Application

1. From Setup, enter *External Data Sources* in the Quick Find box, then select **External Data Sources**.
2. Click **New External Data Source**, or click **Edit** to modify an existing external data source.
3. Complete the fields.

Field	Description
External Data Source	A user-friendly name for the external data source. The label is displayed in the Salesforce user interface.
Name	A unique identifier that's used to refer to this external data source definition through the API.
Type	Select Amazon DyanamoDB .
Named Credential	Enter the named credential URL you defined for Amazon DynamoDB data source. You can skip the Authentication section for the external data source. To access the Amazon DynamoDB external system, Salesforce Connect uses the authentication settings that are defined in the named credential.
Connection Timeout	Number of seconds to wait for a response from the Amazon DynamoDB external system before timing out. By default, the value is set to the maximum of 120 seconds.

Field	Description
Writable External Objects	Select this option only if you want to create, edit, and delete data that's stored in Amazon DynamoDB. By default, external objects are read only.

4. Click **Save**.

See Also

- [Work with External Data Sources](#)
[Salesforce Connect Adapter for Amazon DynamoDB](#)

Create External Objects for Salesforce Connect Adapter for Amazon DynamoDB

Tables in Amazon DynamoDB map to one or more external objects in Salesforce, combining all your data and content for users in your org. The external objects associated with Amazon DynamoDB data source are searchable objects. Use the search box at the top of every page to search by a specific external object or for a global search across Salesforce. You can also use other search tools available in the Salesforce Platform to find external object records mapped to an Amazon DynamoDB table.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create or edit external objects:	Customize Application
-------------------------------------	-----------------------

For external object fields defined for an Amazon DynamoDB data source, the attribute values in DynamoDB must be appropriately mapped to the data type defined for the external object fields. See [Supported Data Types](#).

To create or modify an external object:

1. From Setup, enter *External Objects* in the Quick Find box, then select **External Objects**.
2. Click **New External Object**, or click **Edit** to modify an existing external object.
3. Enter the following:

Field	Description
Label	A user-friendly name for the external object. The label is displayed in the Salesforce user interface.

Field	Description
Plural Label	The plural name of the external object. When you create a tab for this object, this name is used for the tab.
Object Name	A unique identifier used to refer to this external object definition when using the API. Object names must be unique across all standard, custom, and external objects in the org.
External Data Source	The external data source definition that contains the connection details you want to use for this external object.
Table Name	Table in Amazon DynamoDB data source that the external object maps to.

4. Click **Save**.
5. On the external object detail page, view and modify the external object's custom fields and relationships, page layouts, field sets, search layouts, and buttons and links.
 - To create field mappings or add fields to an external object, click **New** on the Custom Fields & Relationships related list.
 - To assign different page layouts by user profile, click **Page Layout Assignments**.

After the external object and its fields are created, you may have to provide additional configuration so that Salesforce can apply a schema-like concept on top of data that's inherently schemaless. Single-table schema designs often require developers to populate attribute values with prefixes that must be parsed for end users. See [Manage Qualifiers for Salesforce Connect Adapter for Amazon DynamoDB](#).

See Also

- [Define External Objects](#)
- [External Object Relationships](#)
- [Manage Custom Objects](#)
- [Salesforce Connect Adapter for Amazon DynamoDB](#)

Sync an External Data Source for Salesforce Connect Adapter for Amazon DynamoDB

After you create the external objects, sync the corresponding external data source to get the metadata, such as the primary key schema and indexes on the DynamoDB table.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit external data sources: Customize Application

1. From Setup, enter *External Data Sources* in the Quick Find box, then select **External Data Sources**.
2. Click the name of the external data source and then click **Validate and Sync**.
3. Select the Amazon DynamoDB tables you want to sync and click **Sync**.

After syncing is complete, manually create external object fields for the synced tables.

 **Note** If the Amazon DynamoDB primary key or indexes change, you must resync the tables to get the updated metadata information.

See Also

[Amazon DynamoDB Developer Guide: Primary Key and Secondary Indexes](#)

[Salesforce Connect Adapter for Amazon DynamoDB](#)

Manage Qualifiers for Salesforce Connect Adapter for Amazon DynamoDB

Amazon DynamoDB uses unique design patterns to model data efficiently that support multiple table and single table designs. To make data stored in Amazon DynamoDB table accessible to external objects, use qualifiers to reinterpret external data. The qualifiers describe the item attributes and indicates whether an item belongs to the schema represented by the external object. For multiple-table schema designs, the table qualifier is optional.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

[Qualifier Options for Salesforce Connect Adapter for Amazon DynamoDB](#)

You can use various qualifier options to parse an Amazon DynamoDB table into an external object by applying filters to the table rows and performing lightweight transformations of the returned data.

[Qualifier Example for One-to-Many and Many-to-Many Relationships in Amazon DynamoDB](#)

An enterprise stores and retains fine-grain transaction details for millions of customers, resulting in data volumes reaching billions of records. In this example, Salesforce is the system of record for core customer information and details of all the customer's orders are stored in Amazon DynamoDB for scale reasons.

[Sort Example for Orders and Order Items in Amazon DynamoDB](#)

To effectively sort orders and order items stored in an Amazon DynamoDB table in a way that benefits

a business process, create secondary indexes that identify attribute values as the basis for sorting.

DynamoDB Qualifier Examples for Parsing Formulas

This example shows a simple formula to concatenate two text values. The value of the `location` virtual attribute is evaluated using the values of `city` and `state` attributes. For example, if city is Los Angeles and state is California, location value is defined as Los Angeles, California.

See Also

[Amazon DynamoDB Developer Guide: Partition Key Design](#)

[Amazon DynamoDB Developer Guide: Adjacency List Design Pattern](#)

[Salesforce Connect Adapter for Amazon DynamoDB](#)

Qualifier Options for Salesforce Connect Adapter for Amazon DynamoDB

You can use various qualifier options to parse an Amazon DynamoDB table into an external object by applying filters to the table rows and performing lightweight transformations of the returned data.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

The following is the syntax for a JSON qualifier expression.

```
TableLocation ::=  
    "tableLocationName": "tableLocationValue" |  
    "tableLocationName": {TableLocation(, TableLocation)*}  
  
NamedColumnQualifier ::=  
    "columnName": {ColumnQualifier(, ColumnQualifier)*}  
  
ColumnQualifier ::=  
    "columnName": "physicalColumnName" |  
    "type": ("S" | "N" | "BOOL") |  
    "presence": ("exists" | "null" | "not null" | "not exists" | "any") |  
    "virtual": (true | false) |  
    "filter": "sqlLikeExpression" |  
    "postFilter": "matchRegex" |  
    "values": [ValueDefinition(, ValueDefinition)*]  
  
ValueDefinition ::=
```

```
{ValueQualifier, ( ValueQualifier) *}

ValueQualifier ::=

  "definition": "functionDefinition" |
  "columnOrder": ["ColumnName" (, "ColumnName") *]
```

- **TableLocation** : Specific set of properties for a data source type that are used to locate the table. For example, database name and database catalog.
- **NamedColumnQualifier** : Default value for the **columnName** is **NamedColumnQualifier**'s column name. If the **ColumnQualifier** is virtual, **columnName** is not specified.
- **ColumnQualifier**
 - **columnName** : Name of the external physical attribute, column, or field.
 - **type** : Some field types may be mapped to more than one attribute type. For example, dates may be mapped as an ISO 8601 string (default) or as a Unix epoch number. You can't use the **type** qualifier to override field types to incompatible attribute types. For example, a field of type number can't be mapped to an attribute type of string.
 - **presence** : Represents whether the attribute must be *present*, *null* (be present and null), *not null* (be present but not null), *absent*, or *any* (presence isn't relevant). Default value is *any*. Amazon DynamoDB server evaluates the presence criteria and translates it as part of the [PartiQL](#) query.
 - **virtual** : If true, the attribute doesn't physically exist in the external data source. Default value is *false*. Virtual attributes are derived from external attributes using the **values** formula. Use virtual attributes to transform encoded or composed attributes, such as transforming a sort key attribute to a user-friendly value. You can recompose physical attribute values by defining value functions with virtual attribute values, for example to create or update records.
 - **filter** : An expression evaluated by the Amazon DynamoDB server to filter matching items.

Allowed expressions for Amazon DynamoDB:

 - **equals**: **ExactValue**
 - **begins with**: **LeftValue%**
 - **contains**: **%MiddleValue%**

Ends with or other substring matches aren't supported.

 - **postFilter** : An expression that runs after the query results are obtained from the Amazon DynamoDB service. Though the postFilter expression refines matching result records, use qualifiers such as **presence** and **filter** for better performance. Back-references (for example, "\1") and lazy quantifiers (for example, "*?") aren't supported for performance reasons.
 - **values** : A list of candidate Salesforce formula functions that derive the attribute value from other attributes. For example, encoding sort keys and deriving composed keys.
- **ValueDefinition** : Contains an array of value qualifiers.
- **ValueQualifier**
 - **definition** : A candidate value defined using Salesforce formula functions. A formula function can be a constant expression or can depend on one or more attribute values.
 - **columnOrder** : The lexical attribute order of the derived attribute value. The lexical order

determines the field sort order in SOQL.

See Also

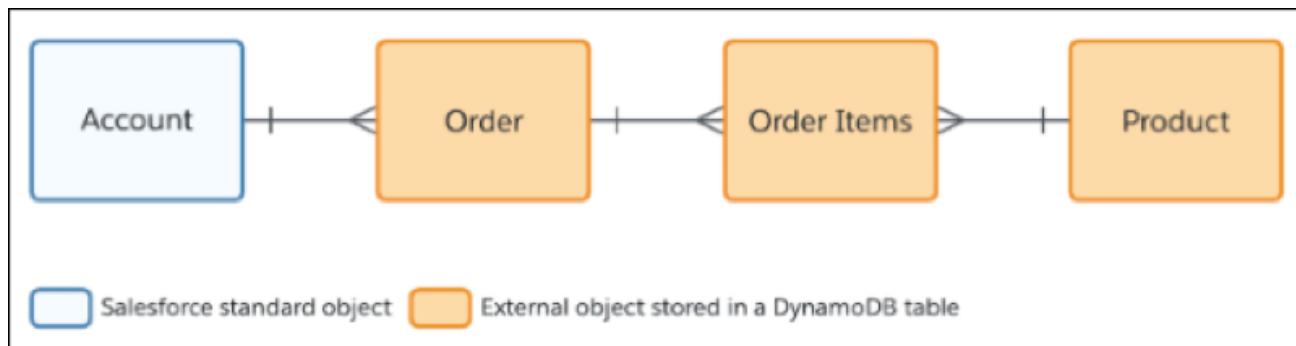
[Amazon DynamoDB Developer Guide: Use PartiQL Functions with Amazon DynamoDB](#)

[Manage Qualifiers for Salesforce Connect Adapter for Amazon DynamoDB](#)

Qualifier Example for One-to-Many and Many-to-Many Relationships in Amazon DynamoDB

An enterprise stores and retains fine-grain transaction details for millions of customers, resulting in data volumes reaching billions of records. In this example, Salesforce is the system of record for core customer information and details of all the customer's orders are stored in Amazon DynamoDB for scale reasons.

In this typical relational model, customers are stored in Salesforce as Accounts. Orders, order items, and products are stored in Amazon DynamoDB and virtualized in Salesforce as external objects. Deploying such a solution provides the enterprise's sellers and customer service agents complete access to customers' order history.



Because Amazon DynamoDB is a NoSQL database and doesn't support JOIN queries, recommended design practice is to store all related information for a business operation in a *single* DynamoDB table. To differentiate items that are products and not orders or order items, you have to use other strategies. Another design practice is to identify access patterns and ensure that common operations can be performed with a single query, thereby maintaining high performance even with very large tables. The following is a sample table based on the recommended design practices.

PK	SK	CUSTOMER	ORDERDATE	TOTAL	SHIPPING	QTY	PRICE	SUBTOTAL	NAME
1012	ORDER	A123	2022-07-12T14:33:14Z	25.75	5.75				
1012	product#1					2	5.00	10.00	Widget X
1012	product#2					1	15.00	15.00	Widget Y
1	PRODUCT						5.00		Widget

PK	SK	CUSTOMER	ORDERDATE	TOTAL	SHIPPING	QTY	PRICE	SUBTOTAL	NAME
									t X
2	PRODUCT						15.00		Widget Y
3	PRODUCT						20.00		Widget Z
1013	ORDER	B456	2022-07-15T18:05:45Z	9.25					
1013	product#1			49.25		1	5.00	5.00	Widget X
1013	product#2					1	15.00	15.00	Widget Y
1013	product#3					1	20.00	20.00	Widget Z

Amazon DynamoDB manages one-to-many relationship between orders and order items based on the [Adjacency list design pattern](#). This pattern allows an order and all its order items to be retrieved via a single query. For example, an order management application deployed on AWS executes the following one query and renders a single receipt for an order. For this purpose, the name of the product is duplicated into the order item at checkout. The goal of the table design isn't to fully normalize data as in relational database design.

```
SELECT * FROM "OrderManagement" WHERE "pk" = "1012"
```

With the structure of the Amazon DynamoDB table known, we can create qualifiers in Salesforce so that the Salesforce Connect adapter for DynamoDB can separate products, orders, and order items. Along with the other elements of the qualifier, the sort key (SK) is used to differentiate between the items (similar to Record Types). For products and orders, the sort key contains a static value of PRODUCT or ORDER as the differentiator. Because of the static value, certain operations are simplified.

- Note** Though JSON doesn't support comments, snippets in this example topic include comments for clarity. If you want to deploy this sample, use a JSON validation tool to point out the syntax to skip.

Manage Qualifiers for Order

The partition key (PK) of an order is numeric, but it's more flexible for the field to be a string relative to Amazon DynamoDB. Hence, the qualifier for Order looks for numeric values via regex even though the attribute type is still a string. The following is a JSON snippet for an Order qualifier.

```
{
  "tableName": "OrderManagement",
  "columns": {
    "pk": {
      "postFilter": "^[0-9]+$", // regex matching integers
      "type": "S" // attribute must be a String
    },
    "sk": {
      "values": [
        {
          "definition": "\"ORDER\""
        }
      ]
    }
  }
}
```

While it's beneficial to map the partition key (PK) field to an Order Number (or a similar field) in Salesforce, the sort key (SK) doesn't need to be mapped to a field on the Order object. It doesn't provide any business value to end users.

Manage Qualifiers for Product

Based on the same approach as for Order, the following is a JSON snippet for a Product qualifier.

```
{
  "tableName": "OrderManagement",
  "columns": {
    "pk": {
      "postFilter": "^[0-9]+$",
      "type": "S"
    },
    "sk": {
      "values": [
        {
          "definition": "\"PRODUCT\""
        }
      ]
    }
  }
}
```

Both Product and Order have other attributes in the Amazon DynamoDB table that must be mapped to

external objects fields. Because these attributes don't need any qualification or translation, they don't need to appear in the qualifier. Salesforce Connect adapter for Amazon DynamoDB maps these other attributes to field types. For example:

- ORDERDATE string attribute is converted to DateTime field type, if it's in ISO 8601 format.
- PRICE and TOTAL number attributes are converted to Currency type configured for the Salesforce org.

Manage Qualifiers for Order Items

Qualifiers allow you to use Salesforce formulas to create virtual attributes that don't physically exist in Amazon DynamoDB. Such attributes can be referenced as external column names when you define the fields on the external object.

The sort key (SK) for Order Items contains an identifier for the product, as well as a static prefix **product#** to clearly identify that the value refers to a product.

The approach used in this example combines PK and SK in a way that works well relative to Amazon DynamoDB. The partition key and sort key together form a unique composite key for an item. When the following qualifier is applied, the external ID in Salesforce for a particular Order Item takes the form of 1012-1 (PK and SK separated by a hyphen). The external ID is mapped to a Salesforce ID to enable the Salesforce Platform capabilities available to external objects.

The following is a JSON snippet for the qualifier that leverages the approach described and uses a formula to remove the prefix and increase readability.

```
{  
    "tableName": "OrderManagement",  
    "columns": {  
        "pk": {  
            "postFilter": "^[0-9]+$", // regex matching integers  
            "type": "S" // attribute must be a String  
        },  
        "sk": {  
            "filter": "products#%" // prefixed with "products#"  
        },  
        "productCode": {  
            "virtual": true, // productCode does not exist in DynamoDB  
            "values": [  
                {  
                    "definition": "RIGHT(sk, LEN(sk) - FIND(\"#\", sk))" // Formula parses out the prefix  
                }  
            ]  
        }  
    }  
}
```

```
}
```

When you create an external lookup to Product from Order Item, use the virtual attribute `productCode` as the external column name. The external lookup to Order uses the PK as the external column name and using this column you can create many-to-many relationships.

See Also

[DynamoDB Qualifier Examples for Parsing Formulas](#)

[Manage Qualifiers for Salesforce Connect Adapter for Amazon DynamoDB](#)

Sort Example for Orders and Order Items in Amazon DynamoDB

To effectively sort orders and order items stored in an Amazon DynamoDB table in a way that benefits a business process, create secondary indexes that identify attribute values as the basis for sorting.

A secondary index allows efficient access to data with attributes other than the primary key values. You can create multiple secondary indexes and give your applications access to different query patterns. A general rule of thumb for creating indexes:

When you query for rows that match condition A, you want to sort the results by attribute B.

A secondary index is a data structure that contains a subset of attributes you select from the table. You must select all the attributes that you want to appear in the query results, even if they're not used for sorting.

Suppose you want to sort orders for a customer by date, and then sort the line items within an order by quantity. To do this, an Amazon DynamoDB administrator must create specific indexes to support the query and the sort operation:

- `customerID-orderDate-index` : query for the orders for a specific customer (partition key), then sort by date (sort key).
- `orderID-quantity-index` : query for the order items for a specific order (partition key), then sort by quantity (sort key).

If you want to display unit price and any other corresponding details for orders or order items, make sure to include the attributes while you're creating the index.

See Also

[Qualifier Example for One-to-Many and Many-to-Many Relationships in Amazon DynamoDB](#)

[Manage Qualifiers for Salesforce Connect Adapter for Amazon DynamoDB](#)

DynamoDB Qualifier Examples for Parsing Formulas

This example shows a simple formula to concatenate two text values. The value of the `location` virtual attribute is evaluated using the values of `city` and `state` attributes. For example, if city is Los Angeles and state is California, location value is defined as Los Angeles, California.

```
{
  "columns": {
    "city": {
      "presence": "any",
      "type": "S",
      "values": [
        {
          "definition": "LEFT(location, FIND("\\", "\\", location)-1)"
        }
      ],
      "state": {
        "presence": "any",
        "type": "S",
        "values": [
          {
            "definition": "MID(location, FIND("\\", "\\", location)+2, LEN(location))"
          }
        ],
        "location": {
          "virtual": "true",
          "values": [
            {
              "definition": "city+\\", \\"+state"
            }
          ]
        }
      }
    }
  }
}
```

This example depicts use of parsing formulas to decode an address stored in a single attribute in Amazon DynamoDB. The `data` field captures the street address as `country#region#city#address` for querying purposes. For example, USA#MI#Ann Arbor#707 Oxford Rd.

```
{
  "columns": {
    "pk": {
      "filter": "suppliers#%",
      "values": [
        {
          "definition": "\\"suppliers#\\" +supplier"
        }
      ]
    },
    "sk": {
      "presence": "exists",
      "values": [
        {

```

```
        "definition": "\"SUPPLIER\""
    }
]
},
"supplier": {
    "virtual": true,
    "type": "S",
    "values": [
        {
            "definition": "MID(pk, FIND("#\", pk) + 1, LEN(pk))"
        }
    ]
},
"data": {
    "presence": "any",
    "type": "S",
    "values": [
        {
            "definition": "country + "#" + region + "#" + city + "#" + address"
        }
    ]
},
"country": {
    "virtual": true,
    "type": "S",
    "values": [
        {
            "definition": "LEFT(data, FIND("#\", data)-1)"
        }
    ]
},
"region": {
    "virtual": true,
    "type": "S",
    "values": [
        {
            "definition": "MID(data, FIND("#\", data)+1, FIND("#\", data, FIND("#\", data)+1)-FIND("#\", data)-1)"
        }
    ]
},
"city": {
    "virtual": true,
```

```
"type": "S",
"values": [
  {
    "definition": "MID(data, FIND(\"#\\", data, FIND(\"#\\", data)+1)+1, FIND(\"#\\", data, FIND(\"#\\", data, FIND(\"#\\", data)+1)+1)-FIND(\"#\\", data, FIND(\"#\\", data)+1)-1)"
  }
],
},
"address": {
  "virtual": true,
  "type": "S",
  "values": [
    {
      "definition": "RIGHT(data, LEN(data)-FIND(\"#\\", data, FIND(\"#\\", data, FIND(\"#\\", data)+1)+1))"
    }
  ]
}
}
```

See Also

[Amazon DynamoDB Developer Guide: Using the BEGINS_WITH Function with PartiQL for DynamoDB](#)
[Manage Qualifiers for Salesforce Connect Adapter for Amazon DynamoDB](#)

Considerations for Salesforce Connect Adapters for Amazon DynamoDB

Understand the special behaviors, limits, and recommendations for using the Salesforce Connect adapter for Amazon DynamoDB.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- When you define an external object, you must specify the prefixes used in the keys. This enables applying a schema-like concept on top of Amazon DynamoDB data that's inherently schema-less. See [Amazon DynamoDB Developer Guide: Core Components of Amazon DynamoDB](#).
- When you define a DynamoDB external data source, you must manually set up metadata for the related external objects, and fields on the external objects.

- Not all relational-style SOQL queries can be translated in a way that returns results in DynamoDB, such as table joins. Such queries can't execute and may cause a runtime error displayed in the Lightning UI. Amazon DynamoDB delivers reliably fast performance at extremely high scale by mandating developers store data in a way that matches how it will be accessed. Typical relational databases don't impose this restriction on development teams, but JOINs and subqueries at the very highest levels of scale become problematic. See *Amazon DynamoDB Developer Guide: From SQL to NoSQL*.
- External object fields map to a DynamoDB attribute value in its entirety, so any compound values with embedded keys are visible to the user.
- Salesforce Platform features that aren't supported:
 - Reports
 - SOSL
 - Sorting is only available based on the sort key of the underlying table in Amazon DynamoDB, or any secondary indexes defined for that table. See [Sort Example for Orders and Order Items](#).
 - Filtering isn't supported on [virtual fields](#). Filtering is supported only for columns that are present in the DynamoDB data source.
 - Certain operations, such as updates via Flows work only when the primary key for the mapped DynamoDB table is of type *string*.
- A SOQL query may cause a runtime error if:
 - The query scans an Amazon DynamoDB table that is configured to prevent such scans.
 - The query can't be translated in a way that returns results in Amazon DynamoDB such as table joins. Because Amazon DynamoDB is a nonrelational database, it doesn't support table joins.

To understand more, see [Amazon DynamoDB Developer Guide: Working with Scans in DynamoDB](#).

- We recommend consistent reads for working with Salesforce integration. For information reading data from a DynamoDB table, see [Amazon DynamoDB Developer Guide: Read Consistency](#).
- Auto-numbering of identifiers isn't supported. So DML operations must always include primary key fields.
- Asynchronous DML operations aren't supported. Instead use the synchronous variant of the DML operations. [Amazon DynamoDB database service](#) is designed to run high-performance applications at any scale.
- For external lookups and indirect lookups on external objects:
 - External Lookup Relationship: Look up an external parent by its primary key.
 - Indirect Lookup Relationship: Look up a parent by key fields on the parent that uniquely identify a record. The child record holds the values to match for the parent unique field. For efficient queries, we recommend indexing parent key fields that uniquely identify the record and match the child record.
- External ID values that uniquely identify external object records are case sensitive.

See Also

[Salesforce Connect Adapter for Amazon DynamoDB](#)

[Set Up Salesforce Connect to Access External Data in Amazon DynamoDB](#)

Access External Data with the Salesforce Connect Adapter for SQL

Connect your users to external data sources that expose their capabilities via REST APIs and offer query and DML operations using SQL.

Salesforce Connect SQL Adapter for Amazon Athena

Provide your users access to Amazon Athena's capability to analyze data directly in Amazon Simple Storage Service (Amazon S3) and integrate AWS-hosted data in Salesforce applications.

Salesforce Connect SQL Adapter for Snowflake

Query structured data stored in Snowflake and combine it with the customer 360-degree view captured in Salesforce.

Considerations for Salesforce Connect Adapters for SQL

Understand the special behaviors, limits, and recommendations for using the Salesforce Connect adapters for SQL.

Salesforce Connect SQL Adapter for Amazon Athena

Provide your users access to Amazon Athena's capability to analyze data directly in Amazon Simple Storage Service (Amazon S3) and integrate AWS-hosted data in Salesforce applications.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Perform complex queries on large datasets and get fast query results using the Salesforce Connect SQL adapter for Amazon Athena. With the SQL adapter, take advantage of Amazon Athena's capability to run queries against data directly in Amazon S3 without managing RDBMS infrastructure or ETL tools.

Amazon Athena is an interactive query service that makes it easy to query structured data stored in Amazon S3 using standard SQL. Use the Salesforce Connect SQL adapter for Amazon Athena to query data stored in Amazon S3 and combine it with the customer 360-degree view captured in Salesforce.

Amazon Athena provides additional data connectors that run as AWS Lambda functions and can query nearly any data source on AWS. However, one of the trade-offs is that DML operations such as insert, update, and delete aren't supported on all the data sources. For more information, see [Amazon Athena User Guide: Running SQL queries using Amazon Athena](#).

Map Salesforce External Object Field to Amazon Athena Data Types

External object fields of Salesforce Connect SQL adapter for Amazon Athena can be mapped to columns with scalar types such as strings and numbers. External object fields can't be mapped to columns containing complex or nested data types such as an array, map, or struct.

Set Up Salesforce Connect SQL Adapter for Amazon Athena

Provide users easy access to data stored in Amazon S3 so that they can build custom applications that combine the power of the Salesforce and AWS clouds.

[Manage Qualifiers for Salesforce Connect SQL Adapter for Amazon Athena](#)

Tables and databases in Amazon Athena contain the metadata definitions for the underlying source data schema. To be able to make queries to Amazon Athena, use qualifiers to specify the key columns that identify records in the Amazon Athena data source.

See Also

[Salesforce Connect](#)

[Set Up Salesforce Connect SQL Adapter for Amazon Athena](#)

[Manage Qualifiers for Salesforce Connect SQL Adapter for Amazon Athena](#)

Map Salesforce External Object Field to Amazon Athena Data Types

External object fields of Salesforce Connect SQL adapter for Amazon Athena can be mapped to columns with scalar types such as strings and numbers. External object fields can't be mapped to columns containing complex or nested data types such as an array, map, or struct.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

The table details the mappings between column types in Amazon Athena to external object field types in Salesforce.

Amazon Athena Data Type	Salesforce External Object Field Type
boolean	Checkbox
tinyint, smallint, int and integer, bigint, double, float, decimal	Number Mapped to default precision and scale, which you can override.
char	Text or Text Area
varchar, string	Text, Text Area, or Text Area (Long)
date	Date Supported format is YYYY-MM-DD
timestamp	Date/Time

Amazon Athena Data Type	Salesforce External Object Field Type
	<p>Supported format is yyyy-MM-dd HH:mm:ss.f</p> <p>For example, '2008-09-15 03:04:05.324'</p>

 **Note** For **Time** Salesforce field type, there isn't an equivalent column type in Amazon Athena.

This mapping strategy is for Amazon Athena data types to some of the common field types of external objects in Salesforce.

- A char, string, or varchar data type that represents a phone, URL, Text, Text Area, Text Area (Long), or Email is stored as the corresponding Salesforce field type.
- A char, string, or varchar data type that represents a Picklist is rendered as Salesforce defined enumeration values. For Picklist (Multi-Select), it's stored as comma-separated values of the selected choices.
- A decimal data type that represents a percent is stored as a numeric value of percent type.
- A decimal data type that represents currency is stored as a numeric value using the currency symbol configured for the Salesforce org.

 **Note** When you manually create fields for an external object, make sure that the mapped attributes are of a compatible type.

See Also

[Data types in Amazon Athena](#)

[Custom Field Types](#)

[Salesforce Connect SQL Adapter for Amazon Athena](#)

Set Up Salesforce Connect SQL Adapter for Amazon Athena

Provide users easy access to data stored in Amazon S3 so that they can build custom applications that combine the power of the Salesforce and AWS clouds.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit external data sources:

Customize Application

USER PERMISSIONS NEEDED

To create and edit external objects:	Customize Application
To view named credentials:	View Setup and Configuration
To create, edit, or delete named credentials:	Manage Named Credentials or Customize Application
To create and edit custom fields:	Customize Application
To edit permission sets and user profiles:	Manage Profiles and Permission Sets
To edit another user's authentication settings for external systems:	Manage Users

Before you begin, review the basics of Amazon Athena setup and access in [Amazon Athena User Guide](#).

To know more about identity-based AWS Identity and Access Management (IAM) policies to restrict access to Athena operations, see [Identity and access management in Athena](#).

[Define a Named Credential for Salesforce Connect SQL Adapter for Amazon Athena](#)

Create a named credential that specifies the endpoint URL for Amazon Athena and an external credential to provide the required authentication parameters.

[Define a Legacy Named Credential for Salesforce Connect SQL Adapter for Amazon Athena](#)

Create a legacy named credential that specifies the URL of a callout endpoint as an Amazon Athena endpoint and provides the required authentication parameters.

[Create an External Data Source for Salesforce Connect SQL Adapter for Amazon Athena](#)

Connect your Salesforce org to access Amazon Athena's interactive query capabilities.

[Validate and Sync External Data Source Configured for Amazon Athena](#)

After you create an external data source for Amazon Athena, synchronize it to map its tables with external objects in your Salesforce org.

See Also

[Salesforce Connect](#)

[Salesforce Connect SQL Adapter for Amazon Athena](#)

[Manage Qualifiers for Salesforce Connect SQL Adapter for Amazon Athena](#)

Define a Named Credential for Salesforce Connect SQL Adapter for Amazon Athena

Create a named credential that specifies the endpoint URL for Amazon Athena and an external credential to provide the required authentication parameters.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

USER PERMISSIONS NEEDED

To view named credentials: View Setup and Configuration

To create, edit, or delete named credentials: Manage Named Credentials or Customize Applications

1. From Setup, in the Quick Find box, enter *Named Credentials*, and then select **Named Credentials**.
2. Click **External Credentials**.
3. To create a new external credential, click **New**. To edit an existing external credential, click its link in the list of external credentials and then click **Edit**.
4. Complete the fields.

Field	Description
Label	A user-friendly name for the named credential that's displayed in the Salesforce user interface.
Name	A unique identifier that's used to refer to this external credential from callout definitions while creating an external data source for Amazon Athena.
Authentication Protocol	Select AWS Signature Version 4 .
Service	The name of an AWS service, such as <i>athena</i> .
Region	The AWS region for the named credential's endpoint. For example, <i>us-west-2</i> .
AWS Account ID	Optional. The 12-digit number that uniquely identifies your AWS account.
Use STS for Temporary Access	Select the checkbox to provide limited access and specify STS access key, access secret, external ID, and duration. For details, see Create and Edit an AWS Signature v4 External Credential .

5. Save your changes. You're taken to the Named Credentials screen.
6. To further edit the new external credential, click **External Credential**.
7. Select the external credential you created.
8. Scroll to Permission Set Mappings.
9. To create a permission set mapping for this external credential, click **New**.
10. Complete these fields. When you use STS for temporary access, the Access Key and Secret fields are disabled and display the temporary credentials, if any.

Field	Description
Permission Set	Select an available permission set. When you do, different groups of Salesforce users can mirror access permissions of IAM roles.

Field	Description
	<p>Make sure that you define the required access to the User External Credentials object (use permission sets or profiles to configure object access). Only users with access to the User External Credentials object can make callouts to the external source. For details, see Named Credentials and External Credentials.</p>
Sequence Number	<p>Assign a sequence number. A sequence number specifies the order of principals to apply when a user participates in more than one principal. For example, a user can be part of multiple permission sets that are applicable for a credential provider. Priority is from lower to higher numbers.</p>
IAM Role ARN	<p>The Amazon Resource Name (ARN) of the role that the credential assumes.</p> <p>To get the ARN for an IAM role:</p> <ol style="list-style-type: none"> a. In the navigation pane of the IAM console, choose Roles. b. In the list of roles, choose the role that you want to map to the permission set. c. In the Summary section, copy the ARN value. <p>For details, see AWS Identity and Access Management User Guide: Tutorials.</p>

11. Create a named credential that uses the authentication configuration you defined in the external credential. From Setup, in the Quick Find box, enter *Named Credentials*, select **Named Credentials**, and then click **Named Credentials**.
12. To create a new named credential, click **New** and complete the fields.

- a. Fill out the **Label**, **Name**, and **URL** fields.

The sample format for the URL is `https://athena.REGION.amazonaws.com`. For example, if the AWS region is `us-west-2`, the URL is `https://athena.us-west-2.amazonaws.com`.

For details, see [Create and Edit a Named Credential](#).

- b. For the **External Credential** field, select the external credential you created that uses the AWS Signature Version 4 authentication protocol.
- c. Select **Generate Authorization Header**.

Use the named credential that captures the authentication configuration in external credentials to authenticate Salesforce users against Amazon Athena, provide limited access to AWS resources, and periodically refresh access tokens.

See Also

[Named Credentials](#)

Define a Legacy Named Credential for Salesforce Connect SQL Adapter for Amazon Athena

Create a legacy named credential that specifies the URL of a callout endpoint as an Amazon Athena endpoint and provides the required authentication parameters.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

USER PERMISSIONS NEEDED

To view named credentials:	View Setup and Configuration
----------------------------	------------------------------

To create, edit, or delete named credentials:	Customize Applications
---	------------------------

! **Important** In Winter '23, Salesforce introduced an improved named credential that is extensible, customizable, and more secure. We strongly recommend that you use this preferred credential instead of legacy named credentials, which are no longer updated or enhanced. For information on extensible, customizable named credentials, see [Named Credentials Schema](#).

1. From Setup, in the Quick Find box, enter *Named Credentials*, and then select **Named Credentials**.
2. To create a legacy named credential, click **New Legacy** from the dropdown menu.
3. Complete the fields.

Field	Description
Label	A user-friendly name for the named credential that's displayed in the Salesforce user interface.
Name	A unique identifier that's used to refer to this named credential from callout definition while creating an external data source for Amazon Athena.
URL	<p>The URL of the Amazon Athena endpoint.</p> <p>The sample format of the URL is <code>https://athena.REGION.amazonaws.com</code> .</p> <p>For example, if the AWS region is <i>us-west-2</i>, the URL is <code>https://athena.us-west-2.amazonaws.com</code> .</p>
Identity Type	To designate one user account on the Amazon Athena external system for all your Salesforce org users, select Named Principal .

Field	Description
Authentication Protocol	Select AWS Signature Version 4 , and then complete these fields.

- a. **AWS Access Key ID:** The first part of the access key used to sign programmatic requests to AWS.
- b. **AWS Secret Access Key:** The second part of the access key used to sign programmatic requests to AWS.
- c. **AWS Region:** The AWS region name for the named credential's endpoint. For example, *us-west-2*.
- d. **AWS Service:** The AWS utility to access. For example, *athena*.

4. Save your changes.

See Also

[AWS Account and Access Keys](#)

[Named Credentials](#)

[Salesforce Connect SQL Adapter for Amazon Athena](#)

Create an External Data Source for Salesforce Connect SQL Adapter for Amazon Athena

Connect your Salesforce org to access Amazon Athena's interactive query capabilities.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit external data sources: Customize Application

1. From Setup, in the Quick Find box, enter *External Data Sources*, and then select **External Data Sources**.
2. Click **New External Data Source**, or click **Edit** to modify an existing external data source.
3. Complete these fields.

Field	Description
External Data Source	A user-friendly name for the external data source. The label is displayed in the Salesforce user interface.

Field	Description
Name	A unique identifier that's used to refer to this external data source definition through the API.
Type	Select SQL .
Provider	Select Amazon Athena .
Data Catalog	<p>The name of the AWS Glue Data Catalog where the target database is registered.</p> <p>In most cases, it's the default <code>AwsDataCatalog</code> data source.</p>
Named Credential	<p>Enter the named credential URL that you defined for Amazon Athena data source.</p> <p>You can skip the Authentication section for the external data source. To access the Amazon Athena external system, Salesforce Connect uses the authentication settings that are defined in the named credential.</p>
Connection Timeout	Number of seconds to wait for a response from the Amazon Athena external system before timing out. By default, the value is set to the maximum of 120 seconds.
Use Cached Results	Select to reuse the stored query results and accelerate the performance of your repeat queries.
Maximum Age for Cached Results	Enter the maximum age for using the cached query results. The valid range is 1–10,080 minutes.



Note Salesforce Connect SQL adapter for Amazon Athena doesn't support writable external objects. The adapter only supports read operations on the queried data.

4. Save your changes.

See Also

[Work with External Data Sources](#)

[Salesforce Connect SQL Adapter for Amazon Athena](#)

Validate and Sync External Data Source Configured for Amazon Athena

After you create an external data source for Amazon Athena, synchronize it to map its tables with external objects in your Salesforce org.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external

objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create an external object from an external data source: Customize Application

1. From Setup, in the Quick Find box, enter *External Data Sources*, and then select **External Data Sources**.
2. Open the external data source that you created for the Amazon Athena external system.
3. Click **Validate and Sync**.
4. Select the Amazon Athena database that contains the tables that you want to sync.
You can see only the first 50 databases from Amazon Athena.
5. Select the tables, and click **Sync**.

 **Note** Make sure that the external object field names exactly match the table column names in Amazon Athena. Otherwise, queries to Amazon Athena cause a runtime error.

After the external object and its fields are created, you must provide additional configuration so that Salesforce can access Amazon Athena to run queries and respect settings configured by the AWS administrator. See [Manage Qualifiers for Salesforce Connect SQL Adapter for Amazon Athena](#).

See Also

[Work with External Data Sources](#)

[Salesforce Connect SQL Adapter for Amazon Athena](#)

[Manage Qualifiers for Salesforce Connect SQL Adapter for Amazon Athena](#)

Tables and databases in Amazon Athena contain the metadata definitions for the underlying source data schema. To be able to make queries to Amazon Athena, use qualifiers to specify the key columns that identify records in the Amazon Athena data source.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

When you query an Amazon Athena table, the data catalog and database that you configured determine the table location. To provide additional configuration, follow these steps.

1. From Setup, in the Quick Find box, enter *External Data Sources*, and then select **External Data Sources**.
2. Select and click the external data source to edit.
3. Under External Objects, select the synced external object to which you want to add qualifiers and click **Manage Qualifiers**.

Field	Description
Database	Displays the configured Athena database that contains the synced table for the corresponding external object.
Key Column Names	<p>Specify the column names that represent a unique key for a row in the synced Amazon Athena table. The Salesforce Connect SQL adapter uses these column values to build an external ID for external object records in Salesforce.</p> <p>Composite keys can be represented as a comma-delimited list. For example, <code>orderId,productId</code>.</p>
Workgroup	<p>Specify the name of the workgroup used by the Salesforce Connect SQL adapter to make queries to Amazon Athena. The primary workgroup is the default.</p> <p>AWS administrators use workgroups to control query access and costs. To respect those settings, specify the <code>workgroup</code> option.</p> <p>If you want to create a workgroup to isolate queries from different workloads, see Using workgroups for running queries.</p>

4. Save your changes.

See Also

[Amazon Athena User Guide: How workgroups work](#)
[Salesforce Connect SQL Adapter for Amazon Athena](#)

Salesforce Connect SQL Adapter for Snowflake

Query structured data stored in Snowflake and combine it with the customer 360-degree view captured in Salesforce.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external

objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

In many cases, Salesforce is the system of record for core customer information and sales, and details of all the customer's orders are stored in Snowflake for scale reasons. With the Salesforce Connect SQL adapter for Snowflake, sellers or support agents have easy access to invoice, payment, shipment, and return data.

[Map Salesforce External Object Fields to Snowflake Data Types](#)

External object fields can be mapped to Snowflake data table columns with scalar types such as strings and numbers. External object fields can't be mapped to columns containing complex or semi-structured data types such as a variant, object, or array.

[Set Up Salesforce Connect SQL Adapter for Snowflake](#)

Provide users easy access to data stored in Snowflake so that they can build custom applications that combine the power of the Salesforce and Snowflake platforms.

[Manage Qualifiers for Salesforce Connect Adapter SQL for Snowflake](#)

Tables and databases in Snowflake contain the metadata definitions for the underlying source data schema. To be able to make queries to Snowflake, use qualifiers to specify the key columns that identify records in the Snowflake data source.

See Also

[Salesforce Connect](#)

Map Salesforce External Object Fields to Snowflake Data Types

External object fields can be mapped to Snowflake data table columns with scalar types such as strings and numbers. External object fields can't be mapped to columns containing complex or semi-structured data types such as a variant, object, or array.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

This table details the mappings between column types in Snowflake and external object field types in Salesforce.

Snowflake Data Type	Salesforce External Object Field Type
BOOLEAN	Checkbox
NUMBER, DECIMAL, NUMERIC, INT, INTEGER, BIGINT, SMALLINT, TINYINT, BYTEINT, FLOAT, FLOAT4, FLOAT8, DOUBLE, DOUBLE PRECISION, REAL	Number For simplicity, Snowflake exposes all numeric columns in a consistent way that maps to Number fields in Salesforce. To align fields across Snowflake and Salesforce, match the precision and scale carefully based on your data and the column definition.
STRING, TEXT, VARCHAR, CHAR, CHARACTER	Text, Text Area, Text Area (Long), Phone, Email, URL, Picklist, and Picklist (Multi-Select) If you map Snowflake data to Picklist or Picklist (Multi-Select) field types in Salesforce, ensure that values are in a picklist value set and that multi-select picklist values are separated by a semicolon.
DATE	Date
TIME	Time
DATETIME, TIMESTAMP, TIMESTAMP_LTZ, TIMESTAMP_NTZ, TIMESTAMP_TZ	Date/Time Data stored in DATETIME and TIMESTAMP fields without a timezone in Snowflake is interpreted in Coordinated Universal Time (UTC) in Salesforce. Salesforce users see these dates in their locale's timezone.
GEOGRAPHY	Text

This mapping strategy is for Snowflake data types to some of the common field types of external objects in Salesforce.

- A CHAR, STRING, or VARCHAR data type that represents a Phone, URL, Text, Text Area, Text Area (Long), or Email is stored as the corresponding Salesforce field type.
- A CHAR, STRING, or VARCHAR data type that represents a Picklist is rendered as Salesforce-defined enumeration values. For Picklist (Multi-Select), it's stored as semicolon-separated values of the selected choices.
- A decimal data type that represents a percent is stored as a numeric value of the percent type.
- A decimal data type that represents currency is stored as a numeric value using the currency symbol configured for the Salesforce org.

 **Note** When you create fields for an external object manually, make sure the mapped attributes are of a compatible type.

See Also

[Snowflake Documentation: Summary of Data Types](#)[Custom Field Types](#)[Salesforce Connect SQL Adapter for Snowflake](#)

Set Up Salesforce Connect SQL Adapter for Snowflake

Provide users easy access to data stored in Snowflake so that they can build custom applications that combine the power of the Salesforce and Snowflake platforms.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit external data sources:	Customize Application
To create and edit external objects:	Customize Application
To view named credentials:	View Setup and Configuration
To create, edit, or delete named credentials:	Manage Named Credentials or Customize Application
To create and edit custom fields:	Customize Application
To edit permission sets and user profiles:	Manage Profiles and Permission Sets
To edit another user's authentication settings for external systems:	Manage Users

Before you begin, review the basics of Snowflake setup and access in the [Snowflake Getting Started guide](#). We also recommend creating a view or dynamic table in Snowflake to limit the data that you want Salesforce users to be able to access.

[Define a Named Credential for Salesforce Connect SQL Adapter for Snowflake](#)

Create a named credential that specifies the endpoint URL for Snowflake and an external credential to provide the required authentication parameters.

[Create an External Data Source for Salesforce Connect SQL Adapter for Snowflake](#)

Connect your Salesforce org to access Snowflake's interactive query capabilities.

[Validate and Sync External Data Source Configured for Snowflake](#)

After you create an external data source for Snowflake, synchronize it to map its data to external objects in your Salesforce org.

See Also

- [Salesforce Connect](#)
- [Salesforce Connect SQL Adapter for Snowflake](#)
- [Manage Qualifiers for Salesforce Connect Adapter SQL for Snowflake](#)

Define a Named Credential for Salesforce Connect SQL Adapter for Snowflake

Create a named credential that specifies the endpoint URL for Snowflake and an external credential to provide the required authentication parameters.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: all editions

USER PERMISSIONS NEEDED

To view named credentials: View Setup and Configuration

To create, edit, or delete named credentials: Manage Named Credentials or Customize Applications

When you use the Salesforce Connect adapter for SQL to integrate with Snowflake, Salesforce acts as a client (consumer) of Snowflake's services. Enabling connectivity between services requires additional configuration in both Salesforce and Snowflake. To integrate Salesforce and Snowflake, perform this series of tasks.

1. In Salesforce, configure an authentication provider, and copy its redirect URI for the OAuth flow.
2. In Snowflake, configure an integration that references the redirect URI from Salesforce.
3. In Salesforce, update the authentication provider that you created with parameters from the Snowflake integration.
4. If your Snowflake account is configured so that AWS PrivateLink bypasses the public internet, set up an outbound connection in Salesforce to send traffic from Salesforce to your AWS Virtual Private Cloud (VPC).
5. In Salesforce, create an external credential that uses the authentication provider that you configured.
6. In Salesforce, create a named credential that uses the external credential that you configured.

Configure an Authentication Provider in Salesforce

To get started, create an authentication provider using OpenID Connect in Salesforce.

Give the authentication provider a name like *Snowflake*, and enter placeholder values in other required fields. You update these fields after you create the integration in Snowflake. For more information, see [Configure an Authentication Provider Using OpenID Connect](#).

After you create the authentication provider, on the Auth. Provider Detail page, copy the callback URL

that Salesforce generated. When you create the integration in Snowflake, the callback URL is the redirect URI that redirects the Salesforce admin back to Salesforce after they authenticate in Snowflake.

Configure an Integration in Snowflake

Next, create an integration in Snowflake to enable access to Snowflake's SQL API. We recommend using Snowflake OAuth for custom clients.

Name the integration Salesforce, and use the callback URL that you copied from Salesforce as the value for the OAUTH_REDIRECT_URI. For more information, see [Configure Snowflake OAuth for Custom Clients](#).

After you create the integration, use this Snowflake query to see the Salesforce integration's parameters.

```
DESCRIBE INTEGRATION SALESFORCE;
```

In the query results, retrieve and save these values to add to the authentication provider that you created in Salesforce.

- `OAUTH_CLIENT_ID`
- `OAUTH_Authorization_Endpoint`
- `OAUTH_TOKEN_ENDPOINT`

Next, retrieve the client secret. Use this query and save the `OAUTH_CLIENT_SECRET` that Snowflake returns.

```
SELECT SYSTEM$SHOW_OAUTH_CLIENT_SECRETS( 'SALESFORCE' );
```

Before you create external and named credentials in Salesforce, we also recommend creating a user with no administrator privileges in your Snowflake account. Assign the Snowflake user a role that isn't ACCOUNTADMIN or SYSTEMADMIN, and grant the user access to the appropriate warehouse, database, schema, and tables. For more information, see [Configuring Access Control](#) in the Snowflake documentation.

Update the Authentication Provider in Salesforce

Next, in Salesforce, update the authentication provider that you created with the values from the Snowflake integration.

Salesforce Authentication Provider Field	Snowflake Integration Value
Consumer Key	<code>OAUTH_CLIENT_ID</code>
Consumer Secret	<code>OAUTH_CLIENT_SECRET</code>

Salesforce Authentication Provider Field	Snowflake Integration Value
Authorize Endpoint URL	OAUTH_AUTHORIZATION_ENDPOINT
Token Endpoint URL	OAUTH_TOKEN_ENDPOINT

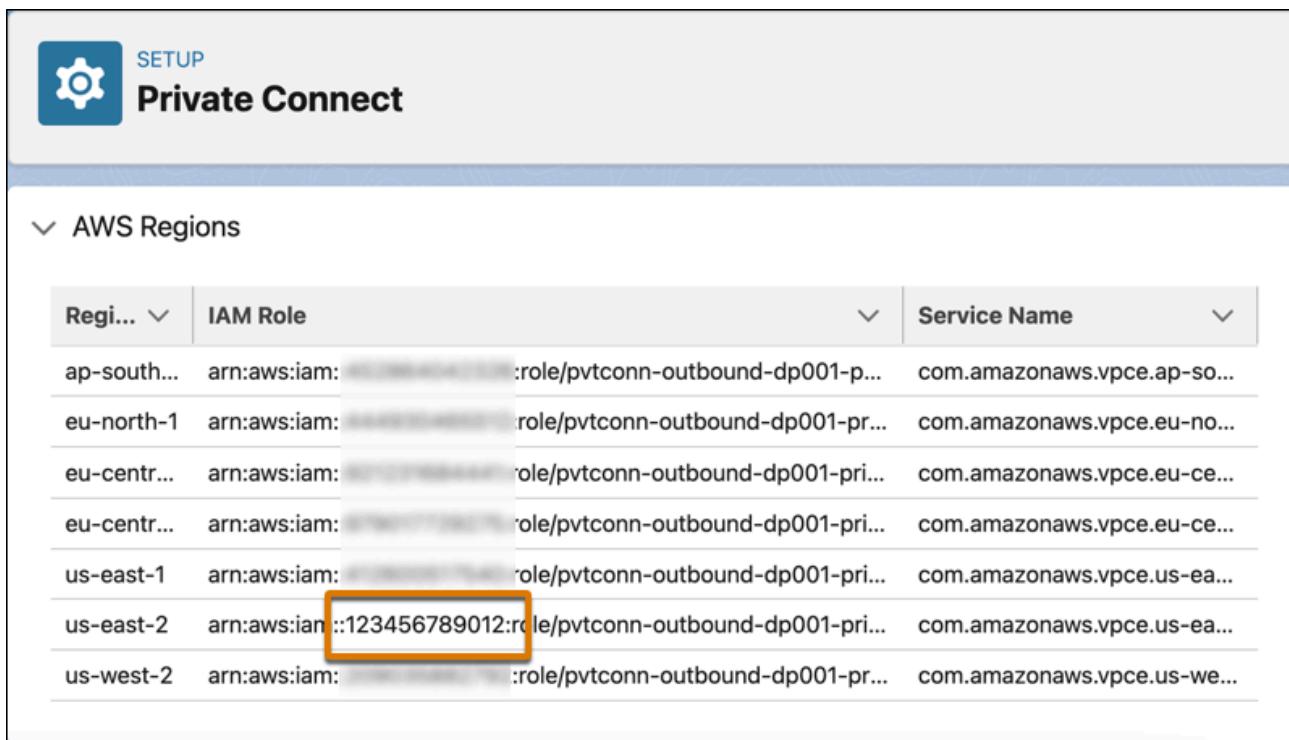
Establish an Outbound Connection with AWS in Salesforce

If your Snowflake account is configured so that AWS PrivateLink bypasses the public internet, you must set up a Private Connect outbound connection in Salesforce to send traffic from Salesforce to your AWS VPC. If your Snowflake account doesn't use AWS PrivateLink, skip this task.

 **Note** Private Connect is available for purchase as an add-on license. See [Secure Cross-Cloud Integrations with Private Connect](#).

Before you create the outbound connection in Salesforce, work with your Snowflake admin or Snowflake Support to allow Salesforce to connect privately. See [AWS PrivateLink and Snowflake](#) and [Allowing Hostnames](#) in the Snowflake documentation.

You must also work with your Snowflake admin to allowlist the Salesforce AWS account for Snowflake. In Snowflake, identify your account's AWS region. Then, from Salesforce Setup, in the Quick Find box, enter *Private*, and then click *Private Connect*. Open the AWS Regions list and find the row for the AWS region that your Snowflake account is in. In the IAM Role column, identify the AWS account number and share it with your Snowflake admin so that they can add the account to the list of allowed hostnames.



The screenshot shows the Salesforce Setup interface under the Private Connect section. A dropdown menu is open, showing the option 'AWS Regions'. Below this, a table lists regions and their corresponding IAM roles. The table has columns for Region, IAM Role, and Service Name. One row for the 'us-east-2' region is highlighted with an orange box around its IAM Role value: 'arn:aws:iam::123456789012:role/pvtconn-outbound-dp001-pr...'. This value represents the AWS account number followed by the role name.

Region	IAM Role	Service Name
ap-south-1	arn:aws:iam::[REDACTED]:role/pvtconn-outbound-dp001-p...	com.amazonaws.vpce.ap-so...
eu-north-1	arn:aws:iam::[REDACTED]:role/pvtconn-outbound-dp001-pr...	com.amazonaws.vpce.eu-no...
eu-central-1	arn:aws:iam::[REDACTED]:role/pvtconn-outbound-dp001-pr...	com.amazonaws.vpce.eu-ce...
eu-west-1	arn:aws:iam::[REDACTED]:role/pvtconn-outbound-dp001-pr...	com.amazonaws.vpce.eu-ce...
us-east-1	arn:aws:iam::[REDACTED]:role/pvtconn-outbound-dp001-pr...	com.amazonaws.vpce.us-ea...
us-east-2	arn:aws:iam::123456789012:role/pvtconn-outbound-dp001-pr...	com.amazonaws.vpce.us-ea...
us-west-2	arn:aws:iam::[REDACTED]:role/pvtconn-outbound-dp001-pr...	com.amazonaws.vpce.us-we...

After your Salesforce account is allowlisted, return to Snowflake and run this query.

```
SELECT SYSTEM$GET_PRIVATELINK_CONFIG();
```

In the query results, retrieve and save the values of `PRIVATELINK-VPCE-ID` and `PRIVATELINK-ACCOUNT-URL` to add to the outbound connection and named credential in Salesforce. We also recommend saving the `PRIVATELINK-ACCOUNT-NAME` to help you troubleshoot issues with the Snowflake team.

In Salesforce, create the outbound connection.

1. From Setup, in the Quick Find box, enter *Private*, and then click **Private Connect**.
2. Click **Create Outbound Connection**, select **AWS PrivateLink**, and then click **Next**.
3. Complete the fields.

Field	Description
Connection Name	Enter a name that references the external system, such as <code>SnowflakeOutboundConnection</code> .
VPC Endpoint Service Name	Enter the <code>PRIVATELINK-VPCE-ID</code> value that you copied from the Snowflake account. For example, <code>com.amazonaws.vpce.us-west-2.vpce-svc-025d08f3df604a276</code> .
Region	Select the AWS region in which your VPC is hosted. For example, for the VPC endpoint <code>com.amazonaws.vpce.us-west-2.vpce-svc-025d08f3df604a276</code> , the AWS region is <code>us-west-2</code> .

4. Select **Yes, I would like to provision my connection now**.
5. Save the connection.

Create an External Credential for the Snowflake Integration

Now you're ready to create an external credential in Salesforce. We recommend using the OAuth 2.0 Browser Flow with the authentication provider that you created for the Snowflake account.

1. From Setup, in the Quick Find box, enter *Named Credentials*, and then select **Named Credentials**.
2. Click **External Credentials**.
3. To create an external credential, click **New**.
4. Complete the fields.

Field	Description
Label	A user-friendly name for the external credential that's displayed in the Salesforce user interface, such as in list views.
Name	A unique identifier that's used to refer to this external credential from callout definitions while creating an external data source for Snowflake.

Field	Description
Authentication Protocol	Select OAuth 2.0 .
Authentication Flow Type	Select Browser Flow .
Scope	To allow offline access, enter <code>refresh_token</code> .
Authentication Provider	Select the authentication provider that you created to access Snowflake.

5. Save the external credential.

You're taken to the Named Credentials screen.

Create the External Credential Principal

Next, create the principal, authenticate to the external system, and grant access to Salesforce users.

1. On the Named Credentials screen, scroll to the Principals section.
2. To create a principal, click **New**.
3. Enter the information for the principal.

Field	Description
Parameter Name	Enter a name that references the external system, such as <code>SnowflakeIntegrationUser</code> .
Sequence Number	Assign a sequence number. A sequence number specifies the order of principals to apply when a user participates in more than one principal. For example, a user can be part of multiple permission sets that are applicable for a credential provider. Priority is from lower to higher numbers.
Identity Type	<p>Select Named Principal.</p> <p>With the Named Principal identity type, your organization uses only one login account to access the Snowflake external system.</p>
Scope	<p>Optional. The Scope field accepts static values and formulas. To tie a group of authenticated users to a role in the Snowflake system, enter <code>session:role:<snowflake_role></code>.</p> <p>For example, if at run time the user has Sales or Service in their department name, set a static scope of <code>session:role:CUSTOMER_SERVICE</code>.</p>

Field	Description
	<p>Or enter a formula to request access dynamically. This example uses the <code>session:role</code> scope to request credential-level access based on each user's department.</p> <pre>{!"session:role:" + \$User.Department}</pre>

4. Save the principal.
5. To authenticate the principal, click its **Actions** menu, and then select **Authenticate**. You're taken to Snowflake. Enter your login credentials, and then return to the Setup page in Salesforce to finish creating the external and named credentials.
6. In Salesforce, create or update a permission set to grant principal access so that users can make authenticated callouts to Snowflake. See [Enable External Credential Principals](#).

Create a Named Credential for the Snowflake Integration

Finally, create a named credential that uses the authentication configuration that you defined in the external credential.

1. From Setup, in the Quick Find box, enter *Named Credentials*, and then click **Named Credentials**.
2. To create a named credential, click **New**.
3. Complete the fields.

Field	Description
Label	A user-friendly name for the named credential that's shown in the Salesforce user interface, such as in list views.
Name	A unique identifier that's used to refer to this named credential from a callout definition while creating an external data source for Snowflake.
URL	<p>Enter a region-based URL with the format <code>https://your_snowflake_org.snowflakecomputing.com/api/v2/statements/</code>.</p> <p>For example, if the Snowflake org is <code>qab39024.us-east-1</code>, the URL is <code>https://qab39024.us-east-1.snowflakecomputing.com/api/v2/statements/</code>.</p> <p>To retrieve your Snowflake org, use this command.</p> <pre>SELECT CURRENT_ACCOUNT();</pre>

Field	Description
	<p>If your URL doesn't include a region, you must include the Snowflake account as a custom HTTP header. See more information in the description of the Allow Formulas in HTTP Header field.</p> <p>If your Snowflake account uses AWS PrivateLink to bypass the public internet, enter the <code>PRIVATELINK-ACCOUNT-URL</code> value that you copied from the Snowflake account appended with <code>/api/v2/statements/</code>. For example, <code>https://qab39024.us-east-1.privatelink.snowflakecomputing.com/api/v2/statements/</code>.</p>
External Credential	Select the Snowflake external credential you created that uses the OAuth 2.0 authentication protocol with the Browser Flow.
Generate Authorization Header	Select this checkbox so that Salesforce generates an authorization header and applies it to each callout that references the named credential.
Allow Formulas in HTTP Header	<p>Optional. Select this checkbox if the named credential's URL doesn't include a region. Then, after you save the named credential, add the Snowflake account as a custom HTTP header. Enter <code>Snowflake-Account</code> as the custom header's name, and then enter the Snowflake account locator, for example <code>xy12345</code>, as the custom header's value.</p> <p>For more information about how to create a custom header, see Custom Headers for Credentials. For more information about how to find the Snowflake account locator, see Account Identifiers in the Snowflake documentation.</p>
Outbound Network Connection	Optional. If your Snowflake account uses AWS PrivateLink to bypass the public internet, select the Private Connect outbound connection that you created to access Snowflake. All callouts from Salesforce that use this named credential are routed through the private connection.

4. Save your changes.

Use the named credential that captures the authentication configuration in external credentials to:

- Authenticate Salesforce users against Snowflake.
- Provide limited access to Snowflake resources.
- Periodically refresh access tokens.

See Also

[Named Credentials and External Credentials](#)

Create an External Data Source for Salesforce Connect SQL Adapter for Snowflake

Connect your Salesforce org to access Snowflake's interactive query capabilities.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit external data sources: Customize Application

1. From Setup, in the Quick Find box, enter *External Data Sources*, and then select **External Data Sources**.
2. Click **New External Data Source**, or click **Edit** to modify an existing external data source.
3. Complete these fields.

Field	Description
External Data Source	A user-friendly name for the external data source. The label is displayed in the Salesforce user interface.
Name	A unique identifier that's used to refer to this external data source definition through the API.
Type	Select SQL .
Provider	Select Snowflake .
Named Credential	Enter the named credential URL that you defined for the Snowflake data source.
Connection Timeout (Seconds)	Number of seconds to wait for a response from the Snowflake external system before timing out. By default, the value is set to the maximum of 120 seconds.
Writable External Objects	Select this option only if you want to create, edit, and delete data that's stored in Snowflake. By default, external objects are read only. Snowflake views are always read only, even if you enable this option for other data stored in Snowflake.

Field	Description
Server Driven Pagination	<p>It's common for Salesforce Connect queries of external data to have a large result set that's broken into smaller batches or pages. By default, the Salesforce Connect SQL adapter for Snowflake uses server-driven paging, meaning that Snowflake controls the paging behavior.</p> <p>To have the Salesforce Connect SQL adapter control the paging behavior (client-driven), deselect this checkbox. See Client-Driven and Server-Driven Paging in the Salesforce Connect SQL Adapter for Snowflake.</p>

- Save your changes.

See Also

- [Work with External Data Sources](#)
- [Salesforce Connect SQL Adapter for Snowflake](#)

Validate and Sync External Data Source Configured for Snowflake

After you create an external data source for Snowflake, synchronize it to map its data to external objects in your Salesforce org.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise, Performance, and Unlimited** Editions

Files Connect for cloud-based external data sources is available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Federated Search is available in: **Enterprise, Professional, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create an external object from an external data source: Customize Application

- From Setup, in the Quick Find box, enter *External Data Sources*, and then select **External Data Sources**.
- Click the name of the external data source that you created for the Snowflake external system.
- Click **Validate and Sync**.
- Select the Snowflake database and schema that contains the tables you want to sync.
- Select the data that you want to sync. The Salesforce Connect SQL adapter for Snowflake supports:
 - Tables

- Dynamic tables
 - Views
6. Choose whether to run the sync as an asynchronous background job.
7. Click **Sync**.

 **Note** Make sure that the external object field names exactly match the table column names in Snowflake. Otherwise, queries to Snowflake cause a runtime error.

After the external object and its fields are created, you must provide additional configuration so Salesforce can access Snowflake to run queries and respect settings configured by the Snowflake administrator. See [Manage Qualifiers for Salesforce Connect Adapter SQL for Snowflake](#).

See Also

- [Work with External Data Sources](#)
[Salesforce Connect SQL Adapter for Snowflake](#)

[Manage Qualifiers for Salesforce Connect Adapter SQL for Snowflake](#)

Tables and databases in Snowflake contain the metadata definitions for the underlying source data schema. To be able to make queries to Snowflake, use qualifiers to specify the key columns that identify records in the Snowflake data source.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

When you query a Snowflake table, the data catalog and database that you configured determine the table location. To provide additional configuration:

1. From Setup, in the Quick Find box, enter *External Data Sources*, and then select **External Data Sources**.
2. Select the external data source to edit.
3. Under External Objects, select the synced external object to which you want to add qualifiers, and then click **Manage Qualifiers**.

Field	Description
Database	The database that contains the table synced to this object. The database is auto-populated when the external data source is synced.
Schema	The schema of the database that contains the table synced to this object.

Field	Description
	The schema is auto-populated when the external data source is synced.
Key Column Names	<p>Specify the column names that represent a unique key for a row in the synced Snowflake table. The Salesforce Connect SQL adapter uses these column values to build an external ID for external object records in Salesforce.</p> <p>Composite keys can be represented as a comma-delimited list. For example, <code>ORDERID, PRODUCTID</code>.</p> <p>Keep these considerations in mind when you specify key columns.</p> <ul style="list-style-type: none"> • Don't use columns that aren't unique, such as BOOLEAN columns. • Ensure that the combination of key column values represents a unique value in the table.

4. Save your changes.

See Also

[Salesforce Connect SQL Adapter for Snowflake](#)

Considerations for Salesforce Connect Adapters for SQL

Understand the special behaviors, limits, and recommendations for using the Salesforce Connect adapters for SQL.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- Including `COUNT` and other aggregation projections in the same SOQL query isn't supported for the Salesforce Connect SQL adapters for Amazon Athena and Snowflake.
- If you plan to write Apex code with custom SOQL queries or Apex SOQL for-loops against data from a Snowflake external data source, we recommend using client-driven pagination. With these types of queries, client-driven pagination ensures that Snowflake returns only page sizes of up to 2,000 rows.

[Client-Driven and Server-Driven Paging in the Salesforce Connect SQL Adapter for Snowflake](#)

It's common for Salesforce Connect queries of external data to have a large result set that's broken into smaller batches or pages. You decide whether to have the paging behavior controlled by the external system (server-driven) or by the SQL adapter for Salesforce Connect (client-driven).

See Also

- [Salesforce Connect](#)
- [Salesforce Connect SQL Adapter for Snowflake](#)
- [Salesforce Connect SQL Adapter for Amazon Athena](#)

Client-Driven and Server-Driven Paging in the Salesforce Connect SQL Adapter for Snowflake

It's common for Salesforce Connect queries of external data to have a large result set that's broken into smaller batches or pages. You decide whether to have the paging behavior controlled by the external system (server-driven) or by the SQL adapter for Salesforce Connect (client-driven).

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Server-Driven Pagination

To align with Snowflake's recommendations, the Salesforce Connect SQL adapter uses server-driven paging by default, and Snowflake determines the page sizes and batch boundaries. With server-driven paging enabled, Salesforce ignores the requested batch sizes, including the batch size specified in REST and SOAP API calls and scope in Batch Apex. If you plan to use these features with external objects from Snowflake, refer to the Client-Driven Pagination section.

Server-driven pagination optimizes the external system's performance and improves the load times for external objects in your org. Also, the external data can change while your users or the Lightning Platform are paging through the result set. Typically, server-driven paging adjusts batch boundaries to accommodate changing external data more effectively than client-driven paging.

Client-Driven Pagination

The Server Driven Pagination field on the external data source specifies whether to use client-driven or server-driven paging. When server-driven paging is disabled on a Snowflake external data source, the requests use the `$limit` and `$offset` system query options to page through the result set.

Client-driven paging is useful if you plan to write Apex code with custom SOQL queries or Apex SOQL for-loops against data from a Snowflake external data source. With these types of queries, client-driven pagination ensures that Snowflake returns only page sizes of up to 2,000 rows.

If your implementation requires iterating over large datasets in Apex on a row-by-row basis, we recommend that you limit the datasets by adding filters to your query. Snowflake can contain billions of

rows, and large queries can consume significant resources.

See Also

[Salesforce Connect SQL Adapter for Snowflake](#)

Access External Data with the Salesforce Connect Adapter for GraphQL

Connect your users to external data sources that expose their capabilities via GraphQL.

To know the GraphQL schema specifications and naming conventions that must be adhered to for the Salesforce Connect adapter for GraphQL, see [Understand GraphQL Schema Requirements](#).

[Salesforce Connect Adapter for GraphQL](#)

GraphQL is a standard query language that allows applications to flexibly access required data and provides a modern way to integrate applications. Unlike traditional REST APIs, GraphQL provides granularity in specifying fields and relationships and is more efficient.

[Set Up Salesforce Connect to Access External Data Exposed via GraphQL](#)

Provide users access to external databases managed via GraphQL and integrate data in Salesforce so that they have a complete view of the business.

[Manage Qualifiers and External IDs for Salesforce Connect Adapter for GraphQL](#)

When you sync a database table to create a Salesforce external object, a table qualifier is automatically generated. The qualifier constructs an external ID for the external object based on the global id field in the GraphQL schema. You can also configure the external ID based on any of the ID field types in the GraphQL schema.

[Considerations for Salesforce Connect Adapters for GraphQL](#)

Understand the special behaviors, limits, and recommendations for using the Salesforce Connect adapter for Amazon GraphQL.

Salesforce Connect Adapter for GraphQL

GraphQL is a standard query language that allows applications to flexibly access required data and provides a modern way to integrate applications. Unlike traditional REST APIs, GraphQL provides granularity in specifying fields and relationships and is more efficient.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

The Salesforce Connect adapter for GraphQL includes special extensions for AWS AppSync and provides

seamless access to Amazon RDS. AWS AppSync hosts the GraphQL endpoint, accepts, and validates GraphQL queries. The GraphQL resolver interprets GraphQL queries into SQL queries and executes them against an Amazon RDS-hosted database. You can then create an external data source with the endpoint URL set to AppSync GraphQL API and provide access to AWS data source from Salesforce.

[Map Salesforce External Object Fields to GraphQL Types](#)

When you map external object fields to GraphQL scalar types, you may want to optimize the type mapping for a better experience in Salesforce. For example, a String in the GraphQL schema can be mapped to a Picklist in Salesforce to improve data quality.

See Also

[Salesforce Connect](#)

[Set Up Salesforce Connect to Access External Data Exposed via GraphQL](#)

[Manage Qualifiers and External IDs for Salesforce Connect Adapter for GraphQL](#)

[Considerations for Salesforce Connect Adapters for GraphQL](#)

Map Salesforce External Object Fields to GraphQL Types

When you map external object fields to GraphQL scalar types, you may want to optimize the type mapping for a better experience in Salesforce. For example, a String in the GraphQL schema can be mapped to a Picklist in Salesforce to improve data quality.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise, Performance, and Unlimited** Editions

GraphQL Scalar Type	Salesforce External Object Field Type
ID	Text
String	Text
Int	Number
Float	Number
Boolean	Checkbox
Enum	Picklist
Enum (List)	Picklist (Multi-select)

 **Note** To map Enum and Enum (List) GraphQL types to picklist or multi-select picklist field types in Salesforce, your resolver must include code that converts enum values to match the values in your

database.

Salesforce Connect adapter for GraphQL supports these type conversions.

- String (ISO 8601 Date) GraphQL type to Date Salesforce field type.
- String (ISO 8601 Date/Time) GraphQL type to Date/Time Salesforce field type.
- Int (epoch timestamp) GraphQL type to Date/Time Salesforce field type.
- Customized GraphQL scalar type to Text Salesforce field type.
- String GraphQL type to Picklist or Picklist (Multi-Select) Salesforce field types.

Mapping isn't supported for custom GraphQL object types.

Mapping for AWS AppSync

This table lists the mapping of AWS AppSync scalars to external object field types in Salesforce.

AWS AppSync Scalar Type	Salesforce External Object Field Type
AWSDate	Date
AWSTime	Time
AWSDateTime	Date/Time
AWSTimestamp	Number
AWSEmail	Email
AWSJSON	Text Area (Long)
AWSPhone	Phone
AWSURL	URL
AWSIPAddress	Text

See Also

- [Custom Field Types](#)
- [Salesforce Connect Adapter for GraphQL](#)

Set Up Salesforce Connect to Access External Data Exposed via GraphQL

Provide users access to external databases managed via GraphQL and integrate data in Salesforce so that they have a complete view of the business.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external

objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit external data sources:	Customize Application
To create and edit external objects:	Customize Application
To view named credentials:	View Setup and Configuration
To create, edit, or delete named credentials:	Customize Application
To create and edit custom fields:	Customize Application
To edit permission sets and user profiles:	Manage Profiles and Permission Sets
To edit another user's authentication settings for external systems:	Manage Users

Before you begin, review GraphQL and AWS AppSync resources:

- To know about GraphQL and explore how it works, see [Introduction to GraphQL](#).
- To know about AWS AppSync, see [What is AWS AppSync?](#)

If you're using AWS AppSync to enable GraphQL API implementation and access Amazon RDS-hosted databases, work with your Amazon administrator to configure AWS AppSync using the setup template.

[Define a Named Credential for Salesforce Connect Adapter for GraphQL](#)

Create a named credential that specifies the endpoint URL of a GraphQL API and an external credential to provide the required authentication parameters.

[Create an External Data Source for Salesforce Connect Adapter for GraphQL](#)

Connect your Salesforce org to data that's stored in external databases managed via GraphQL.

[Sync an External Data Source for Salesforce Connect Adapter for GraphQL](#)

After you create an external data source, synchronize it to map the database tables with external objects in your Salesforce org.

See Also

[Salesforce Connect](#)

[Salesforce Connect Adapter for GraphQL](#)

[Manage Qualifiers and External IDs for Salesforce Connect Adapter for GraphQL](#)

[Considerations for Salesforce Connect Adapters for GraphQL](#)

Define a Named Credential for Salesforce Connect Adapter for GraphQL

Create a named credential that specifies the endpoint URL of a GraphQL API and an external credential to provide the required authentication parameters.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS NEEDED

To view named credentials: View Setup and Configuration

To create, edit, or delete named credentials: Customize Applications

To create an external credential that uses AWS Signature Version 4 authentication protocol, see [Create and Edit an AWS Signature v4 External Credential](#). To create a named credential and link it to the external credential, see [Create and Edit a Named Credential](#).

If you're creating a named credential for GraphQL API URL hosted by AppSync, URL format would be

`https://<unique identifier>.appsync-api.REGION.amazonaws.com/graphql`

For example, if the AWS region is `us-west-2`, the URL would be `https://<unique identifier>.appsync-api.us-west-2.amazonaws.com/graphql`

To get the GraphQL API URL hosted by AppSync, go to AWS AppSync Console > APIs and select the GraphQL API to use. Under Settings > API Details, copy the API URL.

See Also

[Named Credentials](#)

[Salesforce Connect Adapter for GraphQL](#)

Create an External Data Source for Salesforce Connect Adapter for GraphQL

Connect your Salesforce org to data that's stored in external databases managed via GraphQL.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit external data sources: Customize Application

1. From Setup, enter *External Data Sources* in the Quick Find box, then select **External Data Sources**.
2. Click **New External Data Source**, or click **Edit** to modify an existing external data source.
3. Complete the fields.

Field	Description
External Data Source	A user-friendly name for the external data source. The label is displayed in the Salesforce user interface.
Name	A unique identifier that's used to refer to this external data source definition through the API.
Type	Select GraphQL .
Named Credential	<p>Enter the named credential URL you defined for GraphQL API.</p> <p>To access the external system, Salesforce Connect uses the authentication settings that are defined in the named credential.</p>
Connection Timeout	Number of seconds to wait for a response from the external system before timing out. By default, the value is set to the maximum of 120 seconds.
Writable External Objects	Select this option only if you want to create, edit, and delete data that's stored in the external data source. By default, external objects are read only.

4. Click **Save**.

See Also

[Work with External Data Sources](#)
[Salesforce Connect Adapter for GraphQL](#)

Sync an External Data Source for Salesforce Connect Adapter for GraphQL

After you create an external data source, synchronize it to map the database tables with external objects in your Salesforce org.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer Edition**

Available for an extra cost in: **Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit external data sources: Customize Application

1. From Setup, enter *External Data Sources* in the Quick Find box, then select **External Data Sources**.
2. Click the name of the external data source you created for the database managed via GraphQL.
3. Click **Validate and Sync**.
4. Select the database tables and click **Sync** to create Salesforce external objects.

After an external object is created, a table qualifier is automatically created based on the globally unique `id` field in the GraphQL schema. See [Manage Qualifiers and External IDs](#).

[Manage Qualifiers and External IDs for Salesforce Connect Adapter for GraphQL](#)

When you sync a database table to create a Salesforce external object, a table qualifier is automatically generated. The qualifier constructs an external ID for the external object based on the global id field in the GraphQL schema. You can also configure the external ID based on any of the ID field types in the GraphQL schema.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer Edition**

Available for an extra cost in: **Enterprise, Performance, and Unlimited** Editions

[Qualifier Options for Salesforce Connect Adapter for GraphQL](#)

We recommend you use the default global id in the GraphQL schema to construct the external ID of an external object.

See Also

[Salesforce Connect Adapter for GraphQL](#)

[Qualifier Options for Salesforce Connect Adapter for GraphQL](#)

We recommend you use the default global id in the GraphQL schema to construct the external ID of an external object.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

```
TableQualifier ::=  
    "tableName": "physicalTableName" |  
    "keyColumns": [ "keyColumnName"(, "keyColumnName")*] ||  
    "columns": {NamedColumnQualifier(, NamedColumnQualifier)*}  
  
NamedColumnQualifier ::=  
    "columnName": {ColumnQualifier(, ColumnQualifier)*}  
  
ColumnQualifier ::=  
    "virtual": (true | false) |  
    "values": [ValueDefinition(, ValueDefinition)*]  
  
ValueDefinition ::=  
    {ValueQualifier, ( ValueQualifier)*}  
  
ValueQualifier ::=  
    "definition": "functionDefinition" |  
    "columnOrder": [ "columnName"(, "columnName")*]
```

- **TableQualifier**: Specific set of properties that are used to identify a table.
- **NamedColumnQualifier**: Default value for the **columnName** is **NamedColumnQualifier**'s column name. If the **ColumnQualifier** is virtual, **columnName** is not specified.
- **ColumnQualifier**
 - **virtual** : If true, the field doesn't physically exist in the external data source. Default value is *false*. Virtual fields are defined using the **values** formula.
 - **values** : A list of candidate Salesforce formula functions that derive the field value from other fields.
- **ValueDefinition** : Contains an array of value qualifiers.
- **ValueQualifier**
 - **definition** : A candidate value defined using Salesforce formula functions. A formula function can be a constant expression or can depend on one or more field values.
 - **columnOrder** : The lexical field order of the derived field value. The lexical order determines the field sort order in SOQL.

 **Example** This is a sample table qualifier generated for an external ID based on the **id** field of the

MyOrder table.

```
{  
    "columns": {  
        "ExternalId": {  
            "virtual": true,  
            "values": [  
                {  
                    "definition": "SUBSTITUTE(LEFT(id,LEN(id)-17),\"\\\\\\-\\",  
                    "\")"  
                }  
            ]  
        },  
        "id": {  
            "values": [  
                {  
                    "definition": "\"Postgres_MyOrder-\"&SUBSTITUTE(Extern  
alId,\"-\",\"\\\\\\-\\")"  
                }  
            ]  
        }  
    }  
}
```

See Also

[Manage Qualifiers and External IDs for Salesforce Connect Adapter for GraphQL](#)

Considerations for Salesforce Connect Adapters for GraphQL

Understand the special behaviors, limits, and recommendations for using the Salesforce Connect adapter for Amazon GraphQL.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

- To support query pagination, the GraphQL schema must include the `first` field of type `int` and `after` field of type `string`. See [Understand GraphQL Schema Requirements](#) and [GraphQL: Pagination](#).

[Understand GraphQL Schema Requirements](#)

The Salesforce Connect adapter for GraphQL requires that the GraphQL schema adhere to a specific

format.

See Also

[Salesforce Connect Adapter for GraphQL](#)

[Set Up Salesforce Connect to Access External Data Exposed via GraphQL](#)

Understand GraphQL Schema Requirements

The Salesforce Connect adapter for GraphQL requires that the GraphQL schema adhere to a specific format.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience (not for high-data-volume external objects)

Available in: **Developer** Edition

Available for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

This is a representative schema to understand how to construct a schema that works with Salesforce Connect. Notations to understand the representative schema:

- **ObjName**: Denotes the name of a GraphQL object.
- **Bold text**: Represents the pattern to follow for each object.
- **Non-bold text**: Represents common code, not specific to the objects.

```
type Query {  
    objName(  
        limit: Int,  
        offset: Int,  
        orderBy: [ObjName_OrderByInput],  
        where: ObjName_FilterInput,  
        first: Int,  
        after: String  
    ): ObjName_Connection  
        <additional object queries...>  
        node(id: ID!): Node  
    }  
  
type Mutation {  
    create_ObjName(input: ObjName_CreateInput!): ObjName  
        <additional create mutations>  
        ...  
}
```

```
delete_ObjName(id: ID!): ObjName
<additional delete mutations>
...
update_ObjName(input: ObjName_UpdateInput!): ObjName
<additional update mutations>
...
}

type ObjName implements Node {
  id: ID!
  <fieldName: fieldType>
  ...
}

type ObjName_Connection {
  edges: [ObjName_Edge]
  pageInfo: PageInfo!
}

input ObjName_CreateInput {
  <fieldName: fieldType>
  ...
}

input ObjName_UpdateInput {
  id: ID!
  <fieldName: fieldType>
  ...
}

type ObjName_Edge {
  cursor: String!
  node: ObjName
}

input ObjName_FilterInput {
  and: [ObjName_FilterInput]
  not: ObjName_FilterInput
  or: [ObjName_FilterInput]
  id: IDOperator
  <fieldName: fieldTypeOperator>
  ...
}
```

```
}

input ObjName_OrderByInput {
    id: OrderByClause
    <fieldName >: OrderByClause
    ...
}

interface Node {
    id: ID!
}

enum NullsOrder {
    NULLS_FIRST
    NULLS_LAST
}

input OrderByClause {
    direction: Direction
    nulls: NullsOrder
}

type PageInfo {
    endCursor: String
    hasNextPage: Boolean!
    hasPreviousPage: Boolean!
    startCursor: String
}

input StringOperator {
    eq: String
    gt: String
    ge: String
    in: [String]
    like: String
    lt: String
    le: String
    ne: String
    nin: [String]
}

input AWSDateOperator {
    eq: AWSDate
    gt: AWSDate
}
```

```
ge: AWSDate
in: [AWSDate]
like: AWSDate
lt: AWSDate
le: AWSDate
ne: AWSDate
nin: [AWSDate]
}

input AWSDateTimeOperator {
    eq: AWSDateTime
    gt: AWSDateTime
    ge: AWSDateTime
    in: [AWSDateTime]
    like: AWSDateTime
    lt: AWSDateTime
    le: AWSDateTime
    ne: AWSDateTime
    nin: [AWSDateTime]
}

input AWSJSONOperator {
    eq: AWSJSON
    gt: AWSJSON
    ge: AWSJSON
    in: [AWSJSON]
    like: AWSJSON
    lt: AWSJSON
    le: AWSJSON
    ne: AWSJSON
    nin: [AWSJSON]
}

input AWSTimeOperator {
    eq: AWSTime
    gt: AWSTime
    ge: AWSTime
    in: [AWSTime]
    like: AWSTime
    lt: AWSTime
    le: AWSTime
    ne: AWSTime
    nin: [AWSTime]
}
```

```
input BooleanOperator {  
    eq: Boolean  
    gt: Boolean  
    gebolded: Boolean  
    in: [Boolean]  
    like: Boolean  
    lt: Boolean  
    le: Boolean  
    ne: Boolean  
    nin: [Boolean]  
}  
  
enum Direction {  
    ASC  
    DESC  
}  
  
input FloatOperator {  
    eq: Float  
    gt: Float  
    ge: Float  
    in: [Float]  
    like: Float  
    lt: Float  
    le: Float  
    ne: Float  
    nin: [Float]  
}  
  
input IDOperator {  
    eq: ID  
    gt: ID  
    ge: ID  
    in: [ID]  
    like: ID  
    lt: ID  
    le: ID  
    ne: ID  
    nin: [ID]  
}  
  
input IntOperator {  
    eq: Int
```

```
gt: Int  
ge: Int  
in: [Int]  
like: Int  
lt: Int  
le: Int  
ne: Int  
nin: [Int]  
}
```

Each object must have an `ID!` field that's globally unique and this global ID must be same across all the objects. If you have multiple fields of type `ID!` and there isn't field with `id` as the field name, use `keyColumns` qualifier to specify the global ID. See [Qualifier Options](#).

When an external lookup relationship represents the same lookup relationship that exists between GraphQL objects, keep in mind these considerations. The external ID field on the parent external object must match the format of the external lookup field of the child external object. In case they don't match, use qualifiers to manage one of the following:

- Manipulate the parent external object's external ID to match the child external lookup.
- Create a virtual column in the child external object to match the parent external ID.

It's important that the GraphQL global ID and Salesforce external ID can be computed from one another. A possible way to structure a global ID is to use a prefix to namespace the ID and keep it unique. If the primary key of the object is composed of multiple key fields, use a delimiter to separate the prefix and the ID. If any of the key fields' value contains the delimiter, make sure to escape the delimiter.

In this example, `ObjName|` is the prefix and `-` is the delimiter. Using `|` after the `ObjName` helps to parse the prefix from the key composition.

```
id: ObjName|KEY1\|-123-KEY2\|-456  
key1: KEY1-123  
key2: KEY2-456
```

See Also

[GraphQL: Schemas and Types](#)

[Salesforce Connect Adapter for GraphQL](#)

Access External Data with the Salesforce Connect Adaptor for Salesforce Archive

Connect your users to data that's stored in a Salesforce Archive data store.

REQUIRED EDITIONS

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions. Requires the Salesforce Archive license.

 **Note** This content relates to Salesforce Archive. Read about Archive in the [Own from Salesforce Knowledge Base](#).

Before you begin:

Make sure that your archive store contains data. If your archive store is empty or not yet provisioned for your org, errors can occur with the Salesforce Connect setup process.

To set up Salesforce Connect with the Salesforce Archive adaptor, complete these high-level steps.

1. [Set up your Salesforce Archive data store as an external data source](#).
2. Activate the objects that you want to access in your data store.
 - a. From the Salesforce Archive app, click the **Data Stores** tab.
 - b. In the dropdown menu for your data store, click **Configure Objects**.
 - c. In the Activated column, select the objects that you want to access.
 - d. Click **Save Settings**.
3. Create external objects in Salesforce to represent the data in your data store. Complete this step using one of two methods.
 - [Validate and sync](#) your external data source (recommended). This method automatically creates external objects and maps them to the data in your archive data store.
 - [Define external objects manually](#). Use this method only if you want to customize the object names and manually create the object fields.
4. View the external objects.

To view records on the external objects, you can:

- [Create custom tabs](#). This method displays records directly within the Salesforce UI.
- [Query the external objects using Salesforce Object Query Language \(SOQL\)](#).

[Define an External Data Source for Salesforce Connect–Salesforce Archive](#)

Give users access to the data in your Salesforce Archive data store.

[Use Custom Tabs to View Data in a Salesforce Archive Data Store](#)

To view the external objects that you synced with your Salesforce Archive data store, you can create custom object tabs.

Define an External Data Source for Salesforce Connect–Salesforce Archive

Give users access to the data in your Salesforce Archive data store.

REQUIRED EDITIONS

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions. Requires the Salesforce

Archive license.

USER PERMISSIONS NEEDED

To create and edit external data sources: Customize Application

To define your Salesforce Archive data store as an external data source:

1. From Setup, in the Quick Find box, enter *External Data Sources*, and then select **External Data Sources**.
2. Click **New External Data Source**, or click **Edit** to modify an existing external data source.
3. Complete the fields.

FIELD	DESCRIPTION
External Data Source	A user-friendly name for the external data source. The label is displayed in the Salesforce user interface, such as in list views.
Name	<p>A unique identifier that's used to refer to this external data source definition through the API. This field auto-populates after you enter an External Data Source value.</p> <p>The name can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.</p>
Type	Select Privacy Center or Archive Data Store .
Data Store Type	Select Salesforce Archive Store .

4. Click **Save**.

Use Custom Tabs to View Data in a Salesforce Archive Data Store

To view the external objects that you synced with your Salesforce Archive data store, you can create custom object tabs.

REQUIRED EDITIONS

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions. Requires the Salesforce Archive license.

USER PERMISSIONS NEEDED

To create and edit custom tabs:

Customize Application

Complete this task after you've created external objects to represent the data in your archive data store. For more information, see [Access External Data with the Salesforce Connect Adaptor for Salesforce Archive](#).

1. [Create a custom object tab](#) for the external object that you want to view.
2. On the tab you created, replace the Display URL field with the object fields that you want to view.
 - a. While viewing the tab, click the list view controls button.
 - b. Click **Select Fields to Display**.
 - c. Hide the Display URL field by removing it from the Visible Fields column.
 - d. Add the fields that you want to view to the Visible Fields column.
 - e. Save your changes.
3. Update the page layout for the external object.
 - a. From Setup, in the Quick Find box, enter *Object Manager*, and then select **Object Manager**.
 - b. Find and select the external object.
 - c. In the Page Layouts section, select the layout that you want to edit. For newly created external objects, only one layout is available to select.
 - d. Add the fields that you want to view to the page layout. Depending on the editor version you're using, see [Customize Page Layouts with the Enhanced Page Layout Editor](#) or [Customize Page Layouts with the Original Page Layout Editor](#).

You can now view the external object data in the custom tab you created.

Access External Data with the Salesforce Connect Adapter for Privacy Center

Connect your users to data that's stored in a Privacy Center retention store.

REQUIRED EDITIONS

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions. Requires the Privacy Center license.

Before you begin:

- Make sure that your Privacy Center retention store contains data. If your retention store is empty or not yet provisioned for your org, errors can occur with the Salesforce Connect setup process.
- External objects are used to represent the data in your Privacy Center retention store. The external object fields don't appear in global searches.

To set up Salesforce Connect with the Privacy Center adapter, complete these high-level steps.

1. [Set up your Privacy Center retention store as an external data source](#).
2. Activate the objects that you want to access in your data store.

- a. From the Privacy Center app, click the **Data Stores** tab.
 - b. In the dropdown menu for your data store, click **Configure Objects**.
 - c. In the Activated column, select the objects that you want to access.
 - d. Click **Save Settings**.
3. Create external objects in Salesforce to represent the data in your data store. Complete this step using one of two methods.
- [Validate and sync](#) your external data source (recommended). This method automatically creates external objects and maps them to the data in your retention store.
 - [Define external objects manually](#). Use this method only if you want to customize the object names and manually create the object fields.
4. View the retained data using any of these methods.
- [View Records with Custom Tabs](#).
 - [View Records with Related Lists](#).
 - [Query the external objects using Salesforce Object Query Language \(SOQL\)](#).
 - [View Files and Attachments](#).

[Define an External Data Source for Salesforce Connect–Privacy Center](#)

Give users access to the data in your Privacy Center retention store.

[View Records with Custom Tabs](#)

To view the external objects that you synced with your Privacy Center retention store, you can create custom object tabs.

[View Records with Related Lists](#)

Use related lists to view the external objects that you synced with your Privacy Center retention store.

[View Files and Attachments](#)

View files and attachments in your Privacy Center retention store by adding a component to the associated record page.

See Also

[Knowledge Article: Use a Two-Phase Query to Access High-Volume Privacy Center Retained Data](#)

Define an External Data Source for Salesforce Connect–Privacy Center

Give users access to the data in your Privacy Center retention store.

REQUIRED EDITIONS

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions. Requires the Privacy Center license.

USER PERMISSIONS NEEDED

To create and edit external data sources: [Customize Application](#)

To define your Privacy Center retention store as an external data source:

1. From Setup, in the Quick Find box, enter *External Data Sources*, and then select **External Data Sources**.
2. Click **New External Data Source**, or click **Edit** to modify an existing external data source.
3. Complete the fields. Note that the Data Store Type field is located in the Parameters section of the page.

FIELD	DESCRIPTION
External Data Source	A user-friendly name for the external data source. The label is displayed in the Salesforce user interface, such as in list views.
Name	<p>A unique identifier that's used to refer to this external data source definition through the API.</p> <p>The name can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.</p>
Type	Select Privacy Center or Archive Data Store .
Data Store Type	Select Privacy Center Retention Store .

4. Click **Save**.

Once your retention store is defined as an external data source, you're ready to continue with the [Salesforce Connect setup process](#).

See Also

[Access External Data with the Salesforce Connect Adapter for Privacy Center](#)

View Records with Custom Tabs

To view the external objects that you synced with your Privacy Center retention store, you can create custom object tabs.

REQUIRED EDITIONS

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions. Requires the Privacy Center license.

USER PERMISSIONS NEEDED

To create and edit custom tabs:

Customize Application

Complete this task after you've created external objects to represent the data in your retention store. For more information, see [Access External Data with the Salesforce Connect Adapter for Privacy Center](#).

1. [Create a custom object tab](#) for the external object that you want to view.
2. On the tab you created, replace the Display URL field with the object fields that you want to view.
 - a. While viewing the tab, click the list view controls button.
 - b. Click **Select Fields to Display**.
 - c. Hide the Display URL field by removing it from the Visible Fields column.
 - d. Add the fields that you want to view to the Visible Fields column.
 - e. Save your changes.
3. Update the page layout for the external object.
 - a. From Setup, in the Quick Find box, enter *Object Manager*, and then select **Object Manager**.
 - b. Find and select the external object.
 - c. In the Page Layouts section, select the layout that you want to edit. By default, only one layout is available to select when you create a new external object.
 - d. Add the fields that you want to view to the page layout. Depending on the editor version you're using, see [Customize Page Layouts with the Enhanced Page Layout Editor](#) or [Customize Page Layouts with the Original Page Layout Editor](#).

You can now view the external object data in the custom tab you created.

View Records with Related Lists

Use related lists to view the external objects that you synced with your Privacy Center retention store.

REQUIRED EDITIONS

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions. Requires the Privacy Center license.

To display an external object in a related list:

1. From Setup, in the Quick Find box, enter *External Data Sources*, and then select **External Data Sources**.
2. Select the data source associated with your Privacy Center retention store.
3. Under External Objects, find the object that you want to configure as a related list, and then click **Edit**.
4. Click **Fields and Relationships**.
5. Select a field relating to an object within Salesforce where you want the external object to be visible. For example, say that you have an external object for retained Cases, and you want it to appear as a related list on the Account object. In this step, you select the Account ID field because it relates Cases with Accounts.
6. Click **Edit**, and then click **Change Field Type**.
7. Click **Lookup Relationship**, then **Next**.
8. From the Related To dropdown menu, select the corresponding object in Salesforce. For example, if you selected the Account ID field in Step 5, then select Account.

9. Optionally, update the label for the field, and then click **Next**.
10. Choose which user profiles have field visibility, and then click **Next**.
11. Enter a name in the Related List Label field.
12. Select the page layouts that you want to display the list in.
To prevent slow performance, avoid showing the related list by default. For example, place the related list on a separate tab so that users access the retained data only when needed. This approach prevents automatic, large queries on every page load.
13. Save your work.

The external object is now available as a related list when you view the corresponding object in Salesforce.

View Files and Attachments

View files and attachments in your Privacy Center retention store by adding a component to the associated record page.

REQUIRED EDITIONS

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions. Requires the Privacy Center license.

USER PERMISSIONS NEEDED

To access the Retained Files component:	Manage Files or Attachments Download from Data Store, Manage Privacy Center Policies
---	--

To browse retained files and attachments for a given record, add the Retained Files component to the record page.

1. From the record page, click the gear icon, and then click **Edit Page**.
2. Drag and drop the **Retained Files** component onto the page canvas.
3. In the Variant field, make sure that **Privacy Center** is selected.
4. Optionally, select **Load files automatically** if you want files to load by default when you open the record page.
5. Save your work and exit the page editor.
The Retained Files component is now available on the record home page.
6. To download a file from the Retained Files component, click the dropdown arrow, and then click **View File**.

 **Note** Because certain special characters aren't supported, the downloaded file name can sometimes be modified.

Work with External Data Sources

An external data source specifies how to access an external system. Salesforce Connect uses external data sources to access data that's stored outside your Salesforce organization. Files Connect uses external data sources to access third-party content systems. External data sources have associated external objects, which your users and the Lightning platform use to interact with the external data and content.

[Define External Data Sources](#)

Create external data sources to connect to content and data that is stored outside your Salesforce org.

[Validate and Sync an External Data Source](#)

After you configure an external data source, synchronize it to map its tables with external objects in your Salesforce org. The content and data of external objects appear in federated search, together with your Salesforce content and data.

[Define External Objects](#)

Tables in external systems map to external objects in Salesforce, combining all your data and content for users in your org.

[Validate External Object Connections](#)

After you configure an external data source, run the validator tool on each external object to test and troubleshoot its connections. The tool tests for ID uniqueness and the ability to sort and filter results. Validator is available for external objects from Apex and Odata data sources.

[Grant Access to Authentication Settings for External Data Sources](#)

For external data sources that use per-user authentication, grant users access through permission sets and profiles, so users can set up and manage their own authentication settings for accessing the external system.

[Store Authentication Settings for External Systems](#)

You or your administrator can set up and manage your authentication settings for external systems. With valid authentication settings, you can access external systems from within your Salesforce org.

[External Object Relationships](#)

External objects support standard lookup relationships, which use the 18-character Salesforce record IDs to associate related records with each other. However, data that's stored outside your Salesforce org often doesn't contain those record IDs. Therefore, two special types of lookup relationships are available for external objects: external lookups and indirect lookups.

Define External Data Sources

Create external data sources to connect to content and data that is stored outside your Salesforce org.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise, Performance, and Unlimited** Editions

Files Connect for cloud-based external data sources is available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Federated Search is available in: **Enterprise, Professional, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create and edit an external data source: Customize Application

1. From Setup, in the Quick Find box, enter *External Data Sources*, and then select **External Data Sources**.
2. Click **New External Data Source**. To modify an existing external data source, click **Edit**.
3. Complete the steps for your type of external data source.
 - [Files Connect: Google Drive](#)
 - [Files Connect: SharePoint 2010 or 2013](#)
 - [Files Connect: SharePoint Online or OneDrive for Business](#)
 - [Files Connect: Box](#)
 - [Simple URL: Data from Another Web Domain](#)
 - [Salesforce Connect: OData 2.0](#)
 - [Salesforce Connect: OData 4.0](#)
 - [Salesforce Connect: Cross-Org](#)
 - [Salesforce Connect: Custom](#)
 - [Salesforce Connect adapter for Amazon DynamoDB](#)
 - [Salesforce Connect SQL adapter for Amazon Athena](#)
 - [Salesforce Connect SQL adapter for Snowflake](#)
 - [Federated Search](#)

See Also

[The Files Connect Setup Process](#)

[Salesforce Connect](#)

Validate and Sync an External Data Source

After you configure an external data source, synchronize it to map its tables with external objects in your Salesforce org. The content and data of external objects appear in federated search, together with your Salesforce content and data.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise, Performance, and Unlimited** Editions

Files Connect for cloud-based external data sources is available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Federated Search is available in: **Enterprise, Professional, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create an external object from an external data source: Customize Application

When you sync a large number of tables, you can select to execute the sync operation asynchronously as a background job and avoid connection timeout. You can view the list of sync operations performed as background jobs in the corresponding External Data Source page.

Note

- Syncing creates or overwrites Salesforce external objects that map to the external system's schema. Syncing doesn't copy any data into your Salesforce org or write data from your org to the external system.
- Syncing is a one-time process. If the external system's schema changes, the modifications aren't automatically synced to your Salesforce org. To reflect the changes in the external system, resync the objects. After resyncing, full field-level security to the external objects is granted to all users who have the same profile as the user who initiated the resync.
- Each org can have up to 200 external objects. External objects don't count toward the amount for custom objects. Syncing fails if it causes your org to exceed 200 external objects.
- For Salesforce Connect external data sources, make sure that you read and understand the sync considerations. See [Sync an External Data Source for Salesforce Connect](#).

1. From Setup, enter *External Data Sources* in the Quick Find box, then select **External Data Sources**.
2. Click the name of the external data source.
3. Click **Validate and Sync**, and confirm that the connection is successful.
4. Select tables and click **Sync** to do the following for each selected table.
 - Automatically create a Salesforce external object.
 - Automatically create a custom field for each table column that's compatible with a Salesforce metadata field type.

 **Example** The resulting external data source detail page includes a list of related external objects like this.

					Edit	Validate and Sync	Delete
External Objects							
Action	Label	Namespace Prefix	Description	Table Name			
Edit Del	xCustomers		Customers	Customers			
Edit Del	xEmployees		Employees	Employees			
Edit Del	xOrders		Orders	Orders			
Edit Del	xSuppliers		Suppliers	Suppliers			
Edit Del	xInvoices		Invoices	Invoices			
Edit Del	xProducts		Products	Products			

See Also

[The Files Connect Setup Process](#)

[Salesforce Connect](#)

Define External Objects

Tables in external systems map to external objects in Salesforce, combining all your data and content for users in your org.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise, Performance, and Unlimited** Editions

Files Connect for cloud-based external data sources is available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Federated Search is available in: **Enterprise, Professional, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To create or edit external objects:

[Customize Application](#)

External objects are similar to custom objects, except that they map to data that's stored outside your Salesforce org. Each external object relies on an external data source definition to connect with the external system's data. Each external object definition maps to a data table on the external system. Each of the external object's fields maps to a table column on the external system. External objects enable your users and the Lightning Platform to search and interact with the external data.

- Each org can have up to 200 external objects. External objects don't count toward the amount for

- custom objects.
- If the external system allows it, we recommend that you [sync](#) the external data source to automatically create related external objects. You can instead choose to manually define external objects to customize the external object names and manually create the custom fields.

To create or modify an external object:

- From Setup, enter *External Objects* in the Quick Find box, then select **External Objects**.
- Click **New External Object**, or click **Edit** to modify an existing external object.
- Enter the following:

Field	Description
Label	<p>A user-friendly name for the external object. The label is displayed in the Salesforce user interface, such as in list views.</p> <p>We recommend that you make object labels unique across all standard, custom, and external objects in the org.</p>
Plural Label	The plural name of the external object. If you create a tab for this object, this name is used for the tab.
Starts with vowel sound	If it's appropriate for your org's default language, select to precede your label with "an" instead of "a" for any automated messages.
Object Name	<p>A unique identifier used to refer to this external object definition when using the API. Object names must be unique across all standard, custom, and external objects in the org.</p> <p>The Object Name field can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.</p>
Description	An optional description of the external object. A meaningful description helps you distinguish among your external objects when you view them in a list.
Context-Sensitive Help Setting	<p>Defines what appears when users click Help for this Page from the external object record home (overview) and detail pages, as well as list views and related lists.</p> <p>We recommend that you select Open a window using a Visualforce page to display custom help that you create for your users.</p> <p>If you instead keep the default value, your users only see Salesforce Help, which doesn't provide any information about your external data.</p>

Field	Description
	This setting doesn't affect the Help & Training link at the top of each page, which always opens Salesforce Help.
Content Name	<p>Select the Visualforce page that best describes the data that's provided by this external object.</p> <p>This field is available only when you select Open a window using a Visualforce page.</p>
External Data Source	The external data source definition that contains the connection details you want to use for this external object.
Table Name	<p>Table in the external system that the external object maps to.</p> <p>For SharePoint, the table name must match the related scope name.</p>
Display URL Reference Field	<p>Available only for Salesforce Connect. The external object's Display URL standard field values are automatically generated from the external system. For example, with the OData 2.0 adapter for Salesforce Connect, the value is based on the <code>link href</code> that's defined on the OData producer.</p> <p>You can override the default values with the values of a custom field on the same external object. Select the field name, and make sure that the custom field's values are valid URLs.</p>
Allow Reports	Available only for Salesforce Connect.
Deployment Status	Indicates whether the external object is visible to other users.
Launch New Custom Tab Wizard after saving this external object	If selected, the custom tab wizard starts after you save the external object.
Allow Search	<p>If search is also enabled on the external data source, selecting this option lets users find the external object's records via SOSL and Salesforce global searches.</p> <p>By default, search is disabled for new external objects. However, you can validate and sync an external data source to automatically create external objects. Syncing always enables search on the external object when search is enabled on the external data source, and vice versa.</p> <p>However, syncing always overwrites the external object's search status to</p>

Field	Description
	match the search status of the external data source.

4. Click **Save**.
5. On the external object detail page, view and modify the external object's custom fields and relationships, page layouts, field sets, search layouts, and buttons and links.
 - To create field mappings or add fields to an external object, click **New** on the Custom Fields & Relationships related list.
 - To assign different page layouts by user profile, click **Page Layout Assignments**.

 **Tip** After you configure an external data source, run the validator tool on each external object to test and troubleshoot its connections. The tool tests for ID uniqueness and the ability to sort and filter results.

See Also

- [Set Up Salesforce Connect to Access External Data with OData Adapters](#)
- [Set Up Salesforce Connect to Access Data in Another Org with the Cross-Org Adapter](#)
- [Set Up Salesforce Connect to Access External Data with a Custom Adapter](#)
- [Deployment Status for Custom Objects and External Objects](#)

Validate External Object Connections

After you configure an external data source, run the validator tool on each external object to test and troubleshoot its connections. The tool tests for ID uniqueness and the ability to sort and filter results. Validator is available for external objects from Apex and Odata data sources.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Files Connect for cloud-based external data sources is available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Federated Search is available in: **Enterprise**, **Professional**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS NEEDED

To create or edit external objects:	Customize Application
-------------------------------------	-----------------------

1. From Setup, enter *External Data Source* in the Quick Find box, then select **External Data Source**.
2. Select the external data source that contains the external object you want to validate.

3. Next to the external object's name, click **Validate**.
4. Select each query type you want validator to run.
5. Click **Run Queries**.

The validator results are displayed. Queries that ran successfully are listed in the Passed Queries table. The Failed Queries table lists the unsuccessful queries with information to help with troubleshooting. Queries that weren't run because a prerequisite query was unsuccessful are in the Skipped Queries table.

See Also

[Define External Objects](#)

Grant Access to Authentication Settings for External Data Sources

For external data sources that use per-user authentication, grant users access through permission sets and profiles, so users can set up and manage their own authentication settings for accessing the external system.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Files Connect for cloud-based external data sources is available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Federated Search is available in: **Enterprise**, **Professional**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS NEEDED

To edit permission sets and user profiles: [Manage Profiles and Permission Sets](#)

1. From Setup, enter *Permission Sets* in the Quick Find box, then select **Permission Sets** or **Profiles**.
2. Click the name of the permission set or profile that you want to modify.
3. Do one of the following.
 - a. For a permission set, or for a profile in the enhanced profile user interface, click **External Data Source Access** in the Apps section. Then click **Edit**.
 - b. For a permission set, or for a profile in the enhanced profile user interface, click **External Data Source Access** in the Apps section. Then click **Edit**.
 - c. For a profile in the original profile user interface, click **Edit** in the Enabled External Data Source Access section.
4. Add the data sources that you want to enable.
5. Click **Save**.

See Also

[Store Authentication Settings for External Systems](#)[Set Up Salesforce Connect to Access Data in Another Org with the Cross-Org Adapter](#)[Set Up Salesforce Connect to Access External Data with OData Adapters](#)[Set Up Salesforce Connect to Access External Data with a Custom Adapter](#)

Store Authentication Settings for External Systems

You or your administrator can set up and manage your authentication settings for external systems. With valid authentication settings, you can access external systems from within your Salesforce org.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Named credentials are available in: All Editions.

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise, Performance, and Unlimited** Editions.

Files Connect for cloud-based external data sources is available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To store authentication settings for an external data source:	The data source enabled under External Data Source Access
To store authentication settings for a named credential:	The named credential enabled under Named Credential Access
To edit another user's authentication settings for external systems:	Manage Users

 **Note** All credentials stored within the NamedCredential, ExternalDataSource, and ExternalDataUserAuth entities are encrypted under a framework that is consistent with other encryption frameworks on the platform. Salesforce encrypts your credentials by auto-creating org-specific keys. Credentials encrypted using the previous encryption scheme were migrated to the new framework.

Your administrator defines external systems in external data sources and named credentials. External data sources specify how to access data or content that's stored outside your Salesforce org. Named credentials specify callout endpoints, which receive Web service callouts from your org.

Before you begin, your administrator:

- Sets up the external data source or named credential to use per-user authentication.

- Grants you access to the external data source or named credential.
 - Verifies that your org can connect to the external system.
 - Tells you the authentication settings to enter.
 - Sets up the community, if applicable, by using the Salesforce Tabs + Visualforce template.
- If the community is built with the Customer Service template, only your administrator can set up and manage your authentication settings for external systems. You can't complete these steps on your own.

If you don't see the expected settings or options, contact your administrator.

1. Access your authentication settings for external systems with one of these methods.

From within a community, if you have a community license, click **My Settings | Authentication Settings for External Systems**. Otherwise, from your personal settings, enter *Authentication*, then select **Authentication Settings for External Systems**.

From Salesforce, go to your personal settings and enter *Authentication* in the **Quick Find** box, then select **Authentication Settings for External Systems**.

2. Click **New** or **Edit**.

3. Complete the fields.

Field	Description
External System Definition	<p>If you're not sure which option to select, ask your administrator.</p> <ul style="list-style-type: none"> • External Data Source: Provides access to external objects, whose data is stored outside your Salesforce organization. • Named Credential: Enables your actions to trigger authenticated callouts to the endpoint that's specified in the named credential. <p>A named credential can handle the authentication for an external data source. In this scenario, your administrator instructs you to select Named Credential in this field to access external objects.</p>
External Data Source or Named Credential	Which field appears depends on what's selected for External System Definition . If you're not sure which option to select, ask your administrator. Your administrator can change the option labels to make them more relevant or easier to distinguish from each other.
User	Available only to administrators. Select the user whose authentication settings you're entering.

4. Select the authentication protocol that the external system requires. If you're not sure which option to select, ask your administrator.

If you select **Password Authentication**, enter your username and password for the external system.

If you select **OAuth 2.0**, complete these fields.

Field	Description
Authentication Provider	If you're not sure which option to select, ask your administrator. Your administrator can change the option labels to make them more relevant or easier to distinguish from each other.
Scope	If you're not sure what to enter, ask your administrator. Specifies the scope of permissions to request for the access token.
Start Authentication Flow on Save	To authenticate to the external system and obtain an OAuth token, select this checkbox. This authentication process is called an OAuth flow. When you click Save , the external system prompts you to log in. After successful login, the external system grants you an OAuth token for accessing its data from this org. Redo the OAuth flow when you need a new token—for example, if the token expires—or if you edit the Scope or Authentication Provider fields. When the token expires, the external system returns a 401 HTTP error status.

5. Click **Save**.

See Also

- [Define External Data Sources](#)
- [Grant Access to Authentication Settings for External Data Sources](#)
- [Named Credentials](#)
- [Grant Access to Authentication Settings for Legacy Named Credentials](#)
- [Add Tabs to Your Salesforce Tabs + Visualforce Site](#)
- [Personalize Your Salesforce Experience](#)

External Object Relationships

External objects support standard lookup relationships, which use the 18-character Salesforce record IDs to associate related records with each other. However, data that's stored outside your Salesforce org often doesn't contain those record IDs. Therefore, two special types of lookup relationships are available for external objects: external lookups and indirect lookups.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise, Performance, and Unlimited** Editions

Files Connect for cloud-based external data sources is available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Federated Search is available in: **Enterprise, Professional, Unlimited, and Developer** Editions

External lookups and indirect lookups compare a specific field's values on the parent object to the relationship field's values on the child object. When values match, the records are related to each other.

To create an external object relationship, create a custom field on the child object with one of the following field types. If the child is an external object, you can instead change the field type of an existing custom field to one of the following.

- Lookup Relationship
- External Lookup Relationship
- Indirect Lookup Relationship

This table summarizes the types of relationships that are available to external objects.

 **Note** Federated Search supports only external lookup relationships, and the Federated Search external object is always the parent.

Relationship	Allowed Child Objects	Allowed Parent Objects	Parent Field for Matching Records
Lookup	Standard	Standard	The 18-character Salesforce record ID
	Custom	Custom	
	External		
External lookup	Standard	External	The External ID standard field
	Custom		
	External		
Indirect lookup	External	Standard Custom	You select a custom field with the <code>External ID</code> and <code>Unique</code> attributes

Lookup Relationship Fields on External Objects

Use a lookup relationship when the external data includes a column that identifies related Salesforce

records by their 18-character IDs.

[External Lookup Relationship Fields on External Objects](#)

Use an external lookup relationship when the parent is an external object.

[Indirect Lookup Relationship Fields on External Objects](#)

Use an indirect lookup relationship when the external data doesn't include Salesforce record IDs.

See Also

[Create Custom Fields](#)

[Change the Data Type of a Custom Field](#)

[Considerations for Object Relationships](#)

[Include a Files Connect Data Source in Global Search](#)

[Object Relationships Overview](#)

Lookup Relationship Fields on External Objects

Use a lookup relationship when the external data includes a column that identifies related Salesforce records by their 18-character IDs.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise**, **Performance**, and **Unlimited** Editions

Files Connect for cloud-based external data sources is available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Federated Search is available in: **Enterprise**, **Professional**, **Unlimited**, and **Developer** Editions

A lookup relationship field links a child standard, custom, or external object to a parent standard or custom object. A user who's editing a child record can click the field's lookup icon to select a specific parent record, and a user who's viewing a parent record can view a related list of child records.

When you create a lookup relationship field on an external object, enter the External Column Name that contains the 18-character Salesforce IDs for identifying the parent records.



Example

- Account record (parent standard object) displays a related list of external SAP sales orders (child external object).
- Account record (parent standard object) displays a related list of support cases (child standard object).

See Also

- [External Object Relationships](#)
- [Create Custom Fields](#)
- [Considerations for Object Relationships](#)
- [Include a Files Connect Data Source in Global Search](#)

External Lookup Relationship Fields on External Objects

Use an external lookup relationship when the parent is an external object.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise, Performance, and Unlimited** Editions

Files Connect for cloud-based external data sources is available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Federated Search is available in: **Enterprise, Professional, Unlimited, and Developer** Editions

An external lookup relationship links a child standard, custom, or external object to a parent external object.

The values of the standard External ID field on the parent external object are matched against the values of the external lookup relationship field. For a child external object, the values of the external lookup relationship field come from the specified External Column Name.

When you sync two external system tables with a lookup relationship field, an external lookup relationship for the mapped external objects is automatically created. This auto mapping happens only if

- The parent and child objects are synced from the same data source.
- The parent object is already synced or the parent and child objects are in the same sync operation.

 **Note** For the OData 2.0 adapter, creating an external lookup relationship is supported only in the default Olingo library and not supported in OData4J.

Example

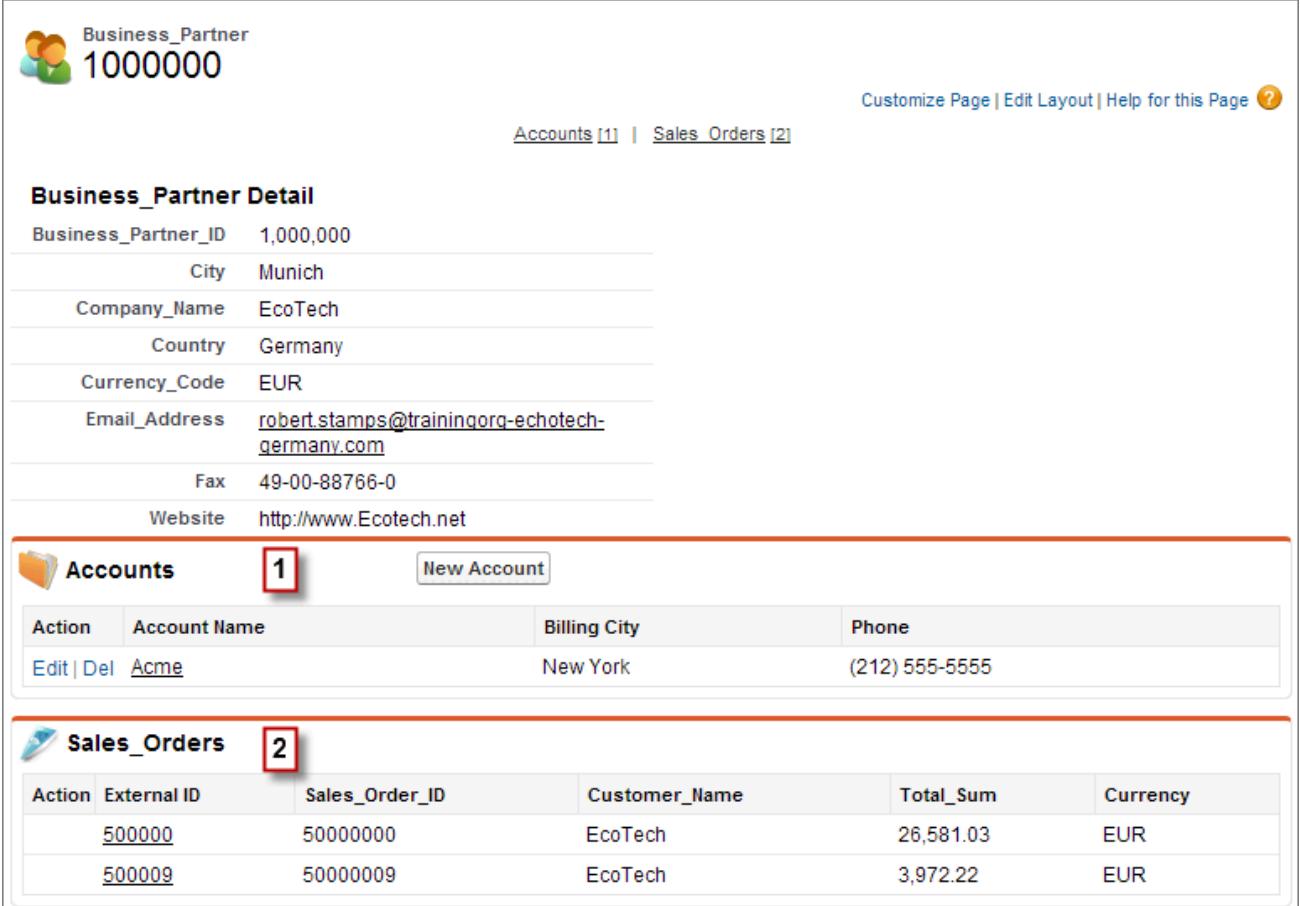
- External product catalog item (parent external object) displays a related list of support cases (child standard object).
- External customer (parent external object) displays a related list of external orders (child external object).

 **Example** For the cross-org adapter for Salesforce Connect, suppose that you store contacts and

accounts in the provider org. From the subscriber org, you want to view each account's related contacts. To do so, create an external lookup field on the subscriber org's Contact external object. Link that external lookup field to the subscriber org's Account external object. Then set up the page layouts for the Account external object to include a related list that displays the related Contact external object records.

 **Example** In this screenshot, a record detail page for the Business_Partner external object includes two related lists of child objects. This example shows how external lookup relationships and page layouts enable users to view related data from within and from outside their Salesforce org on a single page.

- Account standard object (1)
- Sales_Order external object (2)



The screenshot shows a Salesforce detail page for a Business_Partner record with ID 1000000. The page title is "Business_Partner Detail". At the top right are links for "Customize Page", "Edit Layout", and "Help for this Page". Below the title, there are two related lists:

- Accounts [1]**: A table with columns "Action", "Account Name", "Billing City", and "Phone". It shows one record for "Acme" with "New York" as the billing city and "(212) 555-5555" as the phone number.
- Sales_Orders [2]**: A table with columns "Action", "External ID", "Sales_Order_ID", "Customer_Name", "Total_Sum", and "Currency". It shows two records: one for "500000" with "EcoTech" as the customer and a total sum of "26,581.03" (EUR), and another for "500009" with "EcoTech" as the customer and a total sum of "3,972.22" (EUR).

See Also

- [External Object Relationships](#)
- [Create Custom Fields](#)
- [Change the Data Type of a Custom Field](#)
- [Considerations for Object Relationships](#)
- [Include a Files Connect Data Source in Global Search](#)

Indirect Lookup Relationship Fields on External Objects

Use an indirect lookup relationship when the external data doesn't include Salesforce record IDs.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Salesforce Connect is available in: **Developer** Edition and for an extra cost in: **Enterprise, Performance, and Unlimited** Editions

Files Connect for cloud-based external data sources is available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Federated Search is available in: **Enterprise, Professional, Unlimited, and Developer** Editions

An indirect lookup relationship links a child external object to a parent standard or custom object. When you create an indirect lookup relationship field on an external object, you specify the parent object field and the child object field to match against each other.

Specifically, you select a custom unique, external ID field on the parent object to match against the child's indirect lookup relationship field, whose values are determined by the specified External Column Name.

If the external system uses case-sensitive values in the specified External Column Name, make sure that the parent object field is also case-sensitive. When you define the parent object's custom field, select **External ID, Unique, and Treat "ABC" and "abc" as different values (case sensitive)**.

 **Note** Only objects that have a custom field with the `External ID` and `Unique` attributes are available as parent objects in indirect lookup relationships. If you don't see the desired object when you create an indirect lookup relationship field, add a custom unique, external ID field to that object.

 **Example**

- Account record (parent standard object) displays a related list of SAP sales orders (child external object) with matching customer IDs that aren't Salesforce IDs.
- Contact record (parent standard object) displays a related list of social media posts (child external object) with matching social media handles.

See Also

[External Object Relationships](#)

[Create Custom Fields](#)

[Change the Data Type of a Custom Field](#)

[Considerations for Object Relationships](#)

[Include a Files Connect Data Source in Global Search](#)

Connect Business Processes with Real-Time Events

Publish and subscribe to platform events to connect business processes in Salesforce and external sources through the exchange of real-time event data. Also, use event relays to integrate platform events and change data capture events with Amazon EventBridge.

Define and Manage Platform Events

Use platform events to connect business processes in Salesforce and external sources through the exchange of real-time event data. Platform events are secure and scalable. Define fields to customize your platform event data.

Event Relay

Integrate your real-time events with Amazon Web Services (AWS). Use Event Relay to send platform events and change data capture events from Salesforce to Amazon EventBridge.

Define and Manage Platform Events

Use platform events to connect business processes in Salesforce and external sources through the exchange of real-time event data. Platform events are secure and scalable. Define fields to customize your platform event data.

REQUIRED EDITIONS

Available in both Salesforce Classic and Lightning Experience

Available in: **Performance, Unlimited, Enterprise, and Developer Editions**

USER PERMISSIONS NEEDED

To create and edit platform event definitions:	Customize Application
--	-----------------------

By using platform events, publishers can send custom event data through Apex, a process, a flow, Pub/Sub API, or other APIs. Subscribers can receive custom event messages from Salesforce or an external system using Apex, Pub/Sub API clients, processes, or flows. Based on the event message data, subscribers can process custom business logic, such as sending an email or logging a case. For example, a software system monitoring a printer can make an API call to publish an event when the ink is low. The custom printer event can contain custom fields for the printer model, serial number, and ink level. The event is processed in Salesforce by an Apex trigger that places an order for a new cartridge.

Platform events simplify the process of communicating changes and responding to them without writing complex logic. Publishers and subscribers communicate with each other through events. Multiple subscribers can listen to the same event and carry out different actions.

Define Your Platform Event

To define a platform event in Salesforce Classic or Lightning Experience:

1. From Setup, enter *Platform Events* in the Quick Find box, then select **Platform Events**.
2. On the Platform Events page, click **New Platform Event**.
3. Complete the standard fields, and optionally add a description.
4. For Publish Behavior, choose when the event message is published in a transaction.
 - **Publish After Commit** to have the event message published only after a transaction commits successfully. Select this option if subscribers rely on data that the publishing transaction commits. For example, a process publishes an event message and creates a task record. A second process that is subscribed to the event is fired and expects to find the task record. Another reason for choosing this behavior is when you don't want the event message to be published if the transaction fails.
 - **Publish Immediately** to have the event message published when the publish call executes. Select this option if you want the event message to be published regardless of whether the transaction succeeds. Also choose this option if the publisher and subscribers are independent, and subscribers don't rely on data committed by the publisher. For example, the immediate publishing behavior is suitable for an event used for logging purposes. With this option, a subscriber can receive the event message before data is committed by the publisher transaction.
5. Click **Save**.
6. To add a field, in the Custom Fields & Relationships related list, click **New**.
7. To set up the field properties, follow the custom field wizard.

 **Note**

- If you change the publish behavior, expect up to a 5-minute delay for the change to take effect.
- In Lightning Experience, platform events aren't shown in the Object Manager's list of standard and custom objects and aren't available in Schema Builder.

A platform event is a special Salesforce entity, similar in many ways to an sObject. An event message is an instance of a platform event, similar to how a record is an instance of a custom object. Unlike custom objects, you can't update or delete event records. You also can't view event records in the Salesforce user interface, and platform events don't have page layouts. When you delete a platform event definition, it's permanently deleted.

Standard Fields

Platform events include standard fields. These fields appear on the New Platform Event page.

Field	Description
Label	Name used to refer to your platform event in a user interface page.
Plural Label	Plural name of the platform event.
Starts with a vowel sound	If it's appropriate for your org's default language, indicate whether the label is preceded by "an" instead of "a."

Field	Description
Object Name	Unique name used to refer to the platform event when using the API. In managed packages, this name prevents naming conflicts with package installations. Use only alphanumeric characters and underscores. The name must begin with a letter and have no spaces. It can't end with an underscore or have two consecutive underscores.
Description	Optional description of the object. A meaningful description helps you remember the differences between your events when you view them in a list.
Deployment Status	Indicates whether the platform event is visible to other users.

Custom Fields

In addition to the standard fields, you can add custom fields to your custom event. Platform event custom fields support only these field types.

- Checkbox
- Date
- Date/Time
- Number
- Text
- Text Area (Long)

The maximum number of fields that you can add to a platform event is the same as for a custom object. See [Salesforce Features and Edition Allocations](#).

ReplayId System Field:

Each event message is assigned an opaque ID contained in the `ReplayId` field. The `ReplayId` field value, which is populated by the system when the event is delivered to subscribers, refers to the position of the event in the event stream. Replay ID values aren't guaranteed to be contiguous for consecutive events. A subscriber can store a replay ID value and use it on resubscription to retrieve events that are within the retention window. For example, a subscriber can retrieve missed events after a connection failure. Subscribers must not compute new replay IDs based on a stored replay ID to refer to other events in the stream.

EventUuid System Field

A universally unique identifier (UUID) that identifies a platform event message. In Pub/Sub API clients, the event `id` field contains the event UUID value and isn't versioned. In other subscriber clients, the **EventUuid** field is available in API version 52.0 and later. The API version corresponds to the version that an Apex trigger is saved with, or the version specified in a CometD subscriber endpoint.

API Name Suffix for Custom Platform Events

When you create a platform event, the system appends the `_e` suffix to create the API name of the event. For example, if you create an event with the object name `Low_Ink`, the API name is `Low_Ink__e`. The API name is used whenever you refer to the event programmatically, for example, in Apex. API names of standard platform events, such as `AssetTokenEvent`, don't include a suffix.

Event Subscribers

The Subscriptions related list shows all triggers, processes, and platform event-triggered flows that are subscribed to a platform event. Pub/Sub API subscribers, CometD subscribers, such as your own CometD client or the `empApi` Lightning component, aren't listed in this page.

The list shows the replay ID of the event that the system last processed (Last Processed Id field) and the event last published (Last Published Id field). Knowing which replay ID was last processed is useful when there's a gap in the events published and processed. For example, if a trigger contains complex logic that causes a delay in processing large batches of incoming events.

-  **Note** For high-volume platform events, the Last Published Id value isn't available and is always shown as Not Available.

Also, the Subscriptions list shows the state of each subscriber, which can be one of the following.

- `Running` –The subscriber is actively listening to events. If you modify the subscriber, the subscription continues to process events.
- `Error` – The subscriber was disconnected and stopped receiving published events. A trigger reaches this state when it exceeds the number of maximum retries with the `EventBus.RetryableException`. Trigger assertion failures and unhandled exceptions don't cause the error state. We recommend limiting the retries to fewer than nine times to avoid reaching this state. When you fix and save the trigger, or for a managed package trigger, if you redeploy the package, the trigger resumes automatically from the tip, starting from new events. Also, you can resume a trigger subscription in the subscription detail page that you access from the platform event page.
- `Suspended` –The subscriber is disconnected and can't receive events because a Salesforce admin suspended it or due to an internal error. You can resume a trigger subscription in the subscription detail page that you access from the platform event page. To resume a process, deactivate it and then reactivate it. If you modify the subscriber, the subscription resumes automatically from the tip, starting from new events.

-  **Note** Only one “Process” subscriber appears in the Subscriptions related list for all paused flow interviews that are subscribed to the platform event. Processes and platform event-triggered flows are listed individually.

Also, information about event subscribers is exposed in the `EventBusSubscriber` object. You can query this object to obtain details about subscribers.

Suspend or Resume an Apex Trigger Subscription:

Resume a suspended trigger subscription where it left off, starting from the earliest available event message that is stored in the event bus. If you want to bypass event messages that are causing errors or are no longer needed, you can resume a subscription from the tip, starting from new event messages.

To manage a trigger subscription:

1. In the Subscriptions related list, click **Manage** next to the Apex trigger.
2. In the subscription detail page, choose the appropriate action.
 - To suspend a running subscription, click **Suspend**.
 - To resume a suspended subscription, starting from the earliest event message that is available in the event bus, click **Resume**.
 - To resume a suspended subscription, starting from new event messages, click **Resume from Tip**.

You can't manage subscriptions for flows and processes through the Subscriptions related list.

Note

- After you modify a subscriber, the subscription resumes automatically. For more information, see the [Event Subscribers](#) section.
- If you click **Resume** for a trigger that is in the error state, the trigger skips the events that were retried with `EventBus.RetryableException`. The subscription starts with the unprocessed events sent after the error state and that are within the retention window.

Platform Event Considerations

- Field-Level Security—All platform event fields are read only by default, and you can't restrict access to a particular field. Field-level security permissions don't apply and the event message contains all fields.
- Enforcement of Field Attributes—Platform event records are validated to ensure that the attributes of their custom fields are enforced. Field attributes include the Required and Default attributes, the precision of number fields, and the maximum length of text fields.
- Permanent Deletion of Event Definitions—When you delete an event definition, it's permanently removed and can't be restored. Before you delete the event definition, delete the associated triggers. Published events that use the definition are also deleted.
- Renaming Event Objects—Before you rename an event, delete the associated triggers. If the event name is modified after clients have subscribed to this event, the subscribed clients must resubscribe to the updated topic. To resubscribe to the new event, add your trigger for the renamed event object.
- No Associated Tab—Platform events don't have an associated tab because you can't view event records in the Salesforce user interface.
- No SOQL Support—You can't query event messages using SOQL.
- No Record Page Support in Lightning App Builder—When creating a record page in Lightning App Builder, platform events that you defined show up in the list of objects for the page. However, you can't create a Lightning record page for platform events because event records aren't available in the user interface.
- Platform Events in Package Uninstall—When uninstalling a package with the option **Save a copy of this**

- package's data for 48 hours after uninstall** enabled, platform events aren't exported.
- Event Volume in Package Installations and Upgrades—Installing a managed or unmanaged package that contains a standard-volume platform event causes the event type to be saved as high volume in the subscriber org. Upgrading a managed package doesn't change the event volume in the subscriber org.
 - No Support in Professional and Group Editions—Platform events aren't supported in Professional and Group Edition orgs. Installation of a package that contains platform event objects fails in those orgs.

See Also

[Platform Events Developer Guide](#)

[Platform Event Allocations](#)

Event Relay

Integrate your real-time events with Amazon Web Services (AWS). Use Event Relay to send platform events and change data capture events from Salesforce to Amazon EventBridge.

Get Started

By using Event Relay in Salesforce with Amazon EventBridge in AWS, your Salesforce event-driven apps can use AWS services to process events or send events to third-party and SaaS integrations. You can also send platform events from AWS to Salesforce, where subscribers can process them using Salesforce Platform capabilities.

[Relay Events from Salesforce to Amazon EventBridge](#)

This section contains steps to create an event relay in Salesforce.

[Event Relay Statuses and Options](#)

Check out information about event relay statuses and the error recovery option.

[Send Events from AWS Back to Salesforce Using an API Destination](#)

An API destination is the target of a rule, and it enables you to route events to HTTP endpoints. Specifically, you can use an API destination to publish an event to Salesforce from EventBridge using REST API.

Get Started

By using Event Relay in Salesforce with Amazon EventBridge in AWS, your Salesforce event-driven apps can use AWS services to process events or send events to third-party and SaaS integrations. You can also send platform events from AWS to Salesforce, where subscribers can process them using Salesforce Platform capabilities.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Unlimited, and Developer** editions.

Not Available in: Government Cloud

Bidirectional Event Flow Between Salesforce and AWS

With Event Relay, you don't need an integration app to subscribe to events from Salesforce and publish them to AWS. After you set up an event relay, each custom platform event or change data capture event that is published to the Salesforce event bus is automatically forwarded to an event bus in Amazon EventBridge.

Also, you can publish event messages from AWS back to Salesforce using EventBridge API destinations. The bidirectional event flow between Salesforce and AWS enables your event-driven apps to use the capabilities of AWS services and the Salesforce Platform.



Managed Event Stream

Event relay subscribes to the events in the channel and streams them from the Salesforce event bus to EventBridge. Event relay keeps track of the point in the stream where the last event was sent to EventBridge. If transient errors occur, such as connectivity issues, event relay resumes automatically after the errors are resolved. It attempts to resume the event stream from where it left off, if possible. Also, you can monitor the status of your event relays, and can change the status by starting, pausing, and stopping an event relay.

Additional Resources

To learn more about Amazon EventBridge, see [What Is Amazon EventBridge?](#) in the AWS documentation. To learn more about event relay and platform events, refer to these resources.

- [Video: Salesforce Platform and AWS](#). This video covers event relays and another product that integrates with Amazon DynamoDB.
- [Trailhead: Platform Events Basics](#)
- [Trailhead: Change Data Capture Basics](#)
- [Platform Events Developer Guide](#)
- [Change Data Capture Developer Guide](#)

[Event Integration with Amazon Eventbridge](#)

After creating the event relay in Salesforce, the event relay creates a partner event source in EventBridge. You associate the partner event source with an event bus. After you start the event relay in Salesforce, the event bus in EventBridge receives the events published and relayed from the Salesforce event bus. You can process the received events using AWS services.

[Sending Events to Salesforce using API Destinations](#)

After processing the events in AWS, you can publish an event to Salesforce using an API destination. An API destination is an HTTP endpoint that an EventBridge rule can invoke.

[Event Relays in Setup](#)

An event relay is a set of event configurations and a connection to a partner event source in Amazon EvenBridge. An event relay streams events from the Salesforce event bus to EventBridge. You can

create an event relay in Setup.

Event Channels

An event relay uses an event channel. A channel is a stream of events in the Salesforce event bus. A single channel can contain multiple types of platform events, including real-time event monitoring events, or change data capture events.

Event Relay Security

Event relays send events to Amazon EventBridge securely using HTTP/1.1 with TLS. Event data is encrypted while in transit over the web. Salesforce is a registered partner of Amazon EventBridge and uses EventBridge Partner APIs to send events to Amazon EventBridge.

Event Type, Size, and Features

Check out what event types and features are supported with event relays.

Event Relay Considerations

Keep these considerations in mind when using Event Relay.

Event Relay Allocations

Check out allocations related to the event relays that you can create and events relayed to Amazon EventBridge.

Event Integration with Amazon Eventbridge

After creating the event relay in Salesforce, the event relay creates a partner event source in EventBridge. You associate the partner event source with an event bus. After you start the event relay in Salesforce, the event bus in EventBridge receives the events published and relayed from the Salesforce event bus. You can process the received events using AWS services.



For more information, see [Amazon EventBridge Partner Onboarding Guide](#) in the AWS documentation.

Process Events in Amazon EventBridge Using Rules

When you publish an event to the Salesforce event bus, Salesforce relays the event to AWS using the connection that is set up through the event relay configuration. After events are received in Amazon EventBridge from Salesforce, you can process the received events in EventBridge using rules.

Rules match incoming events in an event bus and assign them to targets. Use targets, such as AWS services, to perform actions on events. You can add one or more targets to a rule. With rules, you can filter on the payload so that you can send specific events to specific targets. For more information, see [Amazon EventBridge rules](#) and [Amazon EventBridge targets](#) in the AWS documentation.



Sending Events to Salesforce using API Destinations

After processing the events in AWS, you can publish an event to Salesforce using an API destination. An

API destination is an HTTP endpoint that an EventBridge rule can invoke.

To publish an event to Salesforce, the API destination configuration contains the REST API endpoint of the platform event and performs a POST request to this endpoint. It uses the Salesforce org authentication credentials that you set. The sent event must be defined as a custom platform event in Salesforce. Subscribers that have been set up for this platform event, such as a flow or an Apex trigger, receive the sent platform event and can process it.

See Also

[AWS Documentation: API Destinations](#)

Event Relays in Setup

An event relay is a set of event configurations and a connection to a partner event source in Amazon EventBridge. An event relay streams events from the Salesforce event bus to EventBridge. You can create an event relay in Setup.

This image shows a list of event relays set up for various event channels, in Setup, in the Event Relays page.



Event Relay Components

An event relay includes your AWS account information and a Salesforce event channel that you set up. The event relay status indicates whether the relay is running or is in another state (1). The event channel is the stream of events in the Salesforce event bus to be relayed to EventBridge (2). An event channel contains members for the events to stream. The number of members a channel contains is shown in the event relays list (3). A named credential contains your AWS account and AWS region information where to relay the events (4). After you create an event relay, Salesforce creates a corresponding partner event source in EventBridge.



Monitor the Event Relay Execution in Setup

Check your event relay's execution status, the date, and time of the last relayed event, and any error message.

To view the status of all the event relays, check the Status column in the Event Relays page.



If the status is Error, you can view the error message on the event relay detail page by clicking the event relay.

To get the timestamp of the last event relayed to Amazon EventBridge, click the event relay and check out the **Last Relayed** field.



Event Channels

An event relay uses an event channel. A channel is a stream of events in the Salesforce event bus. A single channel can contain multiple types of platform events, including real-time event monitoring events, or change data capture events.

Each event relay is associated with a unique channel that's identified by its channel ID. Don't reuse a channel for another event relay.

- Tip** To stream the same type of event to different AWS regions, create a separate channel for each AWS region, and then create an event relay for each channel.

Custom Channels for Platform Events and Change Events

Use a custom channel to group event messages from one or more events through channel members. For example, for platform events, if `MyEvent1__e` is a custom platform event defined in Salesforce, create a channel to receive event messages for `MyEvent1__e` if you add a channel member for `MyEvent1__e`. To receive event messages for another event on that channel, such as `MyEvent2__e`, add a channel member for `MyEvent2__e`.

- Tip** If you're using a custom platform event, make sure that platform event is defined in Salesforce. For more information, see [Platform Event Fields](#) in the *Platform Events Developer Guide*.

For the maximum number of custom channels that you can create for custom platform events and real-time event monitoring events, see [Common Platform Event Allocations](#) in the *Platform Events Developer Guide*.

Similarly, for change data capture, you can create a custom channel to receive change event messages. If you create a channel for changes in one Salesforce object, such as `AccountChangeEvent`, the channel contains one member for `AccountChangeEvent`. To receive events for another Salesforce object, such as `LeadChangeEvent`, add another channel member.

- Tip** Salesforce provides change events, so you don't define them.

For more information about creating custom channels, see [Create a Channel and Channel Members](#).

For the maximum number of channels that you can create for change events, see [Default Change Event Allocations](#) in the *Change Data Capture Developer Guide*.

The Standard `ChangeEvents` Channel

The predefined standard channel `ChangeEvents` includes all change events that correspond to the Salesforce objects that you select in Setup. You can select `ChangeEvents` from the channel's dropdown in the event relay creation wizard. Before you can use the standard `ChangeEvents` channel, from Setup, select the objects that you want change events for in the Change Data Capture page. For more information, see [Select Objects for Change Notifications in the User Interface](#) in the *Change Data Capture Developer Guide*.

Filtered Channels for Platform Events and Change Events

Add a filter to a custom channel and send only events that match the filter criteria to Amazon EventBridge. By using a filtered channel, you reduce the volume of events relayed to Amazon EventBridge, process events efficiently, and decrease the consumption of the event delivery allocation. Event relay filtering is available for custom channels that reference custom platform events and change events but not for real-time event monitoring events.

To add a filter to a platform event channel, see [Filter Your Stream of Platform Events with Custom Channels](#) in the *Platform Events Developer Guide*.

To add a filter to a change data capture channel, see [Filter Your Stream of Change Events with Channels](#) in the *Change Data Capture Developer Guide*.

Change Event Enrichment in Channels

Change event messages include values for new and changed fields, but sometimes unchanged field values are needed for processing or replicating data. To enrich change events with one or more fields that are always present in the event message, create a custom channel with enriched fields. To learn how to add enrichment, see [Enrich Change Events with Extra Fields](#) in the *Change Data Capture Developer Guide*.

Event Relay Security

Event relays send events to Amazon EventBridge securely using HTTP/1.1 with TLS. Event data is encrypted while in transit over the web. Salesforce is a registered partner of Amazon EventBridge and uses EventBridge Partner APIs to send events to Amazon EventBridge.

As a registered partner of EventBridge, Salesforce creates partner event sources and streams events to EventBridge without the need for AWS account authentication. Event relays use only the AWS account ID and region, which are stored in named credentials.

See Also

- [Amazon EventBridge Partner Onboarding Guide](#)
- [Create a Named Credential for Event Relay Setup](#)

Event Type, Size, and Features

Check out what event types and features are supported with event relays.

Event Relay supports these event types and maximum sizes.

- Custom high-volume platform events
- Real-time event monitoring events
- Change data capture events
- The maximum event message size that Salesforce supports for published and received events is 1 MB.
- The maximum event message size that Amazon EventBridge supports for incoming and outgoing events is 256 KB.
- If an event message larger than 256 KB is relayed from Salesforce to EventBridge, the event isn't delivered to EventBridge.

These event types aren't supported.

- Standard platform events
- Standard-volume custom platform events

These features aren't supported with Event Relay.

- Salesforce Private Connect
- Developer Edition orgs with namespaces

Event Relay Considerations

Keep these considerations in mind when using Event Relay.

Event Relay on GovCloud

GovCloud users can configure event relay to send events to either AWS GovCloud or AWS Commercial. You're responsible for configuring which data is transferred and for determining the data's destination. Depending on your choices, it's possible for the data to leave the FedRAMP authorized environment.

Deleting an Event Relay

When you delete an event relay, its corresponding partner event source in Amazon EventBridge is deleted. The event bus that is associated with the partner event source in AWS isn't affected.

Event Relays Aren't Copied to Sandbox

When you create, clone, or refresh a sandbox, event relays aren't copied to sandbox. To test event relays in sandbox, recreate the event relays in your sandbox.

Event Encryption with Platform Shield Encryption

Starting in Summer '23, Event Relay supports Salesforce orgs with Shield Platform Encryption enabled. You can send platform events and change data capture events encrypted at rest with Shield Platform Encryption to Amazon EventBridge.

For more information about event encryption at rest with Shield Platform Encryption, see [Encrypting Platform Event Messages at Rest in the Event Bus](#) in the *Platform Events Developer Guide* and [Change Events for Encrypted Salesforce Data](#) in the *Change Data Capture Developer Guide*.

AWS Regions

The Event Relay service is globally available and can connect Salesforce orgs to AWS accounts in default AWS regions. Event relay doesn't support opt-in AWS regions. For more information on AWS region availability, see the [AWS Regional availability reference](#). Customer understands and agrees that the Event Relay service processes data, including Customer Data, on Hyperforce (a Salesforce-controlled Amazon Web Services instance), and such processing can occur in a region outside of the region in which Customer's Salesforce org is located.

Processing occurs in these regions:

- United States
- European Union

Events aren't persisted or stored outside the region where your org is located.

Impact of Salesforce Maintenance Activities

Some Salesforce maintenance activities, such as an org migration to a new data center or instance refresh, reset the stream of stored platform events and change events. The stream reset results in the stored events no longer being available in the event bus. After org migration, event relays resume from the newly published events. Any events received during org migration or instance refresh aren't forwarded to Amazon EventBridge.

You can configure event relays with an error-recovery option to resume from the earliest events stored in the event bus after an unrecoverable error occurs. If an unrecoverable error occurs after the maintenance activity, and the event relay is configured with this option, it doesn't resume from the events stored before the maintenance activity. It resumes from the latest events received and events stored after the maintenance activity. For more information, see [Error Recovery Options](#). For instructions about mitigating the impact of Salesforce maintenance activities, see [How to Prepare for an Org Migration](#) and [Instance Refresh Maintenance](#).

Event Relay Allocations

Check out allocations related to the event relays that you can create and events relayed to Amazon

EventBridge.

Number of Event Relays

The maximum number of event relays that you can create is 200.

Events Delivered to Amazon EventBridge

The maximum number of platform events and change events delivered to Amazon EventBridge in the last 24 hours across all event relays corresponds to your event allocations. The event delivery usage is counted when the event is delivered to Amazon EventBridge. It isn't counted when the event is stored in the event bus before delivery. The event delivery allocations are common for platform events and change events. Also, they're shared with CometD clients, Pub/Sub API clients, and empApi Lightning components. For more information, see [Platform Event Allocations](#) in the *Platform Event Developer Guide* and [Change Data Capture Allocations](#) in the *Change Data Capture Developer Guide*.

Relay Events from Salesforce to Amazon EventBridge

This section contains steps to create an event relay in Salesforce.

Create Prerequisite Items

Create the prerequisite items that your event relay uses, such as a named credential, a channel, a channel member, and optionally a platform event.

Create an Event Relay in Setup

After creating the prerequisite items, follow the steps in this section to create an event relay using a point-and-click interface in Setup.

Verify and Process Events in EventBridge

Verify receiving events in EventBridge and process them with AWS services.

Create Prerequisite Items

Create the prerequisite items that your event relay uses, such as a named credential, a channel, a channel member, and optionally a platform event.

(Optional) Define a Platform Event

If you have a custom platform event defined in your Salesforce org that you want to use for the event relay, you can skip this step. You can also skip this step if you want to use the event relay with a change data capture event. Otherwise, follow the steps in this section to define an example platform event, [Carbon_Comparison__e](#), which the event relay configuration steps use.

Create a Named Credential for Event Relay Setup

A named credential stores your AWS account information. You use the named credential later to set up an event relay. You can create a named credential in the Salesforce user interface in Setup.

Create a Channel and Channel Members

You can create a channel and add members by using Tooling API. A channel is a stream of events. It contains members that specify the type of events that can be received on the channel. A channel can

contain a stream of platform events, including real-time event monitoring events, or change data capture events. For a channel of change data capture events, you can select the standard [ChangeEvents](#) channel instead of creating a custom channel.

(Optional) Define a Platform Event

If you have a custom platform event defined in your Salesforce org that you want to use for the event relay, you can skip this step. You can also skip this step if you want to use the event relay with a change data capture event. Otherwise, follow the steps in this section to define an example platform event, [Carbon_Comparison__e](#), which the event relay configuration steps use.

REQUIRED EDITIONS

USER PERMISSIONS NEEDED

To define a platform event:	Customize Application
-----------------------------	-----------------------

1. From Setup, in the Quick Find box, enter *Platform Events*, and then select **Platform Events**.
2. Click **New Platform Event**.
3. Provide these values.
 - Label: *Carbon Comparison*
 - Plural Label: *Carbon Comparisons*
4. Save your changes.
5. Create these fields. In Custom Fields & Relationships, click **New** for each field, and follow the wizard.
 - Field type: Number, Field Label: *Annual Mileage*, Length: 6, Decimal Places: 0
 - Field type: Text, Field Label: *Current Vehicle*, Length: 100
 - Field type: Text, Field Label: *Model Year*, Length: 4

Create a Named Credential for Event Relay Setup

A named credential stores your AWS account information. You use the named credential later to set up an event relay. You can create a named credential in the Salesforce user interface in Setup.

REQUIRED EDITIONS

USER PERMISSIONS NEEDED

To create a named credential:	Customize Application
-------------------------------	-----------------------

1. From Setup, in the Quick Find box, enter *Named Credentials*, and then select **Named Credentials**.
2. Expand the dropdown next to **New**, and then click **New Legacy**.
3. Complete the fields.
 - a. For **Label**, enter *AWS US-West-2*.
 - b. For **Name**, enter *AWS_US_West_2*.
 - c. For **URL**, enter a URL in the format [arn:aws:aws_region:aws_account_number](#). Replace the

`aws_region` placeholder with your AWS region. Replace the `aws_account_number` placeholder with your 12-digit AWS account ID.

The URL is case-sensitive, and `aws_region` must be in capital letters. For example, the URL for an account in the US-WEST-2 region has this format: `arn:aws:US-WEST-2:XXXXXXXXXXXX`.

(`XXXXXXXXXXXX` is a placeholder for the 12-digit AWS account ID.)

- d. Keep the default values for **Identity Type** and **Authentication Protocol**.

- e. Keep **Generate Authorization Header** selected.



4. Save your changes.

Important You can't modify the AWS account information in a named credential if it has an associated event relay. The event relay doesn't pick up the new settings and can fail to run properly. To change the AWS account information, delete the associated event relay first. Next, modify the named credential, and then recreate the event relay with the named credential.

See Also

[Tooling API Developer Guide: NamedCredential](#)

[Metadata API Developer Guide: NamedCredential](#)

Create a Channel and Channel Members

You can create a channel and add members by using Tooling API. A channel is a stream of events. It contains members that specify the type of events that can be received on the channel. A channel can contain a stream of platform events, including real-time event monitoring events, or change data capture events. For a channel of change data capture events, you can select the standard `ChangeEvents` channel instead of creating a custom channel.

To learn about custom channels and the standard `ChangeEvents` channel, see [Event Channels](#).

Prerequisite: Set Up a REST API Tool

The setup steps in this section use the Tooling REST API. To complete the steps, you can use your favorite REST API tool. We recommend using Postman with the Salesforce API Collection, which contains handy templates for Salesforce API calls, including calls for creating channels and channel members. For instructions on how to set up Postman, see [Quick Start: Connect Postman to Salesforce](#) in *Trailhead*.

Create a Channel for Custom Platform Events or Change Events

To create a channel and channel member for custom platform events, complete the steps in this section for custom platform events. To create a channel and channel member for change events, complete the steps in this section for change events.

After creating a channel, you can add members to the channel. Each member represents one event type. A channel can have multiple members.

-  **Note** To create a channel that holds real-time event monitoring events, you follow a process that's similar to the one for creating a channel for custom platform events, with some differences. PlatformEventChannel must contain the `"eventType": "monitoring"` field, and the `selectedEntity` field in PlatformEventChannelMember references the API name of the event without the `_e` suffix. For the complete steps, see [Create a Custom Channel and Add Real-Time Event Monitoring Events with Tooling API](#) in the *Platform Events Developer Guide*.

[Create a Channel for a Custom Platform Event](#)

Create a channel that holds a stream of custom platform events.

[Add a Custom Platform Event in a New Channel Member](#)

To add a platform event to the channel that you created, create a channel member.

[Create a Channel for a Change Event](#)

Create a channel that holds a stream of change data capture events.

[Add a Change Event in a New Channel Member](#)

To add a change event to the channel that you created, add a channel member.

Create a Channel for a Custom Platform Event

Create a Channel for a Custom Platform Event

Create a channel that holds a stream of custom platform events.

REQUIRED EDITIONS

USER PERMISSIONS NEEDED

To create a PlatformEventChannel: Customize Application

1. Send a POST request to this URI.

```
/services/data/v65.0/tooling/sobjects/PlatformEventChannel
```

2. If you're using Postman, expand **Event Platform | Custom Channels | Platform Event**, and click **Create channel**.
3. To configure a channel that receives custom platform event messages, set `channelType` to `event`. The channel `label` appears in the event relays UI. Use this example request body. In Postman, click **Body**, and replace the body with this JSON body.

```
{  
    "FullName": "Carbon_Comparison_Channel__chn",  
    "Metadata": {  
        "channelType": "event",  
    }  
}
```

```
        "label": "Carbon Comparison Channel"
    }
}
```

4. Send the request. The response received looks similar to this response.

```
{
    "id": "0YLRM00000001es4AA",
    "success": true,
    "errors": [],
    "warnings": [],
    "infos": []
}
```

Now that you created a channel for platform events, add a channel member for one custom platform event. See [Add a Custom Platform Event in a New Channel Member](#).

See Also

[Tooling API Developer Guide: PlatformEventChannel](#)

[Metadata API Developer Guide: PlatformEventChannel](#)

Add a Custom Platform Event in a New Channel Member

Add a Custom Platform Event in a New Channel Member

To add a platform event to the channel that you created, create a channel member.

REQUIRED EDITIONS

USER PERMISSIONS NEEDED

To create a PlatformEventChannel: Customize Application

Prerequisites:

- [Create a Channel for a Custom Platform Event](#)
- Define the Carbon_Comparison__e platform event in [\(Optional\) Define a Platform Event](#) for this channel member.

1. Send a POST request to this URI.

```
/services/data/v65.0/tooling/sobjects/PlatformEventChannelMember
```

2. If you're using Postman, expand **Event Platform | Custom Channels | Platform Event**, and click

Create channel member.

3. Specify the channel in the `eventChannel` field and the event in the `selectedEntity` field. This example references a custom platform event, `Carbon_Comparison__e`. Use this example request body. In Postman, click **Body**, and replace the body with this JSON body.

```
{  
    "FullName": "Carbon_Comparison_Channel_chn_Carbon_Comparison_e",  
    "Metadata": {  
        "eventChannel": "Carbon_Comparison_Channel_chn",  
        "selectedEntity": "Carbon_Comparison__e"  
    }  
}
```

4. Send the request. The response received looks similar to this response.

```
{  
    "id": "0v8RM0000000G3iYAE",  
    "success": true,  
    "errors": [],  
    "warnings": [],  
    "infos": []  
}
```

A channel (PlatformEventChannel) can have multiple channel members (PlatformEventChannelMember), which means that you can add multiple platform events to a channel. This example adds only one platform event, `Carbon_Comparison__e`. To add another event to the channel, create another PlatformEventChannelMember.

See Also

[Tooling API Developer Guide: PlatformEventChannelMember](#)

[Metadata API Developer Guide: PlatformEventChannelMember](#)

Create a Channel for a Change Event

Create a Channel for a Change Event

Create a channel that holds a stream of change data capture events.

REQUIRED EDITIONS**USER PERMISSIONS NEEDED**

To create a PlatformEventChannel:

Customize Application

1. Send a POST request to this URI.

```
/services/data/v65.0/tooling/sobjects/PlatformEventChannel
```

2. If you're using Postman, expand **Event Platform | Custom Channels | Platform Event**, and click **Create channel**.
3. To configure a channel that receives change event messages, set `channelType` to `data`. The channel `label` appears in the event relays UI. Use this example request body. In Postman, click **Body**, and replace the body with this JSON body.

```
{  
    "FullName": "Account_Channel_chn",  
    "Metadata": {  
        "channelType": "data",  
        "label": "Account Channel"  
    }  
}
```

4. Send the request. The response received looks similar to this response.

```
{  
    "id": "0YLRM00000001fR4AQ",  
    "success": true,  
    "errors": [],  
    "warnings": [],  
    "infos": []  
}
```

Now that you created a channel for change events, add a channel member for one change event. See [Add a Change Event in a New Channel Member](#).

See Also

[Tooling API Developer Guide: PlatformEventChannel](#)

[Metadata API Developer Guide: PlatformEventChannel](#)

Add a Change Event in a New Channel Member

Add a Change Event in a New Channel Member

To add a change event to the channel that you created, add a channel member.

REQUIRED EDITIONS

USER PERMISSIONS NEEDED

To create a PlatformEventChannel: Customize Application

Prerequisites:

- [Create a Channel for a Change Event](#)

Change events are predefined by Salesforce so you don't define them. To learn about Salesforce objects that support change events, see [Change Event Object Support](#) in the *Change Data Capture Developer Guide*.

1. Send a POST request to this URI.

```
/services/data/v65.0/tooling/sobjects/PlatformEventChannelMember
```

2. If you're using Postman, expand **Event Platform | Custom Channels | Platform Event**, and click **Create channel member**.
3. Specify the channel in the `eventChannel` field and the event in the `selectedEntity` field. This example references a change event, `AccountChangeEvent`. Use this example request body. In Postman, click **Body**, and replace the body with this JSON body.

```
{  
    "FullName": "Account_Channel_chn_AccountChangeEvent",  
    "Metadata": {  
        "eventChannel": "Account_Channel__chn",  
        "selectedEntity": "AccountChangeEvent"  
    }  
}
```

4. Send the request. The response received looks similar to this response.

```
{  
    "id": "0v8RM0000000GHCYA2",  
    "success": true,  
    "errors": [],  
    "warnings": [],  
    "infos": []  
}
```

A channel (PlatformEventChannel) can have multiple channel members (PlatformEventChannelMember), which means that you can add multiple change events to a channel. This example adds only one change event, `AccountChangeEvent`. To add another event to the channel, create another PlatformEventChannelMember.

See Also

[Tooling API Developer Guide: PlatformEventChannelMember](#)

[Metadata API Developer Guide: PlatformEventChannelMember](#)

Create an Event Relay in Setup

After creating the prerequisite items, follow the steps in this section to create an event relay using a point-and-click interface in Setup.

Event relays are available in Tooling API and Metadata API. As an alternative to creating an event relay in Setup, you can create it using the API.

[Create an Event Relay](#)

An event relay associates the channel in the Salesforce event bus with Amazon EventBridge using a named credential. The creation of the event relay results in a partner event source being created in Amazon EventBridge. The partner event source is created with a pending status.

[Activate the Partner Event Source in Amazon EventBridge](#)

Salesforce creates a partner event source in Amazon EventBridge in pending status after you create an event relay. Associate the event bus with the partner event source in EventBridge so that the event source is ready to receive events from Salesforce.

[Start the Event Relay](#)

Change the event relay state in the Edit window.

See Also

[Tooling API Developer Guide: EventRelayConfig](#)

[Metadata API Developer Guide: EventRelayConfig](#)

Create an Event Relay

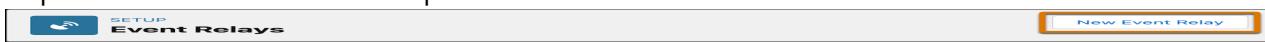
An event relay associates the channel in the Salesforce event bus with Amazon EventBridge using a named credential. The creation of the event relay results in a partner event source being created in Amazon EventBridge. The partner event source is created with a pending status.

REQUIRED EDITIONS

USER PERMISSIONS NEEDED

To create an event relay:

Customize Application

1. From Setup, in the Quick Find box, enter *Event Relays*, and then select **Event Relays**.
2. In the Event Relays page, click **New Event Relay**. Use the event relay creation wizard and provide the requested information in each step.

3. Enter a label for the event relay, for example, *Carbon Comparison Relay*. The label appears in the event relays list view and the event relay detail page. Make sure you choose a meaningful name and try

to make it unique. The name is auto populated based on the label that you provide.

4. Select a named credential that contains your AWS account information from the dropdown. To create a named credential, see [Create a Named Credential for Event Relay Setup](#).

5. Select a channel that references the events that you want to send to EventBridge from the dropdown.

To create a channel, see [Create a Channel and Channel Members](#).

6. Select an error recovery option. This option is used when the system can't resume from the last relayed event after it recovers from an error. You can keep the default.

7. Review the summary screen, and save your changes to create the event relay.

After you create the event relay, the event relay detail page contains the **Partner Event Source Name** field. This field is populated after a short delay with the name of the partner event source that's created in Amazon EventBridge. Wait and then refresh the detail page to get the partner event source. You use this field value in the next step to find the partner event source in Amazon EventBridge.

The event relay is created in a stopped state, and no events are relayed to Amazon EventBridge. Before starting the event relay, activate the partner event source in Amazon EventBridge. You perform these steps next.

Activate the Partner Event Source in Amazon EventBridge

Salesforce creates a partner event source in Amazon EventBridge in pending status after you create an event relay. Associate the event bus with the partner event source in EventBridge so that the event source is ready to receive events from Salesforce.

REQUIRED EDITIONS

USER PERMISSIONS NEEDED

To create an event relay:

Customize Application

1. Get the partner event source name from the event relay. In the Event Relays page, click the event relay. Copy the **Partner Event Source Name** field value.

There can be a small delay before this field value is displayed after you create the event relay. Refresh the page until you get the partner event source name.

2. Log in to the AWS console. Navigate to <https://aws.amazon.com> and sign in using your AWS account credentials.
3. In the search box, enter *Amazon EventBridge*, and then from Services, select **Amazon EventBridge**.
4. In Amazon Eventbridge, under Integration, select **Partner event sources**.
5. In the search box, enter the name of your event source, the **Partner Event Source Name** field value

that you copied earlier.

6. Select your event source, and click **Associate with event bus**.
7. Click **Associate**. The status of the event source changes to Active.



Start the Event Relay

Change the event relay state in the Edit window.

REQUIRED EDITIONS

USER PERMISSIONS NEEDED

To create an event relay:	Customize Application
---------------------------	-----------------------

1. Start the event relay either from the event relays page or the event relay detail page.
 - a. In the Event Relays page, click the dropdown under the Actions column, and select **Edit**.
 - b. In the Event relay detail page, click **Edit**.
2. In the Edit window, in the State dropdown, select **Run**.
3. Save your changes.

Verify and Process Events in EventBridge

Verify receiving events in EventBridge and process them with AWS services.

Create an EventBridge Rule for Logging Events in a CloudWatch Log

To test the connection between Salesforce and AWS and ensure that the event messages are received in EventBridge, add a CloudWatch log as a target of a rule. Incoming event messages from Salesforce are logged in CloudWatch.

Verify Receiving Events in CloudWatch

Verify that events sent from Salesforce are received in EventBridge using a CloudWatch log.

Processing Events with AWS Services

After receiving the events in EventBridge, you can process them with AWS using rules, or you can send them to third-party and SaaS integrations using API destinations.

Create an EventBridge Rule for Logging Events in a CloudWatch Log

To test the connection between Salesforce and AWS and ensure that the event messages are received in EventBridge, add a CloudWatch log as a target of a rule. Incoming event messages from Salesforce are logged in CloudWatch.

For more information, see [Amazon EventBridge rules](#) and [Amazon EventBridge targets](#) in the AWS documentation.

1. In Amazon Eventbridge, click **Rules**.
2. Select your event bus from the dropdown. The name of the event bus is the same as the name of the partner event source that you queried earlier. It's in this format: `aws.partner/salesforce.com/orgID/channelID`
3. In the Rules section, click **Create rule**.
4. Provide a name for your rule.
5. Click **Next**.
6. Under Event source, select **AWS events or EventBridge partner events**.
7. Skip the Sample event section. In Event pattern, for Event source, select **EventBridge partners**.
 - a. From the partner dropdown, select **Salesforce**.
 - b. For Event type, select **All Events**. The event pattern box autopopulates to this value.

```
{  
  "source": [ {  
    "prefix": "aws.partner/salesforce.com"  
  } ]  
}
```

8. The rule matches incoming events based on the defined pattern. If you want to add a filter to match a specific platform event, match events whose `detail-type` field contains the API name of the platform event, for example, `Carbon_Comparison__e`. For more information about event pattern matching, see [Content filtering in Amazon EventBridge event patterns](#) in the AWS documentation.
 - a. Under Event pattern, click **Custom patterns (JSON editor)**.
 - b. In the input box, In Define pattern, select **Custom pattern**, and replace the pattern with this pattern.

```
{  
  "source": [ {  
    "prefix": "aws.partner/salesforce.com"  
  } ],  
  "detail-type": [ "Carbon_Comparison__e" ]  
}
```

9. Click **Next**.
10. Under Target 1, Target types, select **AWS service**.
11. Under Select a target, select **CloudWatch log group**.
12. Complete the log group path. For example: `/aws/events/mygroup/log`.
13. Click **Next** and then **Next**.
14. Review the rule that you created, and then click **Create rule**.

After creating the rule and the Cloudwatch log, verify receiving events in [Verify Receiving Events in CloudWatch](#).

Verify Receiving Events in CloudWatch

Verify that events sent from Salesforce are received in EventBridge using a CloudWatch log.

1. Click the rule that you just created.
2. Under Targets, click the log. The CloudWatch log opens in a new tab.
3. Publish an event from Salesforce. In this example, we publish an event message with REST API.
 - a. Perform a POST request to this URI. If using Postman, click **REST > SObject > SObject Create**, and replace the `:SOBJECT_API_NAME` placeholder with `Carbon_Comparison_e`.

```
/services/data/v58.0/sobjects/Carbon_Comparison_e/
```

Use this request body.

```
{
  "Annual_Mileage__c" : "12003",
  "Current_Vehicle__c" : "Fast Car",
  "Model_Year__c" : "2021"
}
```

To find out about methods for publishing an event, see [Publishing Platform Events](#) in the *Platform Events Developer Guide*. For example, you can use REST API to publish this `Carbon_Comparison_e` event message.

4. You get a response similar to this response. The status of `OPERATION_ENQUEUED` means that the platform event message is published asynchronously.

```
{
  "id": "e00xx0000000001AAA",
  "success": true,
  "errors": [
    {
      "statusCode": "OPERATION_ENQUEUED",
      "message": "e4aa03c9-d0e0-4c80-bf93-2e6779096018",
      "fields": []
    }
  ]
}
```

5. Refresh the CloudWatch log stream. The received event is displayed in the log group, similar to this event.

```
{
  "version": "0",
  "id": "6f33d717-8e35-4488-5690-89e681b5737c",
  "detail-type": "Carbon_Comparison_e",
  "source": "aws.partner/salesforce.com/00DRM00000LqxV2AS/0YLRM00000001es
4AA",
  "account": "XXXXXXXXXXXXXX",
  "time": "2023-04-03T19:23:23Z",
```

```
"region": "us-west-2",
"resources": [],
"detail": {
    "payload": {
        "CreatedById": "005RM000002eap4YAA",
        "Current_Vehicle__c": "Fast Car",
        "CreatedDate": "2023-04-03T19:23:22.686Z",
        "Model_Year__c": "2021",
        "Annual_Mileage__c": 12003
    },
    "schemaId": "mVBhpA7tU9M CtBEbqjycBQ",
    "id": "118dab75-7b45-498e-8478-52d0324c1060"
}
}
```

The received event contains the payload of the Salesforce event in the `detail` field. The top-level fields, such as `version` and `detail-type`, are EventBridge event fields. The `detail-type` field contains the Salesforce event API name. In combination with the `source` field, the `detail-type` field identifies the fields and values that appear in the `detail` field. For more information, see [Amazon EventBridge events](#) in the AWS documentation.

Processing Events with AWS Services

After receiving the events in EventBridge, you can process them with AWS using rules, or you can send them to third-party and SaaS integrations using API destinations.

For example, you can use a Lambda function to make a computation, or you can use the Simple Notification Service (SNS) to send messages to customers. For more information about AWS, see <https://aws.amazon.com/> and the [AWS documentation site](#). For more information about API destinations, see [API Destinations](#) in the AWS documentation.

Event Relay Statuses and Options

Check out information about event relay statuses and the error recovery option.

[Event Relay Execution Statuses](#)

You can run, pause, or stop the execution of an event relay. After you change the execution state, it can take several minutes for the new state to take effect. During this time, the event relay displays a transitional status in the UI. After the state change completes, the final status appears.

[Edit an Event Relay and Change Its Status](#)

You can edit an event relay to change its status, label, and the error recovery option.

[Error Recovery Options](#)

If transient errors occur, such as connectivity issues or errors in the event relay configurations, event relay resumes automatically after the connectivity issues are resolved or the configurations are fixed.

The event relay attempts to resume sending events from the event bus from where it left off. In rare occasions, if it can't resume after the last relayed event, it uses the error recovery option to determine where to resume from.

Monitoring Event Relays Using SOQL Queries

You can perform a SOQL query on the EventRelayFeedback Salesforce object to get information about an event relay, including its status, the date and time of the last relayed event, and any error message. The information contained in EventRelayFeedback is available in Setup, on the Event Relays page.

Event Relay Execution Statuses

You can run, pause, or stop the execution of an event relay. After you change the execution state, it can take several minutes for the new state to take effect. During this time, the event relay displays a transitional status in the UI. After the state change completes, the final status appears.

Transitional Status and Final Status

Because a state change happens asynchronously, changing the event relay state isn't immediate and can take a short while. When the state change is in progress, the event relays list view and the event relay detail page show a transitional status. The transitional status is a visual indication that the state change is underway. Transitional statuses are available only in the UI and not in the API.

For example, when you change the state of a stopped relay to **Run**, the status first changes to **Starting...** before it changes to **Running**. And when you change the state of a paused relay to **Run**, the status first changes to **Resuming...** before it changes to **Running**. This image shows a transitional status of **Starting** for one event relay and a final status of **Paused**, **Running**, and **Stopped** for the other event relays.



 **Note** Changing the state from the **Error** state doesn't cause a transitional status to appear.

These are the possible transitional states.

- **Starting...**
- **Resuming...**
- **Pausing...**
- **Stopping...**

The final statuses are listed in the next section.

Running Status

To start a stopped or paused event relay, set the event relay state to **Run**.

Paused Status

To pause the event relay and temporarily hold off sending events to Amazon EventBridge, set the event relay state to **Pause**. When you resume a paused event relay, the relay of events continues from where it left off. Stored events are relayed from the Salesforce event bus starting after the last relayed event, and new events.

When the event relay is paused, its current state information is saved in the `EventRelayFeedback` Salesforce object. There can be a delay between when you set the state to `Pause` and when the event relay is paused. In this case, the saved state information corresponds to the state when the event relay was actually paused.

Stopped Status

To stop relaying events from Salesforce to Amazon EventBridge, change the event relay state to **Stop**. When you start a stopped event relay, new events published to the Salesforce event bus are delivered to Amazon EventBridge. Any events stored in the Salesforce event bus that were received after the relay was stopped and before it was started aren't sent.

When you stop an event relay, its current state information in `EventRelayFeedback` is deleted.

Error Status

The system sets the event relay status to **Error** if it encounters an error when attempting to run the event relay. Errors can be caused by configuration problems or connectivity issues to Amazon EventBridge. You can view the error in Setup or by querying `EventRelayFeedback`. For a list of error codes, see [EventRelayFeedback](#) in the *Object Reference for the Salesforce Platform*.

The system attempts to recover from the error periodically. When the system recovers from the error, the event relay resumes from where it left off and the status changes to **Running** or a status you select. In rare occasions, if the system can't resume from where it left off, it uses the error recovery option to determine from where to relay events. For more information, see [Error Recovery Options](#).

Edit an Event Relay and Change Its Status

You can edit an event relay to change its status, label, and the error recovery option.

REQUIRED EDITIONS

USER PERMISSIONS NEEDED

To edit an event relay:

Customize Application

1. Open the Edit window for the event relay either from the event relays page or the event relay detail page.

- a. In the Event Relays page, click the dropdown under the Actions column, and select **Edit**.
 - b. In the Event relay detail page, click **Edit**.
2. In the Edit window, change the label, status, or the error recovery option.

Error Recovery Options

If transient errors occur, such as connectivity issues or errors in the event relay configurations, event relay resumes automatically after the connectivity issues are resolved or the configurations are fixed. The event relay attempts to resume sending events from the event bus from where it left off. In rare occasions, if it can't resume after the last relayed event, it uses the error recovery option to determine where to resume from.

Specify the error recovery option when you create or edit the event relay in Setup. Choose one of these options.

- Resume from the latest events received. This option skips sending events that were published during the error. This option is the default.
- Resume from the earliest events stored in the event bus. This option sends new events and any other events less than 72 hours old. You can reprocess all stored events and catch up on missed events.

Also, you can specify the error recovery option in Tooling API or Metadata API via the **relayOption** field in `EventRelayConfig`. When you create an event relay, add the **relayOption** field with the desired value in the `EventRelayConfig` POST request. Or you can update an event relay by performing a PATCH request for `EventRelayConfig` and supplying a value for **relayOption**. For more information about this field, see [EventRelayConfig](#) in the *Tooling API Developer Guide* and [EventRelayConfig](#) in the *Metadata API Developer Guide*.

Monitoring Event Relays Using SOQL Queries

You can perform a SOQL query on the `EventRelayFeedback` Salesforce object to get information about an event relay, including its status, the date and time of the last relayed event, and any error message. The information contained in `EventRelayFeedback` is available in Setup, on the Event Relays page.

To run the queries given in this section, you can use one of these methods.

- The Developer Console's Query Editor. For more information about the Developer Console, see [Developer Console](#) and [Developer Console Query Editor](#).
- The query REST API resource using a REST API tool, such as Postman. See [Query](#) in the *REST API Developer Guide*. For information on how to set up Postman and use the Salesforce API Collection, see [Salesforce APIs for Postman](#) in GitHub. In REST API, spaces in the query must be replaced with +. In Postman, you don't replace spaces in queries because Postman handles them.

You can perform a REST call, a GET request, to this endpoint with the SOQL query appended. Because `EventRelayFeedback` is a Salesforce object, use the data REST API to query the object and not Tooling API. In Postman, navigate to **REST > Query**.

```
/services/data/v58.0/query/?q=<query>
```

For a description of EventRelayFeedback non-system fields, see [EventRelayFeedback](#) in the *Object Reference for the Salesforce Platform*.

To get all the fields of EventRelayFeedback, run this query.

```
SELECT FIELDS (ALL) FROM EventRelayFeedback LIMIT 200
```

Example of a REST query call.

```
/services/data/v58.0/query/?q=SELECT+FIELDS (ALL)+from+EventRelayFeedback+LIMIT+200
```

If you aren't interested in the system fields returned, use this query, which specifies non-system fields in the SELECT clause.

```
SELECT EventRelayNumber, EventRelayConfigId, LastRelayedEventTime, RemoteResource, Status,  
ErrorCode, ErrorMessage, ErrorTime, ErrorIdentifier FROM EventRelayFeedback
```

Example query response received using the REST query resource.

```
{  
    "totalSize": 1,  
    "done": true,  
    "records": [  
        {  
            "attributes": {  
                "type": "EventRelayFeedback",  
                "url": "/services/data/v58.0/sobjects/EventRelayFeedback/7k4RM  
0000004LjjYAE"  
            },  
            "EventRelayNumber": "00000004",  
            "EventRelayConfigId": "7k2RM0000004LoAYAU",  
            "LastRelayedEventTime": "2023-04-03T19:23:23.000+0000",  
            "RemoteResource": "aws.partner/salesforce.com/00DRM00000G2tq2AC/0  
YLRM00000001es4AA",  
            "Status": "RUNNING",  
            "ErrorCode": null,  
            "ErrorMessage": null,  
        }  
    ]  
}
```

```
        "ErrorTime": null,  
        "ErrorIdentifier": null  
    }  
]  
}
```

The fields returned in the query correspond to fields in the UI. For example, **RemoteResource** corresponds to the **Partner Event Source Name** field, and **LastRelayedEventTime** corresponds to the **Last Relayed** field in the event relay detail page. For more information, see [Event Relays in Setup](#).

Send Events from AWS Back to Salesforce Using an API Destination

An API destination is the target of a rule, and it enables you to route events to HTTP endpoints. Specifically, you can use an API destination to publish an event to Salesforce from EventBridge using REST API.

[Create a Connected App in Salesforce for OAuth](#)

To use OAuth to authorize the API call to Salesforce, set up a connected app in Salesforce. The connected app generates a consumer key and secret that you can store in the EventBridge connection in AWS.

[Create an API Destination and Connection in EventBridge](#)

Create an API destination that uses a new connection. In this case, the API destination is a REST API call to Salesforce to publish a platform event back. The connection contains the authentication information for the API call. The connection accepts several authorization methods.

[Create an EventBridge Rule and Connect it to the API Destination](#)

The EventBridge rule routes events from the event bus to the API destination, which results in making a REST call to publish an event back to Salesforce.

[Verify the API Destination](#)

Publish a test event to the event bus that is associated with the rule you created in a previous step, and verify that your subscriber in Salesforce receives the event from Amazon EventBridge.

Create a Connected App in Salesforce for OAuth

To use OAuth to authorize the API call to Salesforce, set up a connected app in Salesforce. The connected app generates a consumer key and secret that you can store in the EventBridge connection in AWS.

To create a connected app in Salesforce:

1. From Setup, enter *External Client Apps* in the Quick Find box, then select **Settings** in the External Client App section.
2. Turn on **Allow creation of connected apps**, if it's off.
3. Click **New Connected App**.
4. Enter a name.

5. Enter your contact email.
 6. Optionally, fill out other fields as outlined in [Configure Basic Connected App Settings](#).
 7. In the API (Enable OAuth Settings) section, select **Enable OAuth Settings**.
 8. Select **Enable for Device Flow**.

A callback URL isn't used in the device flow. However, when this flow is enabled, the value for the callback URL defaults to a placeholder.
 9. For OAuth scopes, select **Manage user data via APIs**.
 10. Keep **Require Secret for the Web Server Flow** selected. This option requires the app's client secret in exchange for an access token.
 11. Keep **Require Secret for Refresh Token Flow** selected. This option requires the app's client secret in the authorization request of a refresh token and the hybrid refresh token flow.
 12. Select **Enable Client Credentials Flow**.
 13. When you understand the security risks, accept the warning.
 14. Save your changes.
 15. Click **Continue**.
 16. Click **Manage Consumer Details**.

A new window opens and a verification code is sent to your registered email address.
 17. After you verify your identity, note the consumer key and consumer secret.
 18. Select an execution user for the client credentials flow.

Although there's no user interaction in the client credentials flow, Salesforce still requires you to specify an execution user. By selecting an execution user, you allow Salesforce to return access tokens on behalf of this user.
-  **Note** Permitted Users policies, such as **All users may self-authorize** and **Admin approved users are pre-authorized**, don't apply to the execution user.
- a. From Setup, in the Quick Find box, enter *Apps*, and then select **Manage Connected Apps**.
 - b. Click the connected app you just created.
 - c. Click **Edit Policies**.
 - d. Under Client Credentials Flow, for Run As, click  and find the user who you want to assign the client credentials flow.

For Enterprise Edition orgs, we recommend that you select an execution user who has the API Only User permission.
 - e. Save your changes.

After you create the connected app, it can take up to 10 minutes for the connected app to be ready for use.

See Also

- [Configure a Connected App for the OAuth 2.0 Client Credentials Flow](#)
- [Connected Apps](#)

Create an API Destination and Connection in EventBridge

Create an API destination that uses a new connection. In this case, the API destination is a REST API call to Salesforce to publish a platform event back. The connection contains the authentication information for the API call. The connection accepts several authorization methods.

Prerequisites:

To use OAuth authorization for the API destination connection, create a connected app in Salesforce by following the steps in [Create a Connected App in Salesforce for OAuth](#). You use the consumer key and secret from the connected app for the API destination connection.

1. To create an API destination in the EventBridge console, follow the steps in [Create an API destination](#) in the AWS documentation. Then set up the configurations that are specific to Salesforce.
2. Get your Salesforce org's My Domain name, which is on the My Domain page in Setup. You use the name in the endpoint in the next step.
3. For the event that you want to send to Salesforce, use an existing custom platform event or define a new one. For example, you can create a platform event with the label Carbon Estimate (API name `Carbon_Estimate__e`). To learn how to add fields to a platform event, see [\(Optional\) Define a Platform Event](#). Add the same fields as the ones in the Carbon Comparison event. Also, add this field. Field type: Number (3,0), Label: Carbon Reduction Percentage
4. For API destination endpoint, use this URL after replacing `MyDomainName` with your org's domain name and `MyEvent__e` with the API name of the platform event to return:
`https://MyDomainName.my.salesforce.com/services/data/v58.0/sobjects/MyEvent__e`
For example, for the Carbon Estimate event, the URL is:
`https://MyDomainName.my.salesforce.com/services/data/v58.0/sobjects/Carbon_Estimate__e`
5. For HTTP method, select **POST**.
6. For the connection, select **Create a new connection**, and enter a connection name.
7. For Destination type, select **Partners**, and then select **Salesforce** from Partner Destinations.
If you're setting up OAuth authorization, provide this information for your Salesforce org.
8. For Authorization type, make sure that **OAuth Client Credentials** is selected.
9. For Authorization endpoint, replace the populated endpoint with one of these options.
 - a. If using a production org, enter this URL, and replace `MyDomainName` with your org's My Domain name: `https://MyDomainName.my.salesforce.com/services/oauth2/token`
 - b. If using a sandbox without enhanced domains, enter this URL, and replace `MyDomainName` with your org's My Domain name and `SandboxName` with your sandbox name:
`https://MyDomainName--SandboxName.my.salesforce.com/services/oauth2/token`
 - c. If using a sandbox with enhanced domains, enter this URL, and replace `MyDomainName` with your org's My Domain name and `SandboxName` with your sandbox name:
`https://MyDomainName--SandboxName.sandbox.my.salesforce.com/services/oauth2/token`
10. For HTTP method, select **POST**.
11. For Client ID, enter the consumer key from the connected app in Salesforce.
12. For Client secret, enter the consumer secret from the connected app in Salesforce.
13. Add OAuth HTTP parameters.
 - a. Parameter: **Body field**, Key: `grant_type`, Value: `client_credentials`

 **Note** If your Salesforce org uses multi-factor authentication (MFA) for API access, users must complete a second authentication challenge to access Salesforce APIs. For more information, see [Set Multi-Factor Authentication Login Requirements for API Access](#).

After you create the API destination, you can create a rule whose target is the destination. See [Create an EventBridge Rule and Connect it to the API Destination](#).

See Also

[AWS Documentation: API Destinations](#)

[AWS Compute Blog: Using API destinations with Amazon EventBridge](#)

[Platform Events Developer Guide: Publish Event Messages with Salesforce APIs](#)

Create an EventBridge Rule and Connect it to the API Destination

The EventBridge rule routes events from the event bus to the API destination, which results in making a REST call to publish an event back to Salesforce.

Prerequisites:

Create an API destination in [Create an API Destination and Connection in EventBridge](#).

For more information about EventBridge rules, see [Creating Amazon EventBridge rules that react to events](#) in the AWS documentation.

1. In Amazon Eventbridge, click **Rules**.
2. Select an event bus from the dropdown that is the source of the events that you want to send to Salesforce.
3. In the Rules section, click **Create rule**.
4. Provide a name for your rule.
5. Click **Next**.
6. Under Event pattern, select **Custom patterns (JSON editor)**, and enter a filter. If you don't want to be specific with the filter, use a filter that matches anything except events whose source field is `"dontSend"`.

```
{  
  "source": [ {  
    "anything-but": ["dontSend"]  
  } ]  
}
```

For more information about event pattern matching, see [Content filtering in Amazon EventBridge event patterns](#) in the AWS documentation.

7. Click **Next**.
8. In Select targets, under Target 1, select **EventBridge API destination**.
9. From the dropdown, select the API destination that you just created.
10. Expand **Additional settings**.
11. Select **Part of the matched event**, and provide the part of the event message to pass to the API destination. This step prevents the top-level Amazon event fields from being sent to Salesforce. Only the part containing the Salesforce event fields from the detail section of the original event are sent.
 - a. If the event format is an EventBridge event, provide this value: `$.detail`

- b. If the event originates from a Lambda function, provide a path in the Lambda function result message. For example, to return a section from the function response, use `$.detail.responsePayload.{responseSection}`.
12. Click **Next** and then **Next**.
13. Review the rule, and then click **Create rule**.



Tip Troubleshooting Tip: To troubleshoot the execution of the API destination, you can add a dead letter queue to the target. The dead letter queue is an Amazon SQS queue that receives the messages that couldn't be delivered along with the errors. From the Amazon SQS console, you can poll messages in the queue to view the messages and errors. For more information, see [Event retry policy and using dead-letter queues](#) and [Receiving and deleting messages \(console\)](#) in the AWS documentation.

Verify the API Destination

Publish a test event to the event bus that is associated with the rule you created in a previous step, and verify that your subscriber in Salesforce receives the event from Amazon EventBridge.

1. Start your event subscriber. For example, in Salesforce, you can create an Apex trigger that subscribes to the sample `Carbon_Estimate__e` event or another event that you're using. To create a platform event trigger, see [Subscribe to Platform Event Notifications with Apex Triggers](#) in the *Platform Events Developer Guide*.
2. In Amazon EventBridge, from the Event buses page, click the event bus associated with the rule and API destination.
3. In the event bus page, copy the event bus ARN.
4. Click **Send events**.
5. For Event source, enter the ARN that you copied earlier.
6. For Detail type, enter the platform event API name. For example, `Carbon_Estimate__e`.
7. For Event detail, provide the fields and values in JSON format. For example, this event message contains two fields for the `Carbon_Estimate__e` event.

```
{  
    "Current_Vehicle__c": "Fast Car",  
    "Carbon_Reduction_Percentage__c": 33  
}
```

8. Click **Send**.
9. In your subscriber, verify that the event was received from EventBridge.

Event Studio Overview

Salesforce Eventing encompasses Platform Events and Change Data Capture. Event Studio allows you to track existing event publishing and consumption activity on your Salesforce Event Bus based on event publishers and event subscribers.

The Event Studio page demystifies event activity with a clear view into how events operate within your Salesforce org. It provides tools to assess the health of your org's eventing integrations and proactively identify and debug potential production issues.

Event Publishers: These entities generate and send events onto the Salesforce Event Bus. Understand how various components, such as flows, Apex, and different API calls (Pub/ Sub, REST, Bulk, SOAP), contribute to your event publish entitlements.

Event Subscribers: These entities listen for and consume events from the Event Bus. Monitor how external subscribers, such as Event Relays, Streaming API, Pub/Sub API, and Lightning Web Components count against your event delivery entitlements. You can also view delivery usage from the Apex internal subscriber. Note that Apex events don't count against your delivery allocation.

With visibility into your eventing ecosystem, you can,

- Quickly spot anomalies such as publish spikes, slow subscribers, or clients caught in resubscribe loops, enabling timely debugging of production issues.
- Gain a clear understanding of your usage against entitlements, which helps you manage resources better and facilitates clearer conversations about add-on licenses.
- Have greater confidence and trust in eventing services.

[View Event Publishers and Subscribers](#)

Track existing event publishing and consumption metrics on your Salesforce Event Bus based on event publishers and event subscribers.

View Event Publishers and Subscribers

Track existing event publishing and consumption metrics on your Salesforce Event Bus based on event publishers and event subscribers.

REQUIRED EDITIONS

Available in both Salesforce Classic and Lightning Experience

Available in: **Performance, Unlimited, Enterprise, and Developer** Editions

USER PERMISSIONS NEEDED

To create and edit platform event definitions: Customize Application

The Event Publishers tab uses these fields to provide a list of entities generating and sending events onto the Salesforce Event Bus, so you can see how various components contribute to your event publish entitlements.

- Publish Method: The method used to publish the event, such as Apex, flow, system, Pub/Sub API, REST API, Bulk API, or SOAP API.
- Number of Events Published: The count of events each publisher sent, in a time series or as a total for

the last 24 hours.

- Last Publish Time: The timestamp of the method's most recent event publish in your local time zone.

The Event Subscribers tab uses these fields to offer insights into which clients listen for and consume events from the Event Bus, helping you monitor usage against your event delivery entitlements.

- Subscription Method: The method used to subscribe to events, such as Apex, Pub/Sub API, streaming API, Lightning web component, or Event Relay.
- Number of Clients (Subscriber ID): The number of distinct subscriber clients for this subscription method.
- Number of Events Delivered: The count of events delivered to the subscriber.
- Last Delivery Time: The timestamp of the delivery method's most recent event delivery in your local time zone.

To view event metrics:

1. From Setup, enter *Events* in the Quick Find box, then select **Event Studio**.
2. To view a list of entities sending events, click **Event Publishers**.
3. To view a list of clients consuming events, click **Event Subscribers**.

Sync Data Between Salesforce and Heroku

Heroku Connect lets you sync data between Salesforce and Heroku Postgres.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

Using Heroku Connect with Heroku Postgres, you can build Heroku apps that interact with your Salesforce data using your favorite language, like Ruby, Node.js, Python, PHP, Java, Scala or Clojure or web framework like Rails, Django, or Play. For more information, refer to the [Heroku Connect website](#) and the [Heroku Connect documentation](#) on the Heroku Dev Center website.

Extend Salesforce to External Apps with Lightning Out 2.0

Embed custom Lightning web components (LWC) into external, non-Salesforce apps by using Lightning Out 2.0, a special Salesforce app that you create and configure in Setup. Developers can embed the generated Lightning Out 2.0 app script into a page of your company's external app. Lightning Out 2.0 uses session-based authentication that logs users into Salesforce without leaving the external app, so authenticated users can seamlessly interact with the embedded components as if they were in Salesforce.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, all Agentforce 1 Editions, and Developer Editions**

After you configure some administrative and security settings for the org, use the Lightning Out 2.0 App Manager to create a Lightning Out 2.0 app. Add the domain of the external app that will host your Lightning Out 2.0 app, and then add custom Lightning web components. Next, apply custom styling to the Lightning Out 2.0 app so that the embedded components match the branding of the external app. Paste the generated Lightning Out 2.0 code block into the HTML of the external app, and now your users can engage with Salesforce wherever they are.

 **Tip** For Lightning Out 2.0 documentation aimed at developers, see [Use Components Outside Salesforce with Lightning Out 2.0](#) in the *Lightning Web Components Developer Guide*.

[Prepare to Build a Lightning Out 2.0 App](#)

Before you embed a Lightning Out 2.0 app into an external app, configure some security-related and administrative settings in Salesforce and in your external app. These settings allow Lightning Out 2.0 inline frames (iframes) to load in the external app.

[Set Up Authentication for Lightning Out 2.0](#)

For an external app to access components in a Lightning Out 2.0 app, the end user must be authenticated in Salesforce. The external app must also be authorized to access Salesforce resources. To set up authentication and authorization, implement an OAuth 2.0 flow and configure an External Client App (ECA) in your org. Then configure your external app to either call the Lightning Out 2.0 UI Bridge endpoint or the standard Salesforce UI Bridge endpoint. The call generates a frontdoor URL at run time, which is used to establish a Salesforce session for the Lightning Out 2.0 app.

[Build a Lightning Out 2.0 App](#)

Create and configure Lightning Out 2.0 apps with the low-code Lightning Out 2.0 App Manager. Add custom Lightning web components (LWCs) to each Lightning Out 2.0 app, and optionally define custom property values that your external apps can pass to these components. Generate a Lightning Out 2.0 code block that you can directly embed into an external app's HTML markup.

See Also

[Lightning Web Components Developer Guide: Use Components Outside Salesforce with Lightning Out 2.0](#)

Prepare to Build a Lightning Out 2.0 App

Before you embed a Lightning Out 2.0 app into an external app, configure some security-related and administrative settings in Salesforce and in your external app. These settings allow Lightning Out 2.0 inline frames (iframes) to load in the external app.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, all Agentforce 1 Editions, and Developer Editions**

USER PERMISSIONS NEEDED

To modify My Domain and session security settings: Customize Application

To modify CORS settings: Modify All Data AND Customize Application

Before You Begin

1. Make sure that your external app supports inline frames (iframes).
2. Make sure that your external app is secure and uses HTTPS.
3. Make sure that third-party cookies are allowed in your external app and your users' browsers.

Configure Session Settings

Add the domain names of your external apps to the Trusted Domains allowlist in Session Settings.

1. From Setup, in the Quick Find box, enter *Session Settings*, and then select **Session Settings**.
2. In the Trusted Domains for Inline Frames section of the Session Settings Setup page, click **Add Domain**.
3. Enter the domain of an external app that will host your Lightning Out 2.0 app.
Acceptable URI formats are example.com, https://example.com, and *.example.com.
4. Select **Lightning Out** from the IFrame Type dropdown list.
5. Save your changes.
6. Repeat steps 2-5 for any other domains that will host your Lightning Out 2.0 app.

Enable Third-Party Cookies

Make sure that cross-domain Salesforce session cookies are enabled.

1. From Setup, in the Quick Find box, enter *My Domain*, and then select **My Domain**.
2. In the Routing and Policies section, make sure that the “Require first-party use of Salesforce cookies” setting is deselected.

Configure CORS Settings

Add the domain names of your external apps to the Cross-Origin Resource Sharing (CORS) allowlist.

 **Note** This step is required only if your external app uses the Lightning Out 2.0 UI Bridge access point for client-side authentication. If your external app uses its own server-side authentication flow to generate a Salesforce frontdoor URL, this step isn't required. See [Set Up Authentication for Lightning Out 2.0](#).

1. From Setup, in the Quick Find box, enter **CORS**, and then select **CORS**.
2. To add a domain to the CORS allowlist, click **New**.
3. For Origin URL Pattern, enter the domain of the external app that will host your Lightning Out 2.0 app. The origin URL pattern must include the HTTPS protocol and a domain name.
4. Repeat steps 2 and 3 for any other domains that will host your Lightning Out 2.0 app.
5. In the Cross-Origin Resource Sharing (CORS) Policy Settings section, click **Edit**.
6. Select **Enable CORS for OAuth endpoints**.
7. Save your changes.

See Also

[Set Up Authentication for Lightning Out 2.0](#)

[Specify Trusted Domains for Inline Frames](#)

[Configure Salesforce CORS Allowlist](#)

[Enable CORS for OAuth Endpoints](#)

Set Up Authentication for Lightning Out 2.0

For an external app to access components in a Lightning Out 2.0 app, the end user must be authenticated in Salesforce. The external app must also be authorized to access Salesforce resources. To set up authentication and authorization, implement an OAuth 2.0 flow and configure an External Client App (ECA) in your org. Then configure your external app to either call the Lightning Out 2.0 UI Bridge endpoint or the standard Salesforce UI Bridge endpoint. The call generates a frontdoor URL at run time, which is used to establish a Salesforce session for the Lightning Out 2.0 app.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, all Agentforce 1 Editions, and Developer Editions**

USER PERMISSIONS NEEDED

To create local External Client Apps:	Create, edit, and delete External Client Apps
To configure External Client Apps OAuth settings:	Create, edit, and delete External Client Apps
To modify CORS settings:	Modify All Data AND Customize Application
To create and modify a Lightning Out 2.0 app:	Customize Application

Before you begin, complete the steps in [Prepare to Build a Lightning Out 2.0 App](#).

1. Design and implement a supported [OAuth 2.0 authorization flow](#).

Important Lightning Out 2.0 supports OAuth 2.0 flows that provide a web-scoped access token with user context, including the [web server flow](#) and the [JSON Web Token \(JWT\) bearer flow](#). The client credentials flow isn't supported because it doesn't provide user context, so Salesforce filters out the web scope from its access token.

2. In your org, create an external client app (ECA) with OAuth enabled.

Use these instructions to get set up: [Create an External Client App](#) and [Configure the External Client App OAuth Settings](#).

3. In your ECA, use these OAuth settings.

- Include the **Manage user data via Web browsers (web)** or the **Full access (full)** OAuth scopes.

If your ECA uses the JWT bearer flow, also include the **Perform requests at any time (refresh_token)** scope.

- If your ECA uses one of the listed flows in the Flow Enablement section, enable and configure the settings for that flow.

Remember that Lightning Out 2.0 doesn't support the client credentials flow.

- In the Security section, you can keep these default settings selected: **Require secret for Web Server Flow**, **Require secret for Refresh Token Flow**, and **Require Proof Key for Code Exchange (PKCE extension for Supported Authorization Flows)**.

Lightning Out 2.0 also supports refresh token rotation and JWT-based access tokens.

4. To obtain a frontdoor URL at run time, call a UI Bridge API endpoint from your external app by using one of these methods.

- [Obtain a frontdoor URL from the Lightning Out 2.0 endpoint](#).

This method provides only access to Lightning Out 2.0 resources. It requires almost no extra setup and provides built-in Cross-Origin Resource Sharing (CORS) protection. To use this method, you must add your external app domain name to the Salesforce CORS allowlist. See [Prepare to Build a Lightning Out 2.0 App](#).

- [Obtain a frontdoor URL from the standard UI Bridge API endpoint](#).

This method requires you to configure your own server-side authorization flow for your external app. However, this method allows for more fine-grained control if your company has specific security requirements.

Obtain a Frontdoor URL from the Lightning Out 2.0 Endpoint

The Lightning Out 2.0 servlet extends the UI Bridge API by providing CORS support and restricts Salesforce resource access to only Lightning Out 2.0 apps.

Instead of the standard UI Bridge API endpoint `services/oauth2/singleaccess`, the Lightning Out 2.0 servlet uses the special endpoint `services/oauth2/lightningoutsingleaccess`. This endpoint directs to a Lightning Web Runtime app that sets a valid Salesforce session ID (SID) as a cookie. The SID is sent with all future requests to the org, so the user doesn't need to reauthenticate while using the Lightning Out 2.0 app.

! **Important** Before you make a request to the Lightning Out 2.0 UI Bridge API endpoint, make sure that your Lightning Out 2.0 app is enabled from the Lightning Out 2.0 App Manager. See [Build a Lightning Out 2.0 App](#).

To generate a frontdoor URL, from your external app, send a POST request to the

`https://MyDomainName.my.salesforce.com/services/oauth2/lightningoutsingleaccess` endpoint. Include your Lightning Out 2.0 app ID and the OAuth 2.0 access token as request body parameters.

Request body parameter	Description
<code>lightning_out_app_id</code>	<p>Required. The Lightning Out 2.0 app ID. You can find this ID on the Lightning Out 2.0 App Manager page for the app. This ID corresponds to the value of the <code>app-id</code> attribute on the <code>lightning-out-application</code> web component. For more information about the Lightning Out 2.0 app component, see Understand Lightning Out 2.0 Architecture in the <i>Lightning Web Components Developer Guide</i>.</p>
<code>access_token</code>	<p>Required. The valid OAuth 2.0 access token from the user's initial session. It can be an opaque access token or a JSON Web Token (JWT)-based access token. The access token must have the <code>web</code> or <code>full</code> scope. Specify scopes in the external client app settings for the app that issued the access token. The user associated with the access token can't have the API Only User permission.</p>

Here's an example POST request that uses cURL.

```
curl -X POST \
  'https://MyDomainName.my.salesforce.com/services/oauth2/lightningoutsingleaccess' \
--form 'access_token="access-token-value"' \
--form 'lightning_out_app_id="app-id"'
```

Here's another example POST request that uses JavaScript. Make sure that the `Content-Type` header is

declared as `application/x-www-form-urlencoded`.

```
fetch(
  "https://MyDomainName.my.salesforce.com/services/oauth2/lightningoutsingle
access",
{
  method: "POST",
  headers: {
    "Content-Type": "application/x-www-form-urlencoded"
  },
  body: "access_token=access-token-value&lightning_out_app_id=app-id"
}
)
.then((response) => {
  console.log("Status:", response.status);
  return response.json();
})
.then((data) => {
  console.log("Success:", data);
})
.catch((error) => {
  console.error("Error:", error);
});
```

If successful, the UI Bridge API returns a JSON response with the frontdoor URL.

```
HTTP/1.1 200 OK

Content-Type: application/json
Cache-Control: no-cache,must-revalidate,max-age=0,no-store,private
.....

{"frontdoor_uri":"https://MyDomainName.my.salesforce.com/secur/frontdoor.jsp?o
tp=otp_value"}
```

If unsuccessful, the API returns an error message. For error message descriptions, see [Generate a Frontdoor URL to Bridge into UI Sessions](#).

When the frontdoor URL is returned, set the `frontdoor-url` attribute on the `lightning-out-application` component to the `frontdoor_uri` value. Make sure that the `frontdoor-url` attribute value is a string of only the URL, not the entire JSON response.

 **Tip** To review what HTML elements comprise a Lightning Out 2.0 app, including the `lightning-out-application` component definition, see [Understand Lightning Out 2.0 Architecture](#).

After the `frontdoor-url` attribute is set, a Lightning Web Runtime app loads and sets a valid Salesforce session ID. Your Lightning Out 2.0 app now loads on the external host page.

Obtain a Frontdoor URL from the Standard UI Bridge API Endpoint

Instead of using the Lightning Out 2.0 UI Bridge API endpoint, you can obtain a frontdoor URL from the standard UI Bridge API endpoint `services/oauth2/singleaccess`. This method requires you to configure your own server-side authorization flow for your external app. However, this method allows for more fine-grained control if your company has specific security requirements.

-  **Note** Unlike the Lightning Out 2.0 endpoint, the standard UI Bridge API endpoint doesn't support CORS settings configured in Salesforce Setup.

For instructions, see [Generate a Frontdoor URL to Bridge into UI Sessions](#). In your request, don't include the `redirect_uri` parameter. Instead of being taken to the Salesforce homepage, users see the Lightning Out 2.0 app components on the external host page.

When the frontdoor URL is returned, set the `frontdoor-url` attribute on the `lightning-out-application component` to the `frontdoor_uri` value. Make sure that the `frontdoor-url` attribute value is a string of only the URL, not the entire JSON response.

After the `frontdoor-url` attribute is set, a Lightning Web Runtime app loads and sets a valid Salesforce session ID. Your Lightning Out 2.0 app now loads on the external host page.

See Also

- [External Client Apps](#)
- [Authorize Apps with OAuth](#)
- [Generate a Frontdoor URL to Bridge into UI Sessions](#)
- [Lightning Web Components Developer Guide: Understand Lightning Out 2.0 Architecture](#)

Build a Lightning Out 2.0 App

Create and configure Lightning Out 2.0 apps with the low-code Lightning Out 2.0 App Manager. Add custom Lightning web components (LWCs) to each Lightning Out 2.0 app, and optionally define custom property values that your external apps can pass to these components. Generate a Lightning Out 2.0 code block that you can directly embed into an external app's HTML markup.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, all Agentforce 1 Editions, and Developer Editions**

USER PERMISSIONS NEEDED

To create or modify a Lightning Out 2.0 app: Customize Application

Before you begin, complete the steps in [Prepare to Build a Lightning Out 2.0 App](#) and [Set Up Authentication for Lightning Out 2.0](#).

! **Important** Lightning Out 2.0 currently supports only custom Lightning web components. Lightning base components and Aura components aren't supported. See [Lightning Out 2.0 Limitations](#) in the *Lightning Web Components Developer Guide*.

Create a Lightning Out 2.0 App

1. From Setup, in the Quick Find box, enter *Lightning Out*, and then select **Lightning Out 2.0 App Manager**.
2. Click **New Lightning Out 2.0 App**.
3. Enter a unique name for your Lightning Out 2.0 app.
The name can contain up to 64 letters, digits, and special characters. It must contain at least one letter. Names consisting of only digits or special characters aren't allowed. You can change your app name at a later time in the App Settings tab.
4. Click **Save and Continue**.
The Edit Lightning Out 2.0 page opens.
5. In the App Settings tab, to activate the app, set the App Status to **Enabled**.
If the App Status is **Disabled**, then user authentication fails, and the embedded components don't load in the external app.
6. Open the Host Page Domain Names tab, and then enter the domain names of any external app that will host your Lightning Out 2.0 app.
Include the https:// protocol, and remove the trailing slash (/) at the end of the domain name.
To add multiple domains, click **Add Domain Name** after entering each domain name.
7. To add components to your app, open the App Components tab. Then follow steps 3-5 in the [Configure Components for a Lightning Out 2.0 App](#) section.
8. Save your changes.

Configure Components for a Lightning Out 2.0 App

1. From the Lightning Out 2.0 App Manager, select **Edit App** from the actions dropdown next to the Lightning Out 2.0 app name.
The Edit Lightning Out 2.0 App page opens the App Settings tab.
2. From the Edit Lightning Out 2.0 App page, open the App Components tab.
3. In the Component Name field, enter the name of the custom component. Then click **Add Component**.
The component must already exist in your org. Use a forward slash (/) or a hyphen (-) to separate the namespace from the component name. If the component doesn't have a custom namespace, include the default  namespace.

For example, `complexNs/lwcComponent` and `complex_ns-lwc-component` are equivalent, valid formats.

4. Optionally, to pass styles and properties from your app to an embedded component, click  in the Custom Properties cell for that component. Enter the properties and their values in JSON format. For example, to pass CSS custom properties, enter `{"style": "--custom-color: #000000;"}`. Lightning Out 2.0 currently supports overrides for the `style` property and other properties available to the component. See [Set Component Styles and Properties in a Lightning Out 2.0 App](#).
5. Repeat steps 3 and 4 for any other components that you want to add to your Lightning Out 2.0 app.
6. Save your changes.

Embed a Lightning Out 2.0 App into an External App

1. From the Lightning Out 2.0 App Manager, click the ID of the Lightning Out 2.0 app.
2. To copy the generated Lightning Out 2.0 code block into your external app, click **Copy to Clipboard**. The generated Lightning Out 2.0 code block doesn't contain the `frontdoor_url` attribute on the `lightning-out-application` component. Instead, you set this attribute at run time by calling a UI Bridge API endpoint. See [Set Up Authentication for Lightning Out 2.0](#).
3. Paste the Lightning Out 2.0 app code block into the HTML markup of a page in your external app.

At run time, the components in your Lightning Out 2.0 app render at the location where you embedded the code block. To change where a component renders on the page, move it to the desired location.

For example, to render `c-other-lwc` above `c-my-lwc` on the external app page, adjust the markup so that the `c-other-lwc` element is above the `c-my-lwc` element. You can mix the Lightning Out 2.0 app component markup with external page elements.

```
<--- External App Page --->
<html>
  <body>
    <external-page-element-1></external-page-element-1>
    <script
      type="text/javascript"
      async=""
      src="https://MyDomain.my.salesforce.com/lightning/lightning.out.latest/index.iife.prod.js">
    </script>
    <lightning-out-application
      frontdoor-url="Set at run time"
      app-id="18-digit Salesforce ID"
      components="c-my-lwc, complex_ns-lwc-component">
    </lightning-out-application>
    <c-other-lwc></c-my-lwc>
    <external-page-element-2></external-page-element-2>
```

```
<c-my-lwc style="--custom-color: brown;"></c-my-lwc>
<external-page-element-3></external-page-element-3>
</body>
</html>
```

4. Alternatively, embed your Lightning Out 2.0 app programmatically by using [Document Object Model \(DOM\) interface JavaScript methods](#). Copy the element names and attribute values from the generated Lightning Out 2.0 code block.

For a JavaScript implementation, see the example in [Set Component Styles and Properties in a Lightning Out 2.0 App](#).

See Also

[Set Up Authentication for Lightning Out 2.0](#)

[Lightning Web Components Developer Guide: Understand Lightning Out 2.0 Architecture](#)

[Lightning Web Components Developer Guide: Set Component Styles and Properties in a Lightning Out 2.0 App](#)

Build Your Own Web Site

Site.com and Salesforce Sites are legacy systems to create sites using Salesforce.

Use [Experience Cloud](#) to create scalable sites using the latest Salesforce web technologies.

[Site.com](#)

Site.com is a web content management system (CMS) that makes it easy to build dynamic, data-driven web pages quickly, edit content in real time, and manage your websites.

[Salesforce Sites](#)

Salesforce Sites enables you to create public websites and applications that are directly integrated with your Salesforce organization—without requiring users to log in with a username and password. You can publicly expose any information stored in your organization through a branded URL of your choice. And you can make the site's pages match the look and feel of your company's brand.

Site.com

Site.com is a web content management system (CMS) that makes it easy to build dynamic, data-driven web pages quickly, edit content in real time, and manage your websites.

REQUIRED EDITIONS

Available in: [Salesforce Classic](#)

Available for purchase in: [Enterprise](#), [Performance](#), and [Unlimited](#) Editions

Available (with limitations) in: **Developer Edition**

-  **Note** If you're a new customer who wants to create a site, portal, or community, Communities are a great way to share information and collaborate with people outside your company, such as customers, partners, or employees. See [Experience Cloud](#) to learn more.

From the [Site.com tab](#) in the Site.com app, you can launch Site.com Studio, which provides a separate, dedicated environment for creating and editing pixel-perfect, custom websites. Site administrators and designers can create and style web pages, and add features such as navigation menus, images, and text areas using drag-and-drop page elements, while ensuring the site's pages match the look and feel of the company's brand. And content contributors, such as marketing users, can browse and update website content directly in a simplified Site.com Studio environment. Additionally, websites built with Site.com benefit from running on Salesforce's trusted global infrastructure.

-  **Note** The features available in Site.com Studio vary depending on whether you're a [site administrator, designer, or contributor](#).

These examples illustrate a few ways to use Site.com:

- Create an event site—Advertise upcoming events, such as grand openings, launches, or sales kick-offs on a public event site.
- Promote new products—Launch new products into the market with a promotional website that helps drive sales.
- Publish a support FAQ—Provide helpful information on a public website where customers can view solutions to their issues.
- Create microsites and landing pages—Create temporary landing pages or targeted microsites for marketing campaigns.
- Create a recruiting website—Post job openings to a public site and allow visitors to submit applications and resumes.
- Publish a catalog of products—List all of your company's products on a public website, with model numbers and current prices pulled dynamically from your organization.
- Post company press releases—Publish your company's press releases and sort by publication date.

System Requirements

To use Site.com Studio, we recommend:

- Mozilla® Firefox® or Google® Chrome for best performance.
-  **Note** Microsoft® Edge is supported, but it is strongly recommended that you do not use Internet Explorer®.
- Disabling the Firebug extension for Firefox, if installed, as it can impact performance.
 - A minimum browser resolution of 1024x768.

[About Site.com Feature Licenses](#)

To access Site.com, each user in your organization must have a Site.com feature license.

[Plan and Implement a Site.com Website](#)

There are many approaches to building a website. The process that best suits your needs depends on many factors, such as the size of your team and the tasks you're responsible for. If you're a site administrator or designer, you're involved in every stage, including adding and maintaining the site's content. Alternatively, you can have contributors who add, edit, and maintain this content. And if you're a contributor, you can be responsible for editing and updating all of the site's content, or you can work with other contributors, designers, and site administrators to bring the site to completion.

There are various stages involved in creating a site with Site.com.

[Create a Site.com Community](#)

Each community has one associated Site.com site that lets you add custom, branded pages to your community. By default, Site.com pages are publicly available and don't require login, but you can also create private pages that only community members can access.

[Create a Site.com Site](#)

To get started with Site.com, create a new blank site.

[Import and Manage Assets](#)

Contributors, publishers, and site administrators can import a variety of assets, such as images, HTML pages, and PDFs, to use in a website. You can import assets and files individually, or use a zipped file. When importing entire websites or large numbers of assets, it's easier to create a zipped file of the content with the desired folder structure. When importing the zipped file for a website, Site.com recreates your website and places everything in the same folder structure.

[Edit Site.com Pages as a Designer or Site Administrator](#)

When working with page templates and site pages, you can add content, structure, and style, all in one place.

[Site.com Page Elements](#)

Page elements are the building blocks of your site pages and page templates. Combined, they provide the page's structure and content.

[Set Up the Contributor's Studio View](#)

Control what your contributors can do in Site.com Studio.

[Cascading Style Sheets Overview](#)

Cascading Style Sheets (CSS) provide a flexible way to add style to the pages of your website. This collection of formatting rules governs the appearance of your pages, and lets you define the fonts, colors, layout, and other presentation features.

[Site Branding Overview](#)

Branding provides a flexible way for you to define different aspects of your website's brand. Once branding properties are defined, your editors can easily customize everything in one centralized place, the Branding Editor. When your website editors customize the properties, they get a preview of their branding changes immediately.

[Custom Site Properties Overview](#)

With custom site properties, you can define and store frequently occurring content on your site. For example, you can store your address and phone number as a custom property so that it can be reused by anyone who is editing your site. You can apply stored properties to pages, headers and footers, and widgets quickly by using expression language syntax.

[Site.com Data Services Overview](#)

Site.com data services combine many features that let you connect to standard and custom Salesforce objects. Retrieve data from your organization's objects and dynamically display it on your site pages, or alternatively, gather and submit data from your customers. And when you update data in your Salesforce object, the changes are reflected automatically on the live site—no site updates required!

[Widgets Overview](#)

Widgets let you save time by building custom page elements that you and your team can reuse throughout the site.

[Multilingual Sites Overview](#)

Site.com Studio lets you create different language versions of your site. And because all languages are maintained within the site, you don't need to create and manage a separate site for each language.

[Content Lists and Categories Overview](#)

Content lists let you create and store content items to use in your site, such as press releases, blog posts, and news articles. *Categories* let you create groups of terms to classify the items in your content lists.

[Events Overview](#)

Events enable you to add interactive and animated effects to the pages and page elements of your website.

[The Contributor's Page Editing View](#)

Use the Overview tab to import assets, preview the page, and update the appearance of the page.

[Preview How Pages Appear on Mobile Devices](#)

With live mode, site administrators, designers, and contributors can preview how pages and templates appear on devices such as mobile phones and tablets.

[Preview Site.com Sites](#)

Contributors, designers, and site administrators can preview site pages to see how they look when rendered in a browser window. It's always a good idea to make sure your changes are displayed correctly, as this preview is how the pages appear on the live site.

[Site.com IP Restrictions Overview](#)

Every computer has a unique IP address that it uses to identify itself. Using IP restrictions, you can define a range of permitted IP addresses for the pages, folders, and assets in your site to control visitors' access.

[Manage Domains in Site.com](#)

Before you can publish your site to the Internet, you must set the site's domain information.

See Also

[Set Up Site.com Users](#)

[Plan and Implement a Site.com Website](#)

[Using Site.com Studio as a Site Administrator or Designer](#)

[Using Site.com Studio as a Contributor](#)

About Site.com Feature Licenses

To access Site.com, each user in your organization must have a Site.com feature license.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

- The Site.com Publisher feature license allows the user to access Site.com Studio to create and style websites, control the layout and functionality of pages and page elements, and add and edit content.
- The Site.com Contributor feature license allows the user to access Site.com Studio to edit site content only.

Consider what your users need to do in a site and purchase feature licenses accordingly. See the [Site.com feature table](#) for a complete list of the capabilities that come with each feature license. After you purchase feature licenses, you can [set up Site.com users](#).

You can view the number of assigned feature licenses on your organization's profile.

-  **Note** Organizations using Performance, Unlimited, or Enterprise Editions must purchase Site.com Publisher and Site.com Contributor feature licenses separately. Additionally, a Site.com Published Site license is required for each site that's published to the Internet. For information on purchasing Site.com licenses, contact Salesforce. Developer Edition organizations contain two Site.com Publisher feature licenses and one Site.com Contributor feature license. Developer Edition organizations can't publish sites. Communities users with the "Create and Set Up Communities" permission are assigned the role of site administrator in a community's Site.com site. To let users who don't have this permission edit the site, you must purchase and assign a Site.com Publisher or a Site.com Contributor feature license. Then you must assign a user role at the site level.

[Set Up Site.com Users](#)

Before users can access Site.com, you must allocate a Site.com feature license to each user and assign a user role at the site level. The features available when editing a site in Site.com Studio vary depending on these settings.

[About Site.com User Roles](#)

Each Site.com user must have a user role assigned at the site level, which controls what each user can do in a site. Users can have only one role per site, but their roles can vary between sites. For example, a person can be a site administrator on one site and a contributor on another.

[Manage Site.com Users and Roles](#)

After you create a site, you can add other users and assign roles to them.

[About Site.com Features](#)

>Site.com Studio features vary depending on feature license, user role, and permissions.

See Also

[About Site.com User Roles](#)

[Manage Site.com Users and Roles](#)

Set Up Site.com Users

Before users can access Site.com, you must allocate a Site.com feature license to each user and assign a user role at the site level. The features available when editing a site in Site.com Studio vary depending on these settings.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To create or edit users:	Manage Internal Users
--------------------------	-----------------------

First, review the [Site.com feature table](#) for detailed information on the feature license, permissions, and role required for each user.

After you've determined the appropriate access level required:

1. Allocate a feature license to the user by editing the user's profile. To allocate:
 - A Site.com Publisher feature license, select the **Site.com Publisher User** checkbox.
 - A Site.com Contributor feature license, select the **Site.com Contributor User** checkbox.After you allocate a feature license, users can access Site.com in the Lightning Platform app menu in the Salesforce header.
 **Note** If the checkboxes don't appear, verify that Site.com is enabled for your organization. See [About Site.com Feature Licenses](#).
2. Ensure the View Setup and Configuration permission is enabled. All users who create or edit websites in Site.com Studio need this permission.
3. Additionally, ensure that at least one user in your organization has both a Site.com feature license and the Manage Users permission. This way, someone can always reallocate user roles if a site's users are accidentally deleted.
 **Warning** The Manage Users permission is powerful. It allows a user to manage all other users in your organization and not just Site.com.
4. [Add users and assign user roles](#) within a site. (When a user with the Site.com Publisher feature license creates a site, the user is automatically allocated the role of site administrator at the site level.)

The feature license, permissions, and user role all determine what a user can do in each site. For example, to create an administrative user who can manage all sites in your organization, assign a Publisher feature license and assign the role of site administrator at the site level.

For users who need only limited access to edit site content, but no administrative access, assign a

Contributor feature license and a contributor role at the site level.

Alternatively, to create a user who can manage roles in a site, but without the ability to publish, assign a Publisher feature license, the Manage Users permission, and the designer role at the site level.

-  **Note** Any records created by unauthenticated guest users via a Site.com Site has the Site Guest User as the record's owner. Each site has one Site Guest User.

See Also

[About Site.com User Roles](#)

About Site.com User Roles

Each Site.com user must have a user role assigned at the site level, which controls what each user can do in a site. Users can have only one role per site, but their roles can vary between sites. For example, a person can be a site administrator on one site and a contributor on another.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

To manage user roles in a site, you must either be the site administrator for that site, or have a Site.com feature license and the “Manage Users” permission.

-  **Note** Users with the “Create and Set Up Experiences” permission are assigned the role of site administrator in a community’s Site.com site. However, they don’t appear in the User Roles section on the Overview tab of Site.com Studio.

Users can have one of three roles at the site level:

- Site administrator—Site administrators are users who can create and manage all site content. They can create sites, templates, style sheets, and pages, and also set up domains, publish sites, and assign user roles. This role requires the Site.com Publisher feature license.
- Designer—Designers have the same control over site content as site administrators, but they can’t manage domains or publish sites. By default, they can’t assign roles unless they have the “Manage Users” permission. This role requires the Site.com Publisher feature license.
- Contributor—Contributors have the most restricted access to content and can typically edit page text and images. By default, they can’t assign roles unless they have the “Manage Users” permission. This role requires the Site.com Contributor feature license.

See the [Site.com feature table](#) for a detailed list of each user role’s capabilities.

See Also

- [Manage Site.com Users and Roles](#)
- [Set Up Site.com Users](#)
- [Set Up the Contributor's Studio View](#)
- [About Site.com Feature Licenses](#)

Manage Site.com Users and Roles

After you create a site, you can add other users and assign roles to them.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To manage Site.com users:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator role assigned at the site level

If you haven't [assigned Site.com feature licenses](#) to your users, you aren't able to add them to a site.

 **Note** Communities users with the "Create and Set Up Communities" permission are assigned the role of site administrator in a community's Site.com site. However, they don't appear in the User Roles section on the Overview tab of Site.com Studio.

When assigning a user role, be sure to add one that's compatible with the user's Site.com license. When users log into Site.com, their licenses are checked against the role assigned to them at the site level. If the license doesn't allow the permissions associated with the role, then the user is given the permissions associated with the license. For example, if a user has a Site.com Contributor feature license, but is assigned a role of site administrator, they only have Contributor permissions regardless of the assigned role.

To add users and assign roles:

1. On the Overview tab in Site.com Studio, click **Site Configuration | User Roles**.
2. Click **Add Users**.
3. In the Available Users section, highlight the user you want to add.

4. Select the role from the Add as dropdown list.
5. Click the arrow to move the user to the Selected Users section.
6. Click **Save**.

To delete users:

1. In the User Roles view, select the user.
2. Click  | Remove.
3. Click **OK**.

To change a user's role:

1. In the User Roles view, hover over the user's role.
2. Click the arrow to display all the roles.
3. Select the new role.

To delete or change the role of a group of users at the same time, use Bulk Actions.

1. In the User Roles view, select the check box beside each user's name.
2. Click **Bulk Actions**.
3. Select the action.
4. Click **Apply**.

 **Note** When updating the roles of several users at once, you can only assign the same role to all selected users.

See Also

- [Set Up Site.com Users](#)
- [Set Up the Contributor's Studio View](#)
- [Plan and Implement a Site.com Website](#)
- [About Site.com Features](#)

About Site.com Features

>Site.com Studio features vary depending on feature license, user role, and permissions.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

The features available when editing a site in Site.com Studio vary depending on your [Site.com feature license](#) and also your [user role](#) for that particular site.

This table lists the required feature licenses, permissions, and roles for many of the Site.com Studio features.

 **Note**

- Communities users with the “Create and Set Up Communities” permission are assigned the role of site administrator in a community’s Site.com site.
- You can’t create, delete, or duplicate community sites in Site.com.

Site.com Studio Feature Requirements			
Feature	Feature License	Site.com Studio User Role	Permissions
Assign feature license to user profile			“Manage Internal Users”
Add users and roles at the site level	Publisher	Site administrator	Additionally, any user with a Site.com feature license and the “Manage Users” permission.
Enable contributors to create pages, add content blocks and widgets, and edit content blocks and graphics	Publisher	Site administrator or designer	
Create websites	Publisher	Users who create a site are automatically added to that site as a site administrator.	
Delete websites	Publisher	Site administrator or designer	
Import websites	Publisher	Users who import a site are automatically added to the new site as a site administrator.	
Export websites	Publisher	Site administrator or designer	
Duplicate websites	Publisher	Site administrator or designer	
Manage domains (Unavailable for Developer Edition)	Publisher	Site administrator	
Add and edit IP restrictions	Publisher	Site administrator	

Site.com Studio Feature Requirements			
Feature	Feature License	Site.com Studio User Role	Permissions
Publish changes to the live website (Unavailable for Developer Edition)	Publisher	Site administrator	
Create page templates	Publisher	Site administrator or designer	
Create website pages	Publisher or Contributor	Site administrator or designer Contributor only if enabled by the site administrator or designer in the page template's Properties pane.	
Create and modify style sheets	Publisher	Site administrator or designer	
Modify layout and design	Publisher	Site administrator or designer	
Add page elements	Publisher or Contributor	Site administrator or designer Contributor can add content blocks and widgets only if enabled by the site administrator or designer.	
Add data repeaters and other data-bound page elements	Publisher	Site administrator or designer	
Modify the Guest User profile to set public access permissions to Salesforce objects	Publisher	Site administrator or designer	“Manage Profiles and Permission Sets” and “Customize Application”
Import assets, such as images and files	Publisher or Contributor	Any assigned role	
Edit content and images	Publisher or Contributor	Site administrator or designer Contributor only if enabled by the site administrator or designer in the page template's Properties pane.	

Site.com Studio Feature Requirements			
Feature	Feature License	Site.com Studio User Role	Permissions
Preview website pages	Publisher or Contributor	Any assigned role	

Plan and Implement a Site.com Website

There are many approaches to building a website. The process that best suits your needs depends on many factors, such as the size of your team and the tasks you're responsible for. If you're a site administrator or designer, you're involved in every stage, including adding and maintaining the site's content. Alternatively, you can have contributors who add, edit, and maintain this content. And if you're a contributor, you can be responsible for editing and updating all of the site's content, or you can work with other contributors, designers, and site administrators to bring the site to completion. There are various stages involved in creating a site with Site.com.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To create or import Site.com sites:

Site.com Publisher User field enabled on the user detail page

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

Site.com Contributor User field enabled on the user detail page

AND

Contributor role assigned at the site level

- Plan the Site Design and Page Layout (Site administrator or designer)–Before building the pages of the site, spend time planning the site design and basic layout. This stage is key to ensuring a consistent look and feel with the minimum amount of effort. From a hierarchical point of view, think about how many pages you need and whether they have subpages. Also consider how you want site visitors to navigate around your site.

Next, plan the layout of the pages and identify the common elements that every page has. In this example, the site has a header section that includes the company's logo and menu (1), and a footer section (2). However, the main section of the home page (3) differs from the rest of the site pages (4). Take note of these similarities and differences, because they affect how you create your site pages.



- Create the Site (Site administrator or designer)–When you've completed the planning stage, you're ready to get started! Log into the Site.com app and go to the [Site.com tab](#), where you can [create your first site](#). Your new blank site opens in [Site.com Studio](#), a powerful environment for building the pages of your site.



Note Only users with the [Site.com Publisher User](#) field enabled on their user detail page can create and import sites.

- Import Assets (Site administrator or designer)–If you're working with a design agency, they can provide all of the files and assets you need, including a CSS style sheet. If you've created your own design, cut up the design and collect the assets, images, and files you plan to use in the site. [Import the assets](#) into Site.com Studio, where they appear in the Assets section of the Overview tab.
- Create a Page Template (Site administrator or designer)–When you've decided on the layout, the quickest and most effective method is to use [page templates](#) to build the basic layout and then base your site pages on it. Try to keep the design of your main page template simple to make it easier to modify in the future. For more complicated site designs, such as the example graphic, you can use the main page template as the basis for a [child template](#) to achieve maximum flexibility. When you [create your page template](#), you can choose from predesigned layouts that include headers, footers, and columns, or you can create a blank page template.
- Lay Out the Page (Site administrator or designer)–After you create the page template, you can [modify the layout](#) further to match the design of your site.
- Create the Site Pages (Site administrator or designer)–Using the template as a base, you can quickly [create the site pages](#), which automatically inherit all the elements of the page template. Or if you need a standalone page that doesn't follow the site's overall design, you can create a blank page instead.
- Add Features and Page Elements (Site administrator or designer)–Use Site.com's prebuilt [page elements](#) to add features such as [navigation menus](#), [images](#), and [data services](#), and include [content blocks](#) that contributors can edit. And add interactive, animated effects using [events and actions](#).
- Make Your Website Look Good (Site administrator or designer)–Take advantage of [cascading style sheets](#) (CSS) to develop the look and feel of your website. If you're not completely up to speed with CSS, the [Style pane](#) provides an easy, visual way to create and manage styles. Or if you're a CSS expert who likes to get straight into the code, you can hand-code the site's [style sheets](#).
- Add and Edit Content (Contributor)–At this stage, if you're a contributor, the site is usually ready for you to add and edit content such as [text](#), [images](#), [videos](#), and [hyperlinks](#). And as you work, you can [upload](#) any images or files you need.
- Review and Test the Site (Contributor, designer, or site administrator)–Testing the changes to the pages of your site happens throughout the development cycle. As a contributor, designer, or site

administrator, you must [preview your changes](#) to ensure they display as expected in a browser. And if you're a site administrator or designer, you can send a preview link to the site's reviewers so they can review the finished product before it goes live.

- Publish the Site (Site administrator only)—After testing is complete, you're ready to go live with your new site. Just [set the site's domain information](#) and then [publish your changes](#) to make your site live!

[Site.com Tab Overview](#)

If you can't see the Site.com tab, go to the Site.com app. It's available in the Lightning Platform app menu in the Salesforce header. Then click the Site.com tab to view the list of your Site.com sites.

[Using Site.com Studio as a Site Administrator or Designer](#)

Site.com Studio provides a dedicated site-building environment for site administrators and designers.

[Using Site.com Studio as a Contributor](#)

Site.com Studio provides a dedicated content-editing environment for contributors.

[Understand the Site Administrator and Designer's Overview Tab](#)

As a site administrator or designer, when you open a site in Site.com Studio, it opens on the Overview tab. Here you can access and manage the site's components and configure the site's properties.

[Understanding the Contributors's Overview Tab](#)

As a contributor, when you open a site in Site.com Studio, it opens on the Overview tab. Here you can access and edit the site's pages and content, and import images and files.

See Also

[Using Site.com Studio as a Site Administrator or Designer](#)

[Using Site.com Studio as a Contributor](#)

[Set Up the Contributor's Studio View](#)

Site.com Tab Overview

If you can't see the Site.com tab, go to the Site.com app. It's available in the Lightning Platform app menu in the Salesforce header. Then click the Site.com tab to view the list of your Site.com sites.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To create or import Site.com sites:

Site.com Publisher User field enabled on the user detail page

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

Site.com Contributor User field enabled on the user detail page

AND

Contributor role assigned at the site level

From this page you can:

- Click **New** to [create](#) or [import](#) a site. Only users with the **Site.com Publisher User** field enabled on their user detail page can create and import sites.
- Filter the sites you see by selecting a predefined list from the dropdown list. My Sites shows the sites you can access and your role. All Sites shows all the sites in your organization even if you don't have access to some of them.
- Click **Edit** next to a site to open it in Site.com Studio.
- Click **Preview** next to a site to see how it looks when rendered in a browser window.
- Click  next to a site to duplicate, export, or delete it. Only users with the **Site.com Publisher User** field enabled on their user detail page and the role of site administrator or designer can duplicate, export, and delete sites. If a site has been published, you can't delete it until you [take it offline](#).
- See the status of your site.
 - In Development—The site has never been published.
 - Published—The site has been published at least one time.
- Click the title of any column to sort your site list. By default, sites are sorted by name.



Note You can't create, delete, or duplicate community sites in Site.com.

See Also

[Using Site.com Studio as a Site Administrator or Designer](#)

[Using Site.com Studio as a Contributor](#)

[Plan and Implement a Site.com Website](#)

Using Site.com Studio as a Site Administrator or Designer

Site.com Studio provides a dedicated site-building environment for site administrators and designers.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage sites Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To manage domains and publish Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator role assigned at the site level

Using the many features available, you can:

- [Create page templates](#) to base your site pages on.
- [Create site pages](#).
- [Import assets](#), such as images and files.
- [Edit the site's style sheet](#) or [create new style sheets](#).
- [View and edit a page or template](#).
- [Add page elements](#) to your site pages to provide features and functionality.
- [Use data services](#) to connect to Salesforce objects to retrieve and display, or to submit data.
- [Create custom widgets](#) that you and other users can reuse throughout the site.
- [Create a multilingual site](#) that lets site visitors choose their preferred language.
- [Create events](#) to add interactive and animated effects to your website.
- [Add IP restrictions](#) to control site visitors' access to the pages, page templates, folders, and assets in your site.
- [Add URL redirects](#) to inform users and search engines if site content has moved.
- [Create folders](#) to organize your site content.
- [Preview your site](#) or generate an anonymous preview link to send to other users.
- [Manage the domain information](#) for your site.
- [Publish your recent changes](#) to the live site.
- [Duplicate, import, and export](#) sites.

 Note

- Designers can't manage domains or publish content.
- You can't create, delete, or duplicate community sites in Site.com.

See Also

[Understand the Site Administrator and Designer's Overview Tab](#)

[Plan and Implement a Site.com Website](#)

[Set Up the Contributor's Studio View](#)

[Site.com Tab Overview](#)

Using Site.com Studio as a Contributor

Site.com Studio provides a dedicated content-editing environment for contributors.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To edit only content in Site.com sites:

Site.com Contributor User field enabled on the user detail page

AND

Contributor role assigned at the site level

With Site.com Studio, you can:

- Open a page to edit it.
- Create site pages, if your site administrator or designer has enabled page creation.
- Edit the page text.
- Add images and hyperlinks to pages.
- Add page elements to pages.
- Import assets, such as images and files.
- Preview the site in a browser window.

See Also

[Understanding the Contributors's Overview Tab](#)

[Plan and Implement a Site.com Website](#)

Site.com Tab Overview

Understand the Site Administrator and Designer's Overview Tab

As a site administrator or designer, when you open a site in Site.com Studio, it opens on the Overview tab. Here you can access and manage the site's components and configure the site's properties.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To manage domains and publish Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator role assigned at the site level

To manage user roles:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator role assigned at the site level

OR

Manage Users



- Select a view (1) on the Overview tab to view its contents (2).

- All Site Content—[Create folders](#) to organize your site content. In this view, you can also create pages, templates, and style sheets, and import assets.
 - Site Pages—[Create site pages](#), [open and edit pages](#), [access page options](#), [create site map links](#), and [organize the site map](#). You can also switch between the default site map view () and the list view ().
 - Page Templates—[Create page templates](#) to base your site pages on, [open and edit existing templates](#), and [access template options](#).
 - Style Sheets—[Edit the site's style sheet](#) or [create new style sheets](#).
 - Assets—[Import and manage assets](#), such as images and files.
 - Widgets—[Build custom widgets](#) that you and your team can reuse throughout the site.
 - Trash Can—Retrieves deleted items. When you delete a page, template, style sheet, or asset, it goes into the trash can. Deleted items remain in the trash can indefinitely. Retrieved items are restored to their original location. If the original location no longer exists, they are restored to the top-level root directory.
 - Change History—View information about recently published files.
 - Site Configuration—[Configure site properties](#), [add IP restrictions](#), [create URL redirects](#), [manage domain information](#), [manage user roles](#), and [add and manage site languages](#).
- Use the toolbar (3) to:
 - [Import assets](#), such as images and files.
 - [Publish recent changes](#).
 - [Preview your site](#) or generate an anonymous preview link to send to other users.
 - [Duplicate](#) or [export the site](#), [overwrite the site](#) with a version from sandbox, or [create a new site](#) ().
 - Use the site's pull-down menu (4) to:
 - Open recently accessed sites.
 - View Site.com Studio as your contributors see it to ensure that you [set up the view](#) correctly.
 - Exit Site.com Studio and return to Salesforce.
 - [Create a new site](#).
 - [Duplicate the site](#).

 **Note** You can't create, delete, or duplicate community sites in Site.com.

See Also

[Using Site.com Studio as a Site Administrator or Designer](#)

[Plan and Implement a Site.com Website](#)

[Site.com](#)

Understanding the Contributors's Overview Tab

As a contributor, when you open a site in Site.com Studio, it opens on the Overview tab. Here you can access and edit the site's pages and content, and import images and files.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To edit only content in Site.com sites:

Site.com Contributor User field enabled on the user detail page

AND

Contributor role assigned at the site level



- Select a view (1) on the Overview tab to view its contents (2).
 - All Site Content—View all of the site's pages, images, and files.
 - Site Pages—[View and edit pages](#) or [create site pages](#), if available.
 - Assets—[Import assets](#), such as images and files.
 - Site Configuration—[Manage user roles](#) in the site. This is available only if you have the “Manage Users” perm.
- Use the toolbar (3) to:
 - [Import assets](#), such as images and files.
 - [Preview the site](#) in a browser window.
- Use the site's pull-down menu (4) to:
 - Open recently accessed sites.
 - Exit Site.com Studio and return to Salesforce.

See Also

- [Using Site.com Studio as a Contributor](#)
- [Site.com](#)

Create a Site.com Community

Each community has one associated Site.com site that lets you add custom, branded pages to your community. By default, Site.com pages are publicly available and don't require login, but you can also create private pages that only community members can access.

-  **Note** As of Spring '15, the Community Template is no longer available for creating communities. If you already have a Site.com community that's based on the Community Template, it will continue to work. For information on creating a community with a new Lightning Community template, see [Create an Experience Cloud Site](#).

Before You Begin

Communities users with the Create and Set Up Communities permission automatically have full site administrator access to a community's Site.com site. To let users who don't have the permission edit the site, you must purchase and assign a Site.com Publisher or a Site.com Contributor feature license. And then you must assign a user role at the site level.

Tips and Considerations

- Users with the Create and Set Up Experiences permission are assigned the role of site administrator in a community's Site.com site. However, they don't appear in the User Roles section on the Overview tab of Site.com Studio.
- You can't create, delete, or duplicate community sites in Site.com.
- When working with data-bound components, such as data repeaters and forms, keep in mind that the objects listed may not be available to site visitors. For authenticated visitors, their user profiles control object access on public and private pages. For unauthenticated visitors, the site's guest user profile controls object access on public pages.
- When adding forms to authenticated community pages in Site.com, set the current user for Salesforce objects that require the Owner ID field. Setting the current user (as opposed to the default guest user) lets you identify the authenticated user when the form is submitted. To set the current user for the Owner ID field, select the field in the form, and click **Configure**. Under Field Properties in the Properties pane, select **Global Property** as the source, and select **Current userID** as the value.
- The home page, 404 page, login page, and self-registration page that you specify for Site.com Community sites in Site Configuration set the default pages for the Site.com Community site. These default URLs are used unless you specify different URLs in Community Management under **Administration Pages** and **Administration Login & Registration**. Community error pages are specified in Lightning Platform Setup, under Error Pages.
- When your Site.com Community site is inactive, users are redirected to the Service Not Available page defined in Community Management under Pages.
- The contributor's view is not available by default for Site.com Community sites. However, you can use a Site.com Contributor license to grant contributor access to a specific user. See *About Feature Licenses* in the Site.com help for details. Alternatively, a user can preview the Site.com Community site as a contributor by appending `?iscontrib` to the site's URL. For example:

`MyDomainName.builder.salesforce-experience.com/?iscontrib`

Use Site.com to Customize Your Community

Communities users can use Site.com to build custom, branded pages for a community. There are many approaches to building custom pages for your community, but these actions are some of the typical stages involved.

Create Branded Pages Overview

When you create a Community site, Salesforce automatically creates a Site.com site and associates it with your community.

Site.com Authorization Overview

As part of your site design, you might want to control what content is public and private to your site

visitors. New sites are initially set so that all site resources, such as folders and pages, are public. You can change the default setting from the Authorization view found under Site Configuration.

[Display Current Community User Information](#)

Site.com designers creating authenticated pages for a community site can display the current user's information by accessing `CurrentUser` namespace expressions.

[Expressions Available for Displaying Current User Information](#)

Use these `CurrentUser` namespace expressions to display authenticated user information on a Site.com community page.

[Determine the URL of a Site.com Page](#)

Determine a Site.com page's URL to let your users access it directly, make it the home page for your community, and more.

[Add Authenticated Site.com Pages to Community Tabs](#)

After you create a private Site.com page, you can add the page to a tab in your community.

[Add Chatter News or Group Feeds to Community Site.com Pages](#)

Use the Chatter News Feed to display a Chatter feed on your site pages, or display the feeds of a particular group using the Chatter Group Feed.

[Improve Performance with HTML Page Caching for Communities in Site.com](#)

HTML caching lets you improve the performance and page rendering of your community's Site.com site by controlling how often the generated markup of the page is reloaded.

See Also

[Use Site.com to Customize Your Community](#)

[Experience Cloud](#)

Use Site.com to Customize Your Community

Communities users can use Site.com to build custom, branded pages for a community. There are many approaches to building custom pages for your community, but these actions are some of the typical stages involved.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage a community's
Site.com custom pages:

Create and Set Up Communities

USER PERMISSIONS NEEDED

OR

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

- Import Assets—Collect the assets, images, and files you plan to use on your custom page. [Import the assets](#) into Site.com Studio, where they appear in the Assets section of the Overview tab.
- Create Branded Pages—The quickest and easiest way to create branded pages is to use the Community Template, which is automatically included with all Site.com community sites. When you [create a new page based on the Community Template](#), the page includes all of the branded styles in your community, including the community's header and footer. If you want even more control over the look and feel of your community page, you can [create your own page template](#), drag community headers and footers to it from the Widgets section of the Page Elements pane, and add other community styles.



Note As of Spring '15, the Community Template is no longer available for creating communities. If you already have a Site.com that's based on the Communities Template, it continues to work. For information on creating a community with a new Lightning Community template, see [Create an Experience Cloud Site](#).

- Use Branded Community Styles—Develop the look and feel of your custom pages by using the [CommunityBranding style sheet](#), or by [creating branded community styles](#) in your own cascading style sheets (CSS). If you're not completely up to speed with CSS, the [Style pane](#) provides an easy, visual way to create and manage styles. Or if you're a CSS expert who likes to get straight into the code, you can hand-code community styles right in your own [style sheets](#).
- Create Public Pages—Using the template as a base, you can quickly [create pages](#), which automatically inherit all the elements of the page template. Or if you need a standalone page that doesn't follow the overall design, you can create a blank page instead.
- Make Pages Private—By default, any page you create in Site.com Studio is publicly available. However, you can [make pages private](#) so that only logged-in Communities users can access them.
- Add Features, Page Elements, and Community Widgets—Use Site.com's prebuilt [page elements](#) to add features such as [navigation menus](#), [images](#), [content blocks](#), and community [widgets](#). Retrieve data from your organization's objects and dynamically display it on your site pages using [data repeaters](#) and [data tables](#). Alternatively, gather and submit data from visitors using [forms](#).
- Add and Edit Content—at this stage, the page is usually ready for you to add and edit content such as [text](#), [images](#), [videos](#), and [hyperlinks](#). And as you work, you can [upload](#) any images or files you need.
- Review and Test the Page—Testing the changes to your page happens throughout the development cycle. Always [preview your changes](#) to ensure they display as expected in a browser. You can also send a preview link to reviewers so they can review the finished product before it goes live.
- Publish the Page—After testing is complete, you're ready to make the page available to your community

by publishing your changes.

- Add Authenticated Pages to Your Community's Tab—Now that the page is tested and published, if you're working with authenticated pages, the final step is to [add the page to a tab](#) in your community.
- Use Site.com in Sandbox—Site.com is now available on sandbox. When you create a sandbox copy from a production organization, you can include your Site.com sites. You can also copy your sandbox site back to production using the [overwrite](#) feature.

See Also

[Create a Site.com Community](#)

[Experience Cloud](#)

[Brand Your Salesforce Tabs + Visualforce Site](#)

Create Branded Pages Overview

When you create a Community site, Salesforce automatically creates a Site.com site and associates it with your community.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

With Site.com Community sites you can:

- Use the [branded Community template](#) to create Site.com pages for your community.
 **Note** As of Spring '15, the Community Template is no longer available for creating communities. If you already have a Site.com that's based on the Communities Template, it continues to work. For information on creating a community with a new Lightning Community template, see [Create an Experience Cloud Site](#).
- Use the [CommunityBranding style sheet](#) to style Site.com pages by using CSS.
- Create your own community CSS styles using a number of [available Network namespace expressions](#).

[Create Branded Pages from the Community Template](#)

Site.com Community sites include a branded template that you can use to create community site pages.

[Apply Community Styles from the CommunityBranding Style Sheet](#)

The `CommunityBranding` style sheet contains a set of CSS styles created from `Network` namespace expressions.

[Create Styles in a CSS Style Sheet](#)

Branded styles are available in Site.com sites through `Network` namespace expressions.

[Expressions Available for Community Branding](#)

You can use the **Network** namespace expressions listed on this page to create your own Community styles.

[View the CommunityBranding Style Sheet](#)

The CommunityBranding style sheet contains a set of branded styles from your community.

Create Branded Pages from the Community Template

Site.com Community sites include a branded template that you can use to create community site pages.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage a community's Site.com custom pages:

Create and Set Up Communities

OR

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

-  **Note** As of Spring '15, the Community Template is no longer available for creating communities. If you already have a Site.com community that's based on the Community Template, it continues to work. For information on creating a community with a new Lightning Community template, see [Create an Experience Cloud Site](#).

The styles for the Community Template come from the CommunityBranding style sheet, which is automatically included for all new Site.com Community sites.

To create branded pages from the Community Template:

1. On the Site.com Overview tab, hover over Site Pages and click **New**.
2. Type the new community page name. Page names can't include spaces or special characters, such as #, ?, or @.
3. Make sure Community Template is selected for the page template.

4. Click **Create**.

Note

- Community branding options, such as headers, footers, and page colors, are set from the **Administration | Branding** section on the Experience Management page.
- Empty community headers and footers, or headers that contain only images, don't work in Site.com. Be sure to specify customized HTML blocks for your community headers and footers if you're creating Site.com pages from the **Community Template**, or creating community headers and footers using **Network** namespace expressions.
- Community headers and footers are available as widgets in Site.com community pages. To add a community header or footer to a blank page, drag it to the page from the Widgets section of the Page Elements pane.

See Also

- [Create Branded Pages Overview](#)
[View the CommunityBranding Style Sheet](#)
[Site.com Page Templates Overview](#)
[Create Site.com Page Templates](#)

Apply Community Styles from the CommunityBranding Style Sheet

The **CommunityBranding** style sheet contains a set of CSS styles created from **Network** namespace expressions.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage a community's Site.com custom pages:

Create and Set Up Communities

OR

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the

USER PERMISSIONS NEEDED

site level

-  **Note** As of Spring '15, the Community Template is no longer available for creating communities. If you already have a Site.com community that's based on the Community Template, it continues to work. For information on creating a community with a new Lightning Community template, see [Create an Experience Cloud Site](#).

The `CommunityBranding` style sheet is attached to the `Community Template`, and is responsible for the template's branded look and feel. You can access the styles in the `CommunityBranding` style sheet and apply them directly to elements on any page.

To apply community styles using the `CommunityBranding` style sheet:

1. Make sure the `CommunityBranding` style sheet is attached to the Site.com page you want to brand.

 **Note** All Site.com pages based on the `Community Template` automatically have the `CommunityBranding` style sheet attached to them.

2. Select the element on the page you want to style.
3. Open the Style pane.
4. Select Class.
5. Start typing "brand".
A list of all of the available styles in the `CommunityBranding` styles sheet appears.
6. Select the style you want to apply.

See Also

- [Create Branded Pages Overview](#)
- [Create Branded Pages from the Community Template](#)
- [View the `CommunityBranding` Style Sheet](#)
- [Create and Use CSS Style Sheets](#)

Create Styles in a CSS Style Sheet

Branded styles are available in Site.com sites through `Network` namespace expressions.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage a community's Site.com custom pages:

Create and Set Up Communities

OR

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

You can access a full list of available **Network** namespace expressions to create community styles in any CSS style sheet. When you add an expression to a CSS rule, Site.com “pulls in” the style as it's defined in the community, and displays it on your page.

To create community styles in a CSS style sheet:

1. Open an existing style sheet or create a style sheet. (See [Creating and Using CSS Style Sheets](#).)
2. Click **Edit Style Sheet Code**.
3. Add a community style rule by using any of the available **Network** expressions. You can create both ID styles and class styles. For example:

```
#main_content_block {  
    background-color: {!Network.primaryColor};  
    color: {!Network.primaryComplementColor};  
}  
.secondary_content_blocks{  
    background-color: {!Network.zeronyColor};  
    color: {!Network.zeronyComplementColor};  
}
```

4. Apply the new styles to elements on other pages.



Note Remember, the style sheet that contains your community styles must be attached to the page containing your styled elements.

See Also

[Create Branded Pages Overview](#)

[Expressions Available for Community Branding](#)

[Create and Use CSS Style Sheets](#)

[Expressions Available for Community Branding](#)

You can use the **Network** namespace expressions listed on this page to create your own Community

styles.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Community branding options, such as headers, footers, and page colors, are set from the **Administration | Branding** section on the Experience Management page.

Note

- Empty community headers and footers, or headers that contain only images, don't work in Site.com. Be sure to specify customized HTML blocks for your community headers and footers if you're creating Site.com pages from the `Community Template`, or creating community headers and footers using **Network** namespace expressions.
- Community headers and footers are available as widgets in Site.com community pages. To add a community header or footer to a blank page, drag it to the page from the Widgets section of the Page Elements pane.

Network Expression	Corresponding Community Branding Page Element
{ !Network.header }	Custom content of the community header.
{ !Network.footer }	Custom content of the community footer.
{ !Network.zeroneyColor }	The background color for the community header.
{ !Network.zeroneyComplementColor }	The font color used with <code>zeroneyColor</code> .
{ !Network.primaryColor }	The color used for active tabs in the community.
{ !Network.primaryComplementColor }	The font color used with <code>primaryColor</code> .
{ !Network.secondaryColor }	The color used for the top border of lists and tables in the community.
{ !Network.tertiaryColor }	The background color for section headers on edit and detail pages in the community.
{ !Network.tertiaryComplementColor }	The font color used with <code>tertiaryColor</code> .
{ !Network.quaternaryColor }	The background color for pages in the community.

Network Expression	Corresponding Community Branding Page Element
{ !Network.quaternaryComplementColor }	The font color used with <code>quaternaryColor</code> .

See Also

[Create Branded Pages Overview](#)

[Create Styles in a CSS Style Sheet](#)

[View the CommunityBranding Style Sheet](#)

The `CommunityBranding` style sheet contains a set of branded styles from your community.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Community branding options, such as headers, footers, and page colors, are set from the **Administration | Branding** section on the Experience Management page.

To see the Community styles in the `CommunityBranding` style sheet, on the Site.com Overview tab, click Style Sheets, and click the `CommunityBranding` style sheet. The Community styles are listed on the left. To see the code for the style sheet, click **Edit Style Sheet Code**.

A total of fourteen Community class styles are provided. The default contents of the style sheet:

```
.brandZeronaryBgr {
    background-color: { !Network.zeronaryColor } !important;
}
.brandZeronaryFgr {
    color: { !Network.zeronaryComplementColor } !important;
}
.brandPrimaryBgr {
    background-color: { !Network.primaryColor } !important;
}
.brandPrimaryFgr {
    color: { !Network.primaryComplementColor } !important;
}
.brandPrimaryBrd2 {
    border-color: { !Network.primaryComplementColor } !important;
}
```

```
.brandPrimaryFgrBrdTop {  
    border-top-color: {!Network.primaryComplementColor} !important;  
}  
.brandPrimaryBrd {  
    border-top-color: {!Network.primaryColor} !important;  
}  
.brandSecondaryBrd {  
    border-color: {!Network.secondaryColor} !important;  
}  
.brandSecondaryBgr {  
    background-color: {!Network.secondaryColor} !important;  
}  
.brandTertiaryFgr {  
    color: {!Network.tertiaryComplementColor} !important;  
}  
.brandTertiaryBgr {  
    background-color: {!Network.tertiaryColor} !important;  
    color: {!Network.tertiaryComplementColor} !important;  
    background-image: none !important;  
}  
.brandTertiaryBrd {  
    border-top-color: {!Network.tertiaryColor} !important;  
}  
.brandQuaternaryFgr {  
    color: {!Network.quaternaryComplementColor} !important;  
}  
.brandQuaternaryBgr {  
    background-color: {!Network.quaternaryColor} !important;  
}
```

See Also

- [Create Branded Pages Overview](#)
- [Create Branded Pages from the Community Template](#)

Site.com Authorization Overview

As part of your site design, you might want to control what content is public and private to your site visitors. New sites are initially set so that all site resources, such as folders and pages, are public. You can change the default setting from the Authorization view found under Site Configuration.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

The global site authorization options are:

- No Authorization (default)—All resources are public.
- Requires Authorization—All resources are private.
- Custom—All resources are public by default, but can be made private.

The No Authorization and Requires Authorization options let you quickly make your site either all public or all private. But, if you want to control access to individual pages, folders, and other resources, use the Custom option. Selecting Custom enables a **Requires Authorization** checkbox on the Actions menu  for all resources throughout the site. You can define authorization at the site, folder, page, and individual resource level. As you mark items for authorization, a lock icon  appears on them. After a resource, like a page, is marked as private, users who aren't logged into Salesforce are asked to log in when they try to access it.

Resources can inherit their privacy setting from folders. For example, when a resource, such as a site folder, is marked for authorization, anything placed in that folder inherits the folder's authorization setting and becomes private. If you drag that resource into a public folder, it becomes public again. But, if you explicitly mark a resource as private using the Actions menu, and then drag it into a public folder, it still remains private because the privacy setting at the resource level dominates.

When you use the Custom option, an authorization table appears in the Authorization view that lets you manage your private resources/items marked as private. You can remove authorization from a resource by either deleting it from the authorization table, or by deselecting the Requires Authorization box on the item itself.

[Setting Custom Authorization](#)

When you select Custom authorization, you get a great deal of flexibility in controlling access to your site. Not only can you control who has access to top level resources, like folders and pages, but you can also set access at the individual resource level.

[Remove Site.com Authorization](#)

You can remove authorization for a resource by either deleting it from the authorization table under Site Configuration, or by deselecting **Requires Authorization** from the Actions menu.

See Also

[Setting Custom Authorization](#)

[Remove Site.com Authorization](#)

Setting Custom Authorization

When you select Custom authorization, you get a great deal of flexibility in controlling access to your site. Not only can you control who has access to top level resources, like folders and pages, but you can also set access at the individual resource level.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS NEEDED

To manage authorization:

You must be an administrative user on the site

Using Custom authorization at the folder level is a great way to make a large number of resources private without having to mark them individually. Let's say you periodically run sale offers for your paid users. If you drag all the sale pages into a special folder you mark for authorization, they instantly inherit the folder's setting. Users will need to log in to access them. Plus, if you decide to make one of the sale pages available to everyone, you can simply drag it back into a public folder, or to the root of the All Site Content area.

1. Open your site for editing.
2. Click **Site Configuration | Authorization**.
3. Select **Custom**.
4. Click **All Site Content**.
5. Create a folder to hold the private pages if it doesn't already exist.
6. From the folder's Actions  menu, select **Requires Authorization**. You'll see the lock  appear on the folder. It is now private.
7. Drag any pages you want to make private into the folder. A lock  appears on them too.



Example Let's take another example. If you have a page that you'd like to keep private no matter where it resides, you can set its authorization using the Actions menu. After you set it at the individual resource level, it remains private even if you drag it into a folder that isn't set to private. In other words, a resource marked private is always private until you deselect **Requires Authorization** on the Actions menu.

If you check the Authorization page, you'll see all folders and resources marked private are listed in the authorization table where you can view and delete them.

See Also

[Remove Site.com Authorization](#)

[Remove Site.com Authorization](#)

You can remove authorization for a resource by either deleting it from the authorization table under Site Configuration, or by deselecting **Requires Authorization** from the Actions menu.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To manage authorization: You must be an administrative user on the site

1. Open your site for editing.
2. Click **Site Configuration | Authorization**.
3. From the authorization table, click **Delete** next to the item you want to remove. Alternatively, navigate to the All Site Content view. Select the resource. From the Actions  menu, deselect **Requires Authorization**.

 **Example** If a resource is explicitly marked as private using the Actions menu, then you must remove authorization from it using the Actions menu. For example, if a page marked private is dragged into a folder that's public, it remains private. Likewise, if you drag it into a folder that's already private, and remove the authorization on that folder, the page is still private.

See Also

[Setting Custom Authorization](#)

Display Current Community User Information

Site.com designers creating authenticated pages for a community site can display the current user's information by accessing `CurrentUser` namespace expressions.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage a community's Site.com custom pages:

Create and Set Up Communities

OR

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

1. Open the page on which you want to display the current community user's information.
2. From the Page Elements pane, drag a **Content Block** or **Custom Code** page element onto the page.
3. Type `{!CurrentUser.}` and the value that you want to display.
For example, `{!CurrentUser.firstName}`.

Check the list of [available expressions for displaying current user information](#).

4. Add any additional text you require.
For example, `Welcome back {!CurrentUser.firstName}!`.
5. If you're in a Content Block, click **Save**. If you're in a Custom Code element, click **Save and Close**.

 **Note** If an unauthenticated user views a page that contains `CurrentUser` expressions, the current user information does not appear. For example, if an unauthenticated user viewed a page that contained the previous example, the user sees “Welcome back!” as the welcome message.

Expressions Available for Displaying Current User Information

Use these `CurrentUser` namespace expressions to display authenticated user information on a Site.com community page.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

CurrentUser Expression	Displays
<code>{!CurrentUser.name}</code>	Combined first and last name of the user, as displayed on the user detail page.
<code>{!CurrentUser.firstName}</code>	First name of the user, as displayed on the user edit page.
<code>{!CurrentUser.lastName}</code>	Last name of the user, as displayed on the user edit page.
<code>{!CurrentUser.userName}</code>	Administrative field that defines the user's login.
<code>{!CurrentUser.id}</code>	User's Salesforce ID.
<code>{!CurrentUser.email}</code>	Email address of the user.
<code>{!CurrentUser.communityNickname}</code>	Name used to identify the user in a site.

CurrentUser Expression	Displays
{ !CurrentUser.accountId }	Account ID associated with the user. It displays a valid account id for partner and customer users. For all others, it displays '0000000000000000'.
{ !CurrentUser.effectiveAccountId }	Account ID associated with the effective account. This expression displays a valid account ID for partner and customer users. For all others, it displays '0000000000000000'.

Determine the URL of a Site.com Page

Determine a Site.com page's URL to let your users access it directly, make it the home page for your community, and more.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage a community's Site.com custom pages:

Create and Set Up Communities

OR

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

After you create a Site.com page, you can determine the page's URL to:

- Provide your users with a URL that lets them access a public page directly.
- Create a link to the page from other pages, including Salesforce Sites and Visualforce pages.
- Make it the home page for your community using a URL redirect in Salesforce Sites.
- Add a private page to a web tab in your community.

1. To determine the correct URL for the page:

- From the Create Community wizard, click **Customize**.
 - If you navigated away from the Create Community wizard, click **Customize | Communities | All Communities**, then click the **Manage** button next to the community name.
2. Click **Administration Settings**.
3. Copy the URL displayed on the page and paste it into a text editor.
4. To create a URL that points to:
- The Site.com site's home page, append /s/ to the URL. For example,
`https://MyDomainName.my.site.com/ExperienceCloudSiteName/s/`.
 - A specific Site.com page, append /s/<page_name>, where <page_name> is the name of the Site.com page. For example,
`https://MyDomainName.my.site.com/ExperienceCloudSiteName/s/promotion`.

The URL is case-sensitive and “s” must be lowercase.

See Also

[Add Authenticated Site.com Pages to Community Tabs](#)

[Salesforce Sites URL Redirects](#)

Add Authenticated Site.com Pages to Community Tabs

After you create a private Site.com page, you can add the page to a tab in your community.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage a community's Site.com custom pages:

Create and Set Up Communities

OR

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

In this case, you need to create a web tab that points to your Site.com page.

1. In the Properties pane for your page, select **Show Salesforce Header**.
Selecting this option ensures that you see tabs in your community.
2. Enter the tab name as it must appear on the tab in your community.
The web tab you create must have the same name.
3. [Determine the correct URL for the page](#).
The URL must be in the following format
`https://MyDomainName.my.site.com/mycommunity/s/<pagename>`, where *pagename* matches the name of your page.
4. From Setup, enter *Tabs* in the **Quick Find** box, then select **Tabs**.
5. In Web Tabs, click **New** and enter the name of the tab as it appears in the Tab Name field in your page properties.
6. On the Step 3 screen, paste the URL you created in the Button or Link URL text box.
7. Return to the Create Community wizard and add the new tab to your community.

To preview the private page in your community, you must publish your Site.com site.

 **Note** You can't publish your site from the sandbox.

See Also

- [Experience Cloud](#)
- [Create Web Tabs](#)

Add Chatter News or Group Feeds to Community Site.com Pages

Use the Chatter News Feed to display a Chatter feed on your site pages, or display the feeds of a particular group using the Chatter Group Feed.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage a community's Site.com custom pages:

Create and Set Up Communities

OR

Site.com Publisher User field enabled on the user detail page

AND

USER PERMISSIONS NEEDED

Site administrator or designer role assigned at the site level

1. Drag the **News Feed** or **Group Feed** from the Widgets section of the Page Elements pane onto the page.

When you add a widget to a page, it creates a copy or *instance* of the widget. You can't edit the content of a widget, but you can edit the properties.

2. If you're adding a group feed, enter the Group ID in the Properties pane.

The Group ID determines which group feed is displayed on your page. If you want to show the feeds for multiple groups, you can include more than one group feed on a page.

3. Preview the page to test the feed, or use Live Mode to see how the feed renders in different mobile devices.

Consider the following limitations when using a news or group feed in your community Site.com sites:

- Chatter news and group feeds only appear if a user is logged in to the community. They don't appear to guest users or in anonymous preview mode.
- Chatter news and group feeds don't render appropriately on pages less than 700 px wide. We recommend a minimum page width of 700 px to view full content. We also recommend using a white background.
- Chatter news and group feeds only inherit some page branding elements.

Improve Performance with HTML Page Caching for Communities in Site.com

HTML caching lets you improve the performance and page rendering of your community's Site.com site by controlling how often the generated markup of the page is reloaded.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage a community's Site.com custom pages:

Create and Set Up Communities

OR

Site.com Publisher User field enabled on the user

USER PERMISSIONS NEEDED

detail page

AND

Site administrator or designer role assigned at the site level

Lets say 100 people visit the page at the same time. Without caching, the page makes 100 separate requests for the same markup, slowing performance considerably. However, with caching enabled, the page markup is requested and retrieved only one time—the first time someone visits the page. Any subsequent page requests during a set time period are returned from the cache. When the specified time period expires, the cache is refreshed.

 **Note** The caching duration applies only to community pages that are accessed by guest users. When a user logs in to access the page, caching is disabled.

1. In Site.com Studio, open the page.
2. In the **Cache Duration (Minutes)** field of the Cache section of the Properties tab, specify the length of time to cache the page.
By default, the caching duration of a page is set to 30 minutes.

To disable caching, set the page's caching duration to 0.

Create a Site.com Site

To get started with Site.com, create a new blank site.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To create or import Site.com sites:

Site.com Publisher User field enabled on the user detail page

1. On the Site.com tab in the Site.com app, click **New**. Alternatively, in Site.com Studio, click **Create a New Site** in the site's dropdown menu.
2. Click **Create a Blank Website**.

3. Enter the site name.
4. Click **Create**. Your new website opens in Site.com Studio, where you can create page templates and site pages, and add functionality to it.

 **Note** You can't create, delete, or duplicate community sites in Site.com.

[Delete a Site.com Site](#)

You can delete any site that isn't published. If the site is published, you must first unpublish it before you can delete it.

[Duplicate a Site.com Site](#)

Create a copy of a site.

[Export a Site.com Site](#)

You can export your site from Site.com to your hard drive. The site is exported in a packaged format with a .site extension, which you can import into another Salesforce organization. The maximum site size you can import is 2 GB.

[Import a Site.com Site](#)

You can import an exported Site.com site into your organization. When you import a site, you're given the site administrator role in the site. You can import a site file that was exported in Summer '19 or later and has a maximum site size of 2 GB.

[Configure Site Properties](#)

Set properties for the site, such as the home page, site name, and error page, and create an anonymous preview URL that allows other users to review the site before it goes live. The URL is always valid (unless you disable it) and shows the latest work in progress. It's only available to the people you send it to, and can't be found by search engines.

[Enable Clickjack Protection in Site.com](#)

Clickjacking is a type of attack that tricks users to click something, such as a button or link, because they perceive they are clicking something safe. Instead, the button or link performs malicious actions on your site, leading to data intrusion, unauthorized emails, changed credentials, or other site-specific results. Hidden iframes can be placed maliciously on site pages and entice users to click a button or link that appears below the hidden iframe. With clickjack protection, you can configure whether your browser allows external domains to frame your Site.com site pages.

[Site.com Version Overview](#)

Each time you publish your site, it's tracked as a version. You can restore your site back to one of the previously published versions. You can't select individual components when restoring; you must restore the complete site.

[Create URL Redirects in Site.com](#)

If you move or reorganize pages in your site, search engines can have trouble finding the new page locations. To avoid this issue, set up URL redirects to inform users and search engines that site content has moved.

[Import External Websites into Site.com](#)

With Site.com Studio, you can import your existing website and recreate it automatically as a Site.com site, which prevents you from having to recode existing HTML pages.

[Copy and Overwrite a Site](#)

Export a copy of your site, and then use the overwrite feature to replace your current production Site.com or Site.com community site with the exported file.

[Use the Metadata API to Deploy a Site](#)

As a user, you can migrate a site from sandbox to production. You can use the Metadata API to create a deployable package for Site.com sites and Site.com Communities sites.

See Also

[Create Site.com Page Templates](#)

[Create Site.com Pages](#)

[Edit Site.com Pages as a Designer or Site Administrator](#)

Delete a Site.com Site

You can delete any site that isn't published. If the site is published, you must first unpublish it before you can delete it.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

See [Taking a Site Offline](#).

1. On the Site.com tab in the Site.com app, select the site and click  | Delete.
2. Click **OK**.

 **Note** You can't create, delete, or duplicate community sites in Site.com.

See Also

[Export a Site.com Site](#)

Duplicate a Site.com Site

Create a copy of a site.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To create a copy of a site:

1. On the Site.com tab in the Site.com app, select the site and click | **Duplicate**. Alternatively, on the Overview tab in Site.com Studio, click | **Duplicate This Site**.
2. Enter a name for the site.
3. Click **Create**.

Note

- If you're creating a copy of a site that uses [data services](#), you must [set the data access permissions](#) in the new site's guest user profile.
- You can't create, delete, or duplicate community sites in Site.com.

See Also

[Create a Site.com Site](#)

[Export a Site.com Site](#)

[Import a Site.com Site](#)

Export a Site.com Site

You can export your site from Site.com to your hard drive. The site is exported in a packaged format with a .site extension, which you can import into another Salesforce organization. The maximum site size you can import is 2 GB.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

1. On the Site.com tab in the Site.com application, select the site and click  | Export. Alternatively, on the Overview tab in Site.com Studio, click  | Export This Site.
2. If the site is:
 - Smaller than 100 MB, select a location to save the exported .site file on your hard drive and click **Save**.
 - Larger than 100 MB, you receive an email when the export process has completed. Click the link in the email to download the exported .site file.

Note

- Exporting a site doesn't remove it from the current organization.

See Also

[Create a Site.com Site](#)

[Import a Site.com Site](#)

Import a Site.com Site

You can import an exported Site.com site into your organization. When you import a site, you're given the site administrator role in the site. You can import a site file that was exported in Summer '19 or later and has a maximum site size of 2 GB.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To create or import Site.com sites:

Site.com Publisher User field enabled on the user detail page

1. On the Site.com tab in the Site.com app, click **New**. Alternatively, in Site.com Studio, click **Create a New Site** in the site's dropdown menu.
2. Select **Import a Site or Template**.
3. Enter the site name.
4. To locate the exported site on your hard drive, click **Browse**. Exported sites have a .site extension.
5. Click **Create**.

Note

- If you're importing a site that uses **data services**, you must **set the data access permissions** in the imported site's guest user profile. Additionally, data repeaters, data tables, data functions, or forms can require reconfiguration.
- You can't create, delete, or duplicate community sites in Site.com.

See Also

[Create a Site.com Site](#)

[Export a Site.com Site](#)

Configure Site Properties

Set properties for the site, such as the home page, site name, and error page, and create an anonymous preview URL that allows other users to review the site before it goes live. The URL is always valid (unless you disable it) and shows the latest work in progress. It's only available to the people you send it to, and can't be found by search engines.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

1. On the Overview tab, click **Site Configuration**.
2. Click **Edit**.
3. In the Site Configuration view, you can:
 - Replace the name in the **Site Name** field to rename the site.
 - See the **Developer Name** of the site. This read-only field can differ from the **Site Name**. The **Developer Name** is used by the Metadata API.
 - Select **Enable Anonymous Preview** to create a URL that allows other users to preview the site before it goes live. (Click the **View Anonymous Preview** option that appears in the **Preview** menu to access the preview URL, which you can copy and send to other users to review and test your changes.) **Enable Anonymous Preview** is also available from the **Preview menu** on the Overview tab.
 - [Access the site's guest user profile](#).
 - [Set the clickjack protection level](#).
 - Determine whether guest users can view features available only in Lightning. If you disable **Lightning Features for Guest Users**, Lightning features don't load.
 - Enable Content Sniffing Protection to force the browser to use the Content-Type header only.
 - Enable **Browser Cross Site Scripting Protection** to protect against reflected cross-site scripting attacks.
 - Select **Referrer URL Protection** to have the referrer header shows only Salesforce.com rather than the entire URL when loading pages.
 - Select the home page for your website in the Home Page dropdown list.
 - *Site.com Community sites only:*
 - Select the login page for your Site.com Community site in the Login Page dropdown list.
 - Select the page that you've set up for Site.com Community site users who don't have accounts yet from the Registration Page dropdown list.
 - Select the Forgot Password page you've set up for your community using Site.com.
 - Select a user-friendly error page in the 404 Page dropdown list to display when a page can't be found. It's a good idea to create a user-friendly error page to assist site visitors if they encounter a broken link.
4. Click **Save**.

 **Note**

- The home page, 404 page, login page, and self-registration page that you specify for Site.com Community sites in Site Configuration set the default pages for the Site.com Community site.

These default URLs are used unless you specify different URLs in Community Management under **Administration Pages** and **Administration Login & Registration**. Community error pages are specified in Lightning Platform Setup, under Error Pages.

- When your Site.com Community site is inactive, users are redirected to the Service Not Available page defined in Community Management under Pages.

See Also

- [Create a Site.com Site](#)
- [Using Site.com Studio as a Site Administrator or Designer](#)
- [Enable Clickjack Protection in Site.com](#)

Enable Clickjack Protection in Site.com

Clickjacking is a type of attack that tricks users to click something, such as a button or link, because they perceive they are clicking something safe. Instead, the button or link performs malicious actions on your site, leading to data intrusion, unauthorized emails, changed credentials, or other site-specific results. Hidden iframes can be placed maliciously on site pages and entice users to click a button or link that appears below the hidden iframe. With clickjack protection, you can configure whether your browser allows external domains to frame your Site.com site pages.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level



Note To configure clickjack protection for Experience Cloud sites, see [Enable Clickjack Protection in Experience Cloud Sites](#).

1. In Site.com Studio, click **Site Configuration | Edit**.
2. Select your preferred level of clickjack protection.
 - **Allow framing by any page (no protection)**: The least secure level. All external domains can frame

your site pages.

- **Allow framing of site pages on external domains (good protection):** Only trusted external domains can frame your site pages. You specify the domains that you trust in the Trusted Domains for Inline Frames list.
- **Allow framing by the same origin only (recommended):** The default level for Site.com sites. Allows framing of site pages by pages with the same domain name and protocol security.
- **Don't allow framing by any page (most protection):** The most secure level, but this option can cause certain pages to appear as blank pages. To avoid this issue, use the default setting instead.

3. Save your changes.
4. If you chose to allow framing of your site pages on external domains, specify the domains that you trust to frame each site's pages.
 - a. From Setup in Salesforce Classic, enter *Sites* in the Quick Find box, and then select **Sites**.
 - b. Click the site label to open the Site Details page.
 - c. Click **Add Trusted Domain** in the Trusted Domains for Inline Frames section and enter the domain you want to allow iframes on.

You can add up to 512 domains.

 **Tip** Added domains take effect only when **Allow framing of site pages on external domains (good protection)** is selected.

 **Note** Internet Explorer supports clickjack protection through the legacy X-Frame-Options HTTP Header only. This header supports `sameorigin`, `deny` (none), `allowall`, and `allow-from uri`. In particular, `allow-from uri` supports only one URI.

To support a list for IE users, the framing site must identify itself to the site domain by passing in a query parameter in the `iframe` tag. For example, if you add `https://www.example.com` as a trusted external domain and your site URL is `https://MyDomainName.my.site.com`, then the page on `https://www.example.com` must make its iframe as follows:

```
<iframe src="https://MyDomainName.my.site.com?_iframeDomain=https://www.example.com"></iframe>
```

You can also set the trusted external domain in the `_iframeDomain` cookie. This method allows iframes if the `_iframeDomain` URL variable isn't saved when navigating between pages in IE.

```
Cookie iframeDomainCookie = ApexPages.currentPage().getCookies().get('iframeDomain');

if (iframeDomainCookie == null) {
    iframeDomainCookie = new Cookie('iframeDomain', 'www.example.com');

    // Set the new cookie for the page
    ApexPages.currentPage().setCookies(new Cookie[] {iframeDomainCookie});
}
```

See Also

- [Configure Site Properties](#)
- [Enable Clickjack Protection in Experience Cloud Sites](#)
- [Create and Edit Salesforce Sites](#)

Site.com Version Overview

Each time you publish your site, it's tracked as a version. You can restore your site back to one of the previously published versions. You can't select individual components when restoring; you must restore the complete site.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

When working in Site.com Studio, you're always working on an unpublished version of your site. It's your *working copy*. When you restore a version, you overwrite your working copy, not your live site. You must publish the restored version before you see the change on your live site.

In Site.com Studio, you can find your site versions in the Change History view on the Overview tab. Not all items in a site are reverted when you restore a version. Some things, like user roles, remain unchanged even when you restore from a previous version. Everything in a site is under version control except:

- Site name
- Anonymous preview setting
- Guest User Profile settings
- Clickjack protection level
- Domains and path prefixes
- User role settings

[Restore to a Previous Site Version](#)

The Change History list in Site.com Studio tracks all published versions of your site. You can select any previously published version and restore it. You can only restore an entire site, not parts of a site.

See Also

- [Restore to a Previous Site Version](#)

Restore to a Previous Site Version

The Change History list in Site.com Studio tracks all published versions of your site. You can select any previously published version and restore it. You can only restore an entire site, not parts of a site.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

 **Note** The restore version feature is not available in Communities.

When working in Site.com Studio, you're always working on an unpublished version of your site. It's your *working copy*. When you restore a version, you overwrite your working copy, not your live site. You must publish the restored version before you see the change on your live site.

 **Warning** You can't restore the working copy of your site after you revert to a previous version. Therefore, it's a good idea to back up the working copy of the site before reverting to ensure you don't lose any unpublished changes.

1. Select the Overview tab.
2. From the Change History view, select the version you want to restore.
3. Click  | **Restore Version**.
4. Click **OK** at the confirmation message.

After you restored your working site to a previous version, you can continue to make additional changes until you're ready to publish the site.

Create URL Redirects in Site.com

If you move or reorganize pages in your site, search engines can have trouble finding the new page locations. To avoid this issue, set up URL redirects to inform users and search engines that site content has moved.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

You can use URL redirects to:

- Maintain search engine ranking. For example, if you change a page's name from "Gadgets" to "Widgets," creating a redirect rule from /Gadgets to /Widgets lets you restructure the site without affecting your page ranking.
- Make URLs more readable and memorable. For example, site visitors find long or numeric page names, such as /widget65AD890004ab9923, difficult to remember. Instead, you can provide them with a short, friendly URL, such as /widget, and create an alias that redirects to the correct page when the user uses the short URL alias.
- Assist with migration from another system to Site.com if you're still using the same domain. For example, if your old site ran on PHP, you can create a redirection rule from an old page, such as /index.php, to a new page in Site.com, such as /myNewPage.

To assign a redirect to a site page:

1. On the Overview tab, click **Site Configuration | URL Redirects**.
2. Click **Create a Redirect**.
3. Specify the **Redirect type**:

Option	Description
Permanent (301)	Select this option if you want users and search engines to update the URL in their systems when visiting the page. Users visiting a page redirected with this type are sent seamlessly to the new page. Using a permanent redirect ensures that your URLs retain their search engine popularity ratings, and that search engines index the new

Option	Description
	page and remove the obsolete source URL from their indexes.
Temporary (302)	Select this option if you want users and search engines to keep using the original URL for the page. Search engines interpret a 302 redirect as one that could change again at any time, and though they index and serve up the content on the new target page, they also keep the source URL in their indexes.
Alias	<p>Select this option if you don't want the URL to change in the user's browser, but you want to redirect to a different page. Search engines aren't aware of the change or update their records.</p> <p>Alias redirects only work when you redirect from one Site.com page to another. You can't create an alias to an external address.</p>

4. Specify the former page location in the **Redirect from** field.
 - The page location must be a relative URL.
 - The page can have any valid extension type, such as `.html` or `.php`, and can contain parameters. Parameter order is irrelevant.
 - The URL can't contain anchors, such as `/siteprefix/page.html#target`.
 - You can create just one redirection rule *from* a particular URL. If you create a rule with the same Redirect From information, the old rule is overwritten.
5. Specify the new page location in the **Redirect to** field, which can be a relative URL or a fully qualified URL with an `http://` or `https://` prefix. Unlike pages you're redirecting from, pages you're redirecting to can contain anchors.
6. To immediately enable the redirection rule, ensure **Active** is selected. To enable it at a later stage, deselect the property.
7. Click **Save**.

The URL Redirects section displays all URL redirection rules you've created for your site.

- Edit an assigned redirect rule by clicking  | **Edit Redirect**.
- Delete a redirect rule by clicking  | **Delete Redirect**.

Import External Websites into Site.com

With Site.com Studio, you can import your existing website and recreate it automatically as a Site.com site, which prevents you from having to recode existing HTML pages.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

Create a zipped file of your website and the content with the desired folder structure. Create a Site.com site and then [import](#) your zipped file.

During the creation of the new Site.com site, all your HTML pages and assets are copied into the new site in the same locations they are in the zipped file. Here are some guidelines for importing assets.

- Some code in style sheets doesn't convert properly to the Site.com format. If the code doesn't convert properly, you receive a warning message. You can continue to import the zipped file or stop the import. If you continue, the style sheet imports and you can then manually attach it to your pages.
- You can manually edit the HTML of the `<head>` section of a page using Edit Head Markup in the scripts section on the properties pane.
- The maximum file size you can import is 50 MB unless you import and *unzip* a .zip file. In that case, you can import a .zip file of up to 200 MB if you select **Unzip files** during the import process. If your site is larger than 200 MB when zipped, you can create more than one zipped file and import them individually.
- Site.com attempts to format links correctly when importing a page. Links are checked in content blocks, custom code, and head script markup. You can check for any links that are broken on your page by using the link checker. Open the page, select  | **Find Broken Links**. A dialog displays showing any broken links. To fix the link, click **Edit**. The link opens in the HTML editor.

See Also

[Import a Site.com Site](#)

Copy and Overwrite a Site

Export a copy of your site, and then use the overwrite feature to replace your current production Site.com or Site.com community site with the exported file.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To overwrite sites:

Site administrator or designer role assigned at the site level

First, export a copy of your site from your Site.com Studio. Next, import the exported .site file into production using the overwrite feature. Overwrite replaces your production site with the imported site. You can only overwrite sites that are the same type. For example, you can't overwrite a Site.com site with a Site.com community site.

 **Note** We recommend only overwriting Aura sites, and only using Aura .site files to do so. To replace a Build Your Own (LWR) site with an Aura .site file, or to replace any site using a Build Your Own (LWR) .site file, import the .site file to a new site instead.

1. Create your .site file using the [export](#) feature in Site.com Studio.
2. From Site.com Studio in your production site, click **Site Actions | Overwrite This Site**. Alternatively, if you have access to the Site.com tab, you can click  next to the site name to find **Overwrite**.
3. Click **Browse** to find and select the .site file you exported.
4. Click **OK** at the overwrite warning.

 **Warning** You can't revert after you overwrite a site.

Here are a few tips when using overwrite:

- You must publish the site you overwrote for changes to take effect.
- When copying a site, the production site is overwritten. Be sure to back up your production site by exporting a .site file before you overwrite.
- Overwrite doesn't copy data changes. For example, if you created a custom object to use in a repeater in the site that you're copying from, the repeater doesn't work in the production site until you create the same custom object.
- If the production site has assets that don't exist in the site that you're copying from, they're moved to the trash can during the overwrite process so you can restore them if needed.
- If your copied site has a different name than your production site, the production name is preserved and only the contents are changed. For example, if your site is called Site One and you overwrite your production site called Site Two, the production site is still called Site Two.

See Also

[Export a Site.com Site](#)

[Use the Metadata API to Deploy a Site](#)

Use the Metadata API to Deploy a Site

As a user, you can migrate a site from sandbox to production. You can use the Metadata API to create a deployable package for Site.com sites and Site.com Communities sites.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

-  **Note** For information on deploying a Lightning community or a Salesforce Tabs + Visualforce community, see [Deploy a Community from Sandbox to Production](#).

There are several tools that you can use to create a package:

- Change sets (for Site.com and Site.com Communities sites only). The component type is called **Site.com**.
- Lightning Platform Migration Tool for Ant, which works for creating all site types. The metadata type is called **SiteDotCom**.

If using change sets, select Site.com from the list and follow the prompts to create your package. If using Lightning Platform as your tool, you must create a package.xml file. That file is submitted to the Metadata API to create a package.

-  **Note** You can include a Guest User Profile in your package.xml file. If you do so, that Guest User Profile is linked to the site during deployment.

The packaging process generates a folder that contains a content file and a metadata xml file. The content file name is [sitename].site. The metadata .xml file name is [sitename].site-meta.xml.

If you deploy a package that doesn't include a .site file, an empty site is created. If the package contains a site file, and the organization already contains a site with the same name, the site is updated.

-  **Note** There is a file size limitation when using the Metadata API to deploy a site from sandbox to production. The assets in the .site file can't be larger than 40 MB. The site gets created, but the assets show in the new site as broken. To fix the assets, export the assets from the sandbox environment separately and then import them into your new site.

For help with the Metadata API, see the Metadata API Developer Guide. You can find help for change sets in Salesforce Help and for the Migration Tool Guide at [https://developer.salesforce.com/page/](https://developer.salesforce.com/page/Migration_Tool_Guide)
[Migration_Tool_Guide](#).

[Sample Package.xml Files](#)

Here are some sample Site.com package.xml files.

See Also

[Change Sets](#)

[Sample Package.xml Files](#)

Sample Package.xml Files

Here are some sample Site.com package.xml files.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

Here is a sample package.xml file for a Site.com site.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>xyzsite</members>
        <name>SiteDotCom</name>
    </types>
    <version>30.0</version>
</Package>
```

Here is an example of a package.xml for a Salesforce Communities site. The package also includes a Guest User Profile.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>xyzsite</members>
        <name>CustomSite</name>
    </types>
    <types>
        <members>xyzsite</members>
        <name>Network</name>
    </types>
    <types>
        <members>xyzsite1</members>
        <name>SiteDotCom</name>
    </types>
    <types>
        <members>xyzsite Profile</members>
        <name>Profile</name>
    </types>
</Package>
```

```
</types>
<version>30.0</version>
</Package>
```

Import and Manage Assets

Contributors, publishers, and site administrators can import a variety of assets, such as images, HTML pages, and PDFs, to use in a website. You can import assets and files individually, or use a zipped file. When importing entire websites or large numbers of assets, it's easier to create a zipped file of the content with the desired folder structure. When importing the zipped file for a website, Site.com recreates your website and places everything in the same folder structure.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

Site.com Contributor User

AND

Contributor role assigned at the site level

 **Note** The maximum file size you can import is 50 MB unless you import and *unzip* a .zip file. In that case, you can import a .zip file of up to 200 MB if you select **Unzip files** during the import process.

The quickest way to import one or more files is to:

1. Select the files from your computer and drag them directly onto the Studio interface. This is supported for Mozilla® Firefox® and Google® Chrome only. You can drag individual files, or a zipped file.
2. Depending on the types of files you're importing, a dialog box can appear that lets you:

- Select **Unzip files** to extract the contents of a .zip file. If the .zip file includes folders, this structure is maintained in your site.
 - Select **Overwrite existing files** to replace a file that exists in the site.
 - Select **Convert CSS files into style sheets**, if you're a site administrator or designer, to convert a CSS file into a style sheet that you can use to style your website.
-  **Note** If you import a .zip file that includes CSS files, and they fail to convert, they aren't valid. Try unchecking this option and then reimporting the .zip file.
- Select **Convert HTML files into pages** to import HTML pages into your website. The structure of the HTML page is maintained in your site, but the HTML is not validated during import.

Alternatively, to import a single file:

1. Click **Import....**
 2. Click **Browse...** to locate the file.
 3. Select the file and click **Open**.
 4. Depending on the type of file you're importing, you can:
 - Select **Unzip files** to extract the contents of a .zip file. If the .zip file includes folders, this structure is maintained in your site.
 - Select **Overwrite existing files** to replace a file that exists in the site.
 - Select **Convert CSS files into style sheets**, if you're a site administrator or designer, to convert a CSS file into a style sheet that you can use to style your website.
-  **Note** If you import a .zip file that includes CSS files, and they fail to convert, they aren't valid. Try unchecking this option and then reimporting the .zip file.
5. Select **Convert HTML files into pages** to import HTML pages into your website. The structure of the HTML page is maintained in your site, but the HTML is not validated during import.
 6. Click **Import**. A message appears indicating whether the file was imported successfully.
 7. Click .

You can [add images](#) and [videos](#) to the text areas of your site pages, or [create a hyperlink](#) to any imported asset. If you're a site administrator or designer, you can add also [add images directly](#) to the page.

View the list of imported assets in the Assets view on the Overview tab. You can also access assets in the All Site Content view, which displays the folder hierarchy of your site.

- To view a thumbnail of an imported image, hover over it.
- To save an asset to your computer, hover over or select it and click  | **Download**.
- To remove an asset from your site if you're a site administrator or designer, hover over or select it and click  | **Delete**. If the asset is being used in your site, you see a confirmation message with a list of locations where that asset is in use. After an asset file is deleted, it exists in the Salesforce cache for up to 24 hours. After this period it is permanently deleted from our systems.

-  **Note** An asset can still be accessible if it's cached locally by a browser, or cached by an external system.
- To rename an asset on your site if you're a site administrator or designer, hover over the asset and click  | **Rename**.

Export Assets

Designers and site administrators can export all site assets separately from the `.site` file. You can export assets together or individually.

Creating and Managing Folders

As a site administrator or designer, you can create folders to manage your pages, style sheets, templates, and assets.

See Also

[Add Site.com Page Elements](#)

[Using Site.com Studio as a Site Administrator or Designer](#)

[Using Site.com Studio as a Contributor](#)

[Export Assets](#)

Export Assets

Designers and site administrators can export all site assets separately from the `.site` file. You can export assets together or individually.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

When you export all assets, the assets are exported to a zipped file that is named after the site name—for example, `sitename-Assets.zip`.

Export your assets from the Overview tab by clicking  | **Export Site Assets**.

To export a single asset, hover over the asset and select **Download** from the Actions menu.

 **Note** There is a file size limitation when using the Metadata API to deploy a site from sandbox to production. The assets in the `.site` file can't be larger than 40 MB. The site gets created, but the

assets show in the new site as broken. To fix the assets, export the assets from the sandbox environment separately and then import them into your new site.

See Also

[Import and Manage Assets](#)

Creating and Managing Folders

As a site administrator or designer, you can create folders to manage your pages, style sheets, templates, and assets.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To create new folders:

1. In the All Site Content view on the Overview tab, click **New Folder**.
2. Type in the folder name.
3. Click **Create**.

Folders are created at the top level of the folder tree. Once created, you can drag them anywhere in the tree structure. Likewise, you can drag and drop files into the folders you create. To rename, delete, and create sub-folders, right-click the folder or use the Actions menu ().

 **Note** The site map remains the same regardless of how you arrange folders in the All Site Content view.

See Also

[Understand the Site Administrator and Designer's Overview Tab](#)

Edit Site.com Pages as a Designer or Site Administrator

When working with page templates and site pages, you can add content, structure, and style, all in one place.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Open a page or template on the Overview tab by double-clicking it or hovering over it and clicking  **Edit**. The page opens as a new tab.



- Using the  Page Elements and  Page Structure panes (1), you can search for and [add page elements](#) to a page, and [reorder the page element hierarchy](#).
- Using the  Properties,  Style, and  Events panes (2), you can set properties, [add style](#), and [create events](#) for a selected page or element.
- Using the toolbar (3), you can:
 - Undo and redo your actions.
 - Cut, copy, or paste page elements.
 - [Import assets](#), such as images and files.
 - [Preview pages to see how they'll appear when live](#) on different devices.
 - [Preview your site](#) or generate an anonymous preview link to send to other users.
 - [Publish your recent changes](#).
 - [Access other page actions](#) (), such as renaming or deleting the page.
- On the page canvas (4), you can [lay out the page](#) and [select, edit, and move page elements](#).



Tip

- Hide the side panes to increase the canvas size by clicking and . To reopen a pane, click its icon.
- As you edit a page, your changes are saved and the status icon (on the page tab is updated automatically.
- If the site page or page template is based on another template, editable page elements are highlighted with a blue border on the page.

[Site.com Page Templates Overview](#)

Before you begin building the pages of your website, take some time to plan the pages you need, and in particular, which pages have a similar layout. When you've decided on the layout, the quickest method is to use a page template to build the basic layout.

[Create Site.com Page Templates](#)

A page template lets you define the layout and functionality of site pages in one location. By adding common page elements to the template and then basing site pages on it, you can achieve a consistent look and feel throughout your site. And because a template-based page inherits the template's elements, you can make site-wide changes from one location.

[Create Editable Template Areas](#)

As the template creator, you specify which elements users can edit in pages based on the template.

[About Editable Page Elements](#)

When you mark a page element as **Editable** on a page template, that page element becomes editable in any child pages or templates derived from the parent template.

[Create Site.com Pages](#)

As a site administrator or designer, when you create a site page, you can choose to base it on a layout or page template.

[Identify Which Template a Site.com Page Uses](#)

When you edit a template-based page, you can't modify its non-editable page elements. You also can't reposition, resize, or delete editable page elements, or alter the events, properties, or style associated with them. To update these elements or properties, you must edit them in the template the page is based on.

[Rename, Duplicate, and Convert Pages](#)

When working with pages and templates, you can perform common tasks, such as renaming, deleting, or duplicating pages.

[Changing a Page's Doctype Property](#)

The Document Type Definition (DTD) or *doctype* of a page defines which version of HTML it's using. This information is used by some browsers to trigger a standard rendering mode. By default, each page's doctype is set to HTML5, which is the latest version, but you can change it to XHTML 1.0.

See Also

[Using Site.com Studio as a Site Administrator or Designer](#)

Site.com Page Templates Overview

Before you begin building the pages of your website, take some time to plan the pages you need, and in particular, which pages have a similar layout. When you've decided on the layout, the quickest method is to use a page template to build the basic layout.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

About Page Templates

A page template lets you define the layout and functionality of site pages in one location. By adding common page elements to the template and then basing site pages on it, you can achieve a consistent look and feel throughout your site. Page templates don't appear on your public site.

As the template creator, you specify which elements users can edit in pages based on the template. By default, a page element in a template is “locked,” so users can't edit its contents in any template-based page unless you mark the page element as “editable.” Conversely, when users edit an editable page element in a template-based page, their changes are specific to that page and don't affect your template.

For example, this main page template contains a non-editable header and navigation menu that are common to all the pages in the site (1). The main template also has an editable center panel (2) to house the page-specific content of each page that's based on it.



Note

- Page templates must contain at least one editable page element. Otherwise, users can't edit site pages that are based on the template.
- **Panels** are ideal for adding editable areas to page templates.

You can use page templates to:

- Save time and effort by laying out the page structure and using it as a starting point when you create site pages. For example, you could design a template with a fixed header panel and side menu, and an editable center panel, to which you add page-specific page elements and content.
- Quickly make global updates to the layout or style of your website, as any changes you make to the template's design are reflected immediately in all the pages that use it.

- Control how other users (such as contributors or other site administrators and designers) can modify site pages. For example, you can allow contributors to edit specific content blocks only.
- Ensure your template design remains pixel-perfect. When users edit a page that's based on a template, their changes don't affect your template.
- Reuse common design elements by creating child templates.
- Allow contributors to create site pages that are based on the template.

About Child Templates

Child templates are a useful way to reuse common design elements for more complicated page layouts. For example, your website probably has elements that are the same on every page in your site, such as a navigation menu. However, several pages can have elements that are common only to them, such as pages in a subsection of your site that include a subsection header. By using a child template, which is a template that's based on another template, you can reuse the main template design.

Using our main page template as a base, the child template inherits the non-editable header and navigation menu (1), and an editable center panel (2) where we add the non-editable subsection header (3). We must also add a new editable center panel (4) because the center panel of the main template is editable only in pages directly based on the main template.



Now, any page based on the child template includes the non-editable main header, navigation menu, and subsection header, and an editable center panel (5) for that page's content.



Best Practices

- Plan your site structure and the layout of your pages. Taking the time to plan your website first saves time when you build your site.
- Identify which elements are common to all the pages of your site, such as navigation menus or headers, as these elements can be added to the page template.
- Use page templates wherever possible to promote content reuse and save time.
- Try to keep the design of your main page template as simple as possible to make it easier to modify in the future. For more complicated site designs, use [child templates](#) to achieve maximum flexibility.

See Also

[Create Site.com Page Templates](#)

[Create Site.com Pages](#)

[Set Up the Contributor's Studio View](#)

[Identify Which Template a Site.com Page Uses](#)

Create Site.com Page Templates

A page template lets you define the layout and functionality of site pages in one location. By adding common page elements to the template and then basing site pages on it, you can achieve a consistent look and feel throughout your site. And because a template-based page inherits the template's elements, you can make site-wide changes from one location.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

You can create a page template from a layout, or if you've already created a template, you can use it as a base to create a [child template](#), which lets you reuse the design of the main template.

Create a Page Template from a Layout

To start from scratch with a blank template or use a basic page layout:

1. Hover over **Page Templates** on the Overview tab and click **New**, or click **New Page Template** in the Page Templates view.
2. Enter the page template name. Template names can't include special characters, such as #, ?, or @.
3. Click **Layouts** and select either a blank page or a predefined page layout, such as a page with a header and footer.



Note Predefined page layouts use [panels](#) to create columns, headers, and footers. These panels use inline CSS to set their position, so you can easily modify the layout after the page is created. However, if you're familiar with CSS and prefer using CSS rules, you can remove the inline style by selecting the panel, deleting the code from the Code tab in the Style pane (), and clicking **Apply**.

4. Choose a layout mode:
 - To expand the page to fill the width of the browser window, click **Full width**.
 - To set the page width, click **Fixed width** and enter the width.

5. Click **Create**. The page template opens.

Next, you must [complete the template](#).

Tip

- By default, any template you create is only available to other site administrators and designers in your organization. To let contributors create pages based on the template, select **Available to Contributor** in the Properties pane ().
- You can also create templates by [converting or duplicating other pages](#).

Create a Child Template

To use an existing template as a base for a child template:

- The quickest option is to:
 1. Select the template in the Page Templates view on the Overview tab and click  | **Create Child Template**. Alternatively, click **Page Actions** | **Create Child Template** if the template is open.
 2. Enter the page template name. Template names can't include special characters, such as #, ?, or @.
 3. Click **Create**. The child template opens.
- Alternatively:
 1. Hover over **Page Templates** on the Overview tab and click **New**, or click **New Page Template** in the Page Templates view.
 2. Enter the page template name. Template names can't include special characters, such as #, ?, or @.
 3. Click **Page templates** and select the page template.
 4. Click **Create**. The child page template opens.

Complete Your Template

After you've created a template, you must take these next steps to complete it.

- [Lay out the page template](#)
- [Add other page elements to the template](#)
- [Create editable areas](#)
- [Create template-based site pages](#)

See Also

[Create Editable Template Areas](#)

[Identify Which Template a Site.com Page Uses](#)

[Site.com Page Templates Overview](#)

[Set Up the Contributor's Studio View](#)

Create Editable Template Areas

As the template creator, you specify which elements users can edit in pages based on the template.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To make a page element editable in derived pages or child templates, select it on the page template or in the Page Structure pane () and click  | **Make Editable**, or select **Editable** in the Properties pane ().

When the page template is open, editable page elements are highlighted with a blue border on the page. They also display a pencil icon () in the Page Structure pane and in the information popup that appears when you hover over the element on the page.



See Also

[About Editable Page Elements](#)

[Set Up the Contributor's Studio View](#)

[Create Site.com Page Templates](#)

About Editable Page Elements

When you mark a page element as **Editable** on a page template, that page element becomes editable in any child pages or templates derived from the parent template.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Consider these tips when creating editable page elements.

- Editable page elements are highlighted with a blue border in child pages and templates.
- If you make a page element within a panel editable, you can't also make the container panel editable.
- Users can't alter the events, style, or properties of an editable page element in pages based on the template.
- Users can't resize, reposition, or delete editable page elements in pages based on the template. However, if the element's [Auto Height property](#) is enabled in the template, its height will adjust to fit the content of the template-based page.
- Deleting an editable element from a page template removes it from all child pages and templates.
- When you enable the **Editable** property of a page element, any pages or child templates based on the template also inherit the enabled status. In turn, the enabled status in the child template cascades to any of *its* children, and so on. If you don't want its editability to cascade to any lower levels, disable the **Editable** property of a page element in a child template.

Editable Page Elements for Contributors

- Contributors can modify editable content blocks in site pages based on the template. They can also edit content blocks that you place in an editable panel in template-based site pages.
 **Tip** To add a content block that only other site administrators or designers can edit, use [custom code](#) instead.
- Contributors can add content blocks to editable panels in site pages based on the template. If you make widgets available to contributors, they can also add them to editable panels.
- Site administrators and designers can edit any page element you make editable.

Default Content in Editable Page Elements

The content of all editable page elements on a child page or template is linked to the content of the editable elements on its parent page template. When you update the content of an editable page element on the parent template, the changes are pushed down to any child pages or page templates. However, if you modify the content of an editable page element at the child page or template level, you break the link between the elements, and any subsequent changes made to the page element on the parent template won't trickle down to its children. (To return control of the content to the parent template, select the editable page element on the page or in the Page Structure pane and click  | **Revert to Parent Content**. When you do this, any custom content in the editable page element is lost.)

Disabling the **Editable** property of a panel in a parent template overrides any changes made to that panel in child pages or templates. Changes to the panel at the child level disappear, and the panel reflects only the content from the parent template. However, the changes at the child level aren't lost. Re-enabling the **Editable** property of the panel in the parent template restores the custom content previously added to its children. Any changes made to the element at the parent level will no longer show up.

See Also

[Create Editable Template Areas](#)

- [Set Up the Contributor's Studio View](#)
- [Create Site.com Page Templates](#)
- [Edit and Work with Site.com Page Elements](#)

Create Site.com Pages

As a site administrator or designer, when you create a site page, you can choose to base it on a layout or page template.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

As a site administrator or designer, when you create a site page, you can choose to base it on a [page template](#). If you're creating several site pages that have common page elements, such as a navigation menu, you can save time and effort and achieve a consistent look and feel by [creating a page template](#) first and then basing your site pages on it. Alternatively, if none of your site pages have a similar structure or if you need to create a one-off site page that doesn't follow the overall site design, such as a home page, you can create a page based on a basic layout.

Create Site Pages from a Layout

Start from scratch with a completely blank page or use a basic page layout.

1. Hover over **Site Pages** on the Overview tab and click **New**, or click **New | Site Page** when the Site Pages view is open.
2. Enter the site page name. Page names can't include spaces or special characters, such as #, ?, or @.
3. Click **Layouts** and select either a blank page or a predefined page layout, such as a page with a header and footer.



Note Predefined page layouts use [panels](#) to create columns, headers, and footers. These panels use inline CSS to set their position, so you can easily modify the layout after the page is created.

However, if you're familiar with CSS and prefer using CSS rules, you can remove the inline style by selecting the panel, deleting the code from the Code tab in the Style pane (), and clicking **Apply**.

4. Choose a layout mode:
 - To expand the page to fill the width of the browser window, click **Full width**.
 - To set the page width, click **Fixed width** and enter the width.
5. Click **Create**. The site page opens.

Create Site Pages from a Page Template

If you created a page template, you can base your site pages on it.

The quickest option is to:

1. Select the template in the Page Templates view and click  | **Create Page from Template**. Alternatively, click **Page Actions** | **Create Page from Template** if the template is open.
2. Enter the site page name. Page names can't include spaces or special characters, such as #, ?, or @.
3. Click **Create**. The site page opens.

Alternatively:

1. Hover over **Site Pages** on the Overview tab and click **New**, or click **New** | **Site Page** when the Site Pages view is open.
2. Enter the site page name. Page names can't include spaces or special characters, such as #, ?, or @.
3. Click **Page templates** and select the page template.
4. Click **Create**. The site page opens.

 **Tip** You can also create pages by [converting or duplicating other pages](#).

See Also

- [Edit Site.com Pages as a Designer or Site Administrator](#)
- [Add Site.com Page Elements](#)

Identify Which Template a Site.com Page Uses

When you edit a template-based page, you can't modify its non-editable page elements. You also can't reposition, resize, or delete editable page elements, or alter the events, properties, or style associated with them. To update these elements or properties, you must edit them in the template the page is based on.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To identify which page template a site page is based on, hover over the site page in the Sites Pages view on the Overview tab. An information popup appears that displays the page template's name.

The screenshot shows the 'Site Pages' tab selected in the navigation bar. Below it is a 'New Site Page' button and two icons. The main area is titled 'Page Name' and contains a 'Site Map'. A 'Home' page is highlighted with a yellow box, and a tooltip appears below it stating 'Template: Page Template'.

Examine the Page Structure pane when the page is open. The template's name is displayed as a link that you can click to open the template.

The screenshot shows the 'Page Structure' pane. At the top are three icons: back, forward, and search. Below is a search bar with the placeholder 'Search Structure...'. Underneath is a section titled 'Using Page Template:' with a link 'My Page Template' highlighted with a red box. At the bottom is a 'Page' section with a 'Content Block' option.

Tip To view and open the site pages associated with a particular page template, select or hover over the page template in the Page Templates view of the Overview tab and click | **Edit Pages Based on Template**. Click a listed site page to open it.

See Also

[Create Site.com Page Templates](#)

[Site.com Page Templates Overview](#)

Rename, Duplicate, and Convert Pages

When working with pages and templates, you can perform common tasks, such as renaming, deleting, or duplicating pages.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To access more page options, select or hover over a page on the Site Pages view of the Overview tab and click .

To access more template options, select or hover over a template in the Page Templates view of the Overview tab and click .

Alternatively, if the page or template is open, click  on the toolbar.

The available options vary for pages and templates:

Select	To...
Edit	Open the page or template for editing. Alternatively, double-click the page or template.
Rename	Change the page or template name.
Preview	View the page in a browser window.
Duplicate	Create a copy of the page or template.

Select	To...
	Duplicating a page template doesn't duplicate the pages or templates based on it.
Delete	Remove a page or template. You can't delete a template that has pages based on it.
Create Child Template	Create a child template based on the selected template.
Create Page from Template	Create a page based on the template.
Convert Site Page to Template	Change the page into a template.
Convert Template to Site Page	Change the template into a page. You can't convert a template that has pages based on it.
Edit Pages Based on Template	View and open the site pages that are based on the selected page template.
Add IP Restrictions	Control access to the page or template by restricting the range of permitted IP addresses . When you create a page that's based on a template with IP restrictions, the page inherits the IP restrictions.

See Also

- [Edit Site.com Pages as a Designer or Site Administrator](#)
- [Using Site.com Studio as a Site Administrator or Designer](#)

Changing a Page's Doctype Property

The Document Type Definition (DTD) or *doctype* of a page defines which version of HTML it's using. This information is used by some browsers to trigger a standard rendering mode. By default, each page's doctype is set to HTML5, which is the latest version, but you can change it to XHTML 1.0.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

When the page is open:

1. Select the page in the Page Structure pane.
2. In the Properties pane, select an option in the Doctype drop-down list.

See Also

[Changing a Page Element's HTML Tag](#)

[Adding Custom HTML Attributes](#)

[HTML5 Semantic Page-Layout Tags](#)

Site.com Page Elements

Page elements are the building blocks of your site pages and page templates. Combined, they provide the page's structure and content.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

When a site page or page template is open in Site.com Studio, these page elements are available to site administrators and designers.

Page Element	Description
Content Block	Contains the page text, and can also house images, media, and hyperlinks. Also available to contributors if the page has editable areas.
Custom Code	Lets you customize your site by adding markup, such as HTML and JavaScript for elements that aren't provided in Site.com Studio.

Page Element	Description
Image	Adds images directly to the page.
Breadcrumb	Adds a breadcrumb navigation element to the page.
Menu	Creates a menu that lets users navigate through the pages of your site.
Panel	Adds structure to the page and lets you group other page elements together.
Button	Adds a button to the page. You can use the actions in the Events pane to add functionality to the button.
Form	Lets you create web-to-lead forms or gather customer feedback, and submit the data to Salesforce objects.
Input Fields	Provides several field types to add to forms or pages. When added to a form, binds to fields in the form's object. See Input Field Types .
Data Element	Must be contained in a data repeater. Binds to a field in the data repeater's object and acts as a placeholder that shows the content of a specified field for the current record.
Data Function	Connects to a standard or custom Salesforce object, performs calculations on the returned results, and displays the calculation on the page.
Data Repeater	Connects to a Salesforce object and returns a dataset based on filters that you specify. Combines with data elements or custom code page elements to display the results on the page.
Data Table	Connects to a standard or custom Salesforce object, retrieves a data set based on the filter criteria that you specify, and displays one or more records as rows in the table.

Add Site.com Page Elements

Page elements are the building blocks of your site pages and page templates. Panel elements add structure to your pages. You can think of both pages and panels as “containers” for the page elements that you add to them.

Edit and Work with Site.com Page Elements

The Page Structure pane displays the hierarchy of all elements on the page and is a very useful way of selecting, moving, and reordering elements, particularly for more complicated page designs.

[Lay Out Site.com Pages Using Panels](#)

A panel is a useful layout tool that defines the logical divisions of your page and lets you group page elements together for easy movement and positioning. Think of it as a container for other page elements, including other panels, or as a `div` that wraps around the content placed within it. Panels are ideal for adding editable areas to page templates.

[Position Panels Using CSS](#)

A panel is a useful layout tool that defines the logical divisions of your page. Using CSS, you can position panels and improve the layout of the page.

[Add Images Directly to the Page](#)

As a site administrator, you can add images directly to your site pages and page templates or you can add images to content blocks.

[Add Content Blocks to Pages](#)

Content blocks contain the text of your website pages, and can also house images, videos, and hyperlinks. Designers and site administrators can add content blocks to pages when in Design Mode.

[Add Custom Code to Pages](#)

Custom code lets you customize your site using markup, such as HTML and JavaScript.

[About the Site Map and Page Hierarchy](#)

The Site Pages view on the Overview tab contains the pages and site map links of your website. The Site Map folder reflects the hierarchy or tree structure of your site by housing site pages and links that are included in the site map. When you create pages or site map links, they're automatically added to this folder. The Landing Pages folder houses standalone pages that are excluded from the site map, making it ideal for temporary pages, such as promotional or competition pages.

[Adding Custom HTML Attributes](#)

You can add custom HTML attributes to pages and page elements, which are rendered on the HTML tag of the page element. For example, this is useful when working with third-party frameworks that render page elements differently based on certain attributes.

[Changing a Page Element's HTML Tag](#)

By default, panels, data repeaters, data elements, custom code, and content blocks are each defined as a `div`, but you can change this to any other HTML tag using the **HTML Tag** property. This gives you greater flexibility and control over how the page element is displayed on the page.

[HTML5 Semantic Page-Layout Tags](#)

HTML5 defines several semantic page-layout tags that describe the content they contain. These tags make it easier for search engines and screen readers to read and organize your content.

See Also

[Add Site.com Page Elements](#)

[Edit and Work with Site.com Page Elements](#)

Add Site.com Page Elements

Page elements are the building blocks of your site pages and page templates. Panel elements add

structure to your pages. You can think of both pages and panels as “containers” for the page elements that you add to them.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

If a site page or page template is based on another [page template](#), you can only add page elements to [editable panels](#), which are highlighted with a blue border on the page.

Pages, panels, data repeaters, and forms are “container” page elements, so you can add other page elements to them when the page is open.

- In the Page Elements pane () , either:
 - Drag the page element onto the page canvas or container page element.
 - Click the page element, select where to place it in the popup that appears, and click **Apply**.
- In the Page Structure pane () , hover over a container page element and click  | **Add Page Elements**. Click the item you want to add or drag it onto the container page element.
- Select a container page element on the page and click  | **Add Page Elements**. Click the item you want to add or drag it into the container page element.

When you drag a page element into an editable panel, the page element displays a permitted icon and a green border shows where you're placing the element.



If you try dragging a page element into a panel that isn't editable, the page element displays a not-permitted icon.



See Also

[Edit and Work with Site.com Page Elements](#)

- [Add Images Directly to the Page](#)
- [Add Content Blocks to Pages](#)
- [Add Custom Code to Pages](#)
- [Add a Navigation Menu](#)

Edit and Work with Site.com Page Elements

The Page Structure pane displays the hierarchy of all elements on the page and is a very useful way of selecting, moving, and reordering elements, particularly for more complicated page designs.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

- To select a page element, either click it on the page, or select it in the Page Structure pane (). This highlights the element on the page and displays the item's selector bar and **Actions** menu ().
- To edit a page element, such as a content block, image, or custom code, either:
 - Double-click the element on the page.
 - Select the element on the page and click  | **Edit** on its selector bar.
 - Select or hover over the element in the Page Structure pane and click  | **Edit**.
 - Click **Edit Content** on the toolbar (for content blocks only).
- To move a page element, either:
 - Select the element on the page and drag it to the correct position, or click  | **Move** on its selector bar.
 - Drag the item to your preferred location in the Page Structure pane, or select it and click  | **Move** | **Direction**. You can also drag all page elements other than panels to your preferred location in the Page Structure pane.
- To resize a page element, select it and drag the resize handles to the correct size. If the corner resize handles are unavailable, it means the item's Auto Height property is enabled, which adjusts the height depending on its contents. To resize it to a set height, disable the property by either deselecting **Auto**

Height in the Properties pane (), or by clicking one of the bottom resize handles and clicking **Disable Auto Height** in the popup message that appears.



Tip If you disable the Auto Height property on an image, but you want it to retain its aspect ratio—the relationship of height to width—press and hold down the SHIFT key while you drag to resize it.

Alternatively, if you're using CSS to style the page element, adjust the style of the class or ID that styles it.

- To delete an element, select it and either:
 - Click  | **Delete** on the item's selector bar.
 - Press **DELETE**.
 - Click  on the toolbar.
 - In the Page Structure pane, click  | **Delete**.

If a site page or page template is based on another page template:

- The content of all editable page elements on a child page or template is linked to the content of the editable elements on its parent page template. When you update the content of an editable page element on the parent template, the changes are pushed down to any child pages or page templates. However, if you modify the content of an editable page element at the child page or template level, you break the link between the elements, and any subsequent changes made to the page element on the parent template don't trickle down to its children.
- You can't reposition or resize its page elements. However, if the element's Auto Height property is enabled in the template, the height adjusts to fit the content in the template-based page. To edit the page element, you must edit it in the page template.
- You can't delete its page elements. To delete the page element, you must delete it from the page template.
- You can't alter the events, properties, or style of an editable page element, such as its color, position, and size.

See Also

- [Site.com Page Elements](#)
- [Add Site.com Page Elements](#)
- [Add Content Blocks to Pages](#)
- [Lay Out Site.com Pages Using Panels](#)
- [About Editable Page Elements](#)

Lay Out Site.com Pages Using Panels

A panel is a useful layout tool that defines the logical divisions of your page and lets you group page elements together for easy movement and positioning. Think of it as a container for other page elements, including other panels, or as a `div` that wraps around the content placed within it. Panels are ideal for adding editable areas to page templates.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

When you create a page template or site page, you can use predefined page layouts that include headers, footers, and columns, which are created using panels. After it's created, you can then further modify the layout to match your site's design.

If you need to add more divisions to the page and you're not familiar with CSS, the easiest method is to use row and column panels. This feature adds panels with predefined CSS positioning to ensure they align correctly on the page.

-  **Note** Predefined page layouts, and row and column panels use inline CSS to set their position. If you're familiar with CSS and are using CSS rules to style your site, you can remove the inline CSS by deleting it from the Code tab in the Style pane () and clicking **Apply**.

To add row and column panels to a page:

1. Select the page (the top folder icon) in the Page Structure pane ().
2. Click  | **Add Rows and Column Panels**.
3. Select the number of row or column panels you require. If the page already contains content, it is placed in the first new panel.

To add a row panel:

- Above a panel, select the panel on the page or in the Page Structure pane and click  | **Add Rows and Column Panels** | **Insert Row Above**
- Below a panel, select the panel on the page or in the Page Structure pane and click  | **Add Rows and Column Panels** | **Insert Row Below**

To add row and column panels to another panel:

1. Select the panel on the page or in the Page Structure pane.
2. Click  | Add Rows and Column Panels.
3. Select the number of row or column panels you require. If the page already contains content, it is placed in the first new panel.

To add a single panel, drag a **Panel** from the Page Elements pane () onto the page.

By default, the height of a panel automatically adjusts when you add content to it because its Auto Height property is enabled. You can [disable the property to resize and reposition](#) panels. If you hover over a panel on the page, an information popup appears that displays the width and height of the panel.



When you drag a page element onto a panel, the edges change color, indicating that the element is now grouped within it. To remove the element from the group, drag it outside the panel.

See Also

- [Add Site.com Page Elements](#)
- [Position Panels Using CSS](#)
- [Changing a Page Element's HTML Tag](#)

Position Panels Using CSS

A panel is a useful layout tool that defines the logical divisions of your page. Using CSS, you can position panels and improve the layout of the page.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Adding Padding and Margins to Panels

Two CSS properties—[margins and padding](#)—can help with your page layout by creating space between the rows and columns, and the content within. The margin property controls the space outside the panel between its border and outer edge, while the padding property controls the space between the panel's content and border.

To add margins and padding:

1. Select the panel.
2. Open the Dimensions section of the Style pane.
3. In the Margins section, either:
 - Set the margin width for all four sides by entering a value in the All text box and selecting the unit of measurement.
 - Set the margin widths for the top, right, bottom, or left sides independently by entering a value in the appropriate text box and selecting the unit of measurement.
4. Similarly, in the Padding section, set the padding widths as required. Adding padding increases the total width of the panel. For example, if you have a panel with a width of 500px and you add padding of 20px to all sides, the total width of the panel will be 540px.

 **Tip** You can center a panel or block page element using the margin property. Enter `0` in the All text box and select **Auto** in the drop-down list.

Creating Column Panels Using the Float Property

If you need to add more divisions to the page and you're not familiar with CSS, the easiest method is to use [row and column panels](#). Alternatively, using the CSS [float](#) property, you can position panels to the left or right to create columns. (When you add panels using the row and column panels feature, they're automatically positioned using the float property.) For example, you could add two single panels to a container panel and set both panel's float and width properties to create a two-column page layout.

To create a column panel:

1. Select the panel.
2. Open the Layout section of the Style pane.
3. Click  to float the panel to the left, or click  to float the panel to the right. If you're creating a two-column layout, for example, ensure you set the float property of both panels.
4. Adjust the width of the panel to ensure the panels align correctly by either setting the width in the Dimensions section or dragging the panel's border on the page. For example, if you're creating two columns of equal width, set the width of both panels to 50%.

 **Tip** When you use the float property, remember to set the overflow property of the container panel to "hidden." This allows the container panel to grow as the height of the column panels increase. Select the container panel and in the Layout section of the Style pane, select **Hidden** in the

Overflow drop-down list.

See Also

- [Cascading Style Sheets Overview](#)
- [Lay Out Site.com Pages Using Panels](#)

Add Images Directly to the Page

As a site administrator, you can add images directly to your site pages and page templates or you can add images to content blocks.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To add an image directly to the page:

1. Open the site page or page template.
2. Drag an **Image** from the Page Elements pane onto the page.
3. In the Add an Image dialog box, either:
 - Find an existing image by typing its name in the **Search Image** text box and selecting it from the list.
 - Upload an image from your computer in the Upload tab by browsing to the image, clicking **Upload**, and selecting it from the list.
4. Click **Apply**. The image is added to the page.
5. Enter a brief description of the image in the **Alternative Text** field in the Properties pane. If the browser can't display the image, the description is used by screen reader users or as a substitute. A description can also help with search engine optimization (SEO).

See Also

- [Add Site.com Page Elements](#)
- [Edit and Work with Site.com Page Elements](#)

Add Content Blocks to Pages

Content blocks contain the text of your website pages, and can also house images, videos, and hyperlinks. Designers and site administrators can add content blocks to pages when in Design Mode.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To add a content block when the page is open, either drag it from the Page Elements pane onto the page or [target a container page element](#).

To [edit a content block](#), double-click it. For greater control over the text, you can edit the HTML directly by selecting the content block and clicking  | [Edit HTML](#).

If you make a content block in a page template editable, contributors can edit the content block in any page based on the template. To add a content block that only other site administrators or designers can edit, use [custom code](#) instead.

[Understand the Content Editing Toolbar](#)

Designers and site administrators can edit content blocks when in Design mode. Content blocks contain the site page's text, along with images, videos, and hyperlinks.

[Edit Content Blocks in Design Mode](#)

Designers and site administrators can edit content blocks on the page. Content blocks contain the text for site pages, and can also house images, videos, and hyperlinks.

[Add Images to Content Blocks in Design Mode](#)

Designers and site administrators can add images to content blocks when viewing a page in Design Mode.

[Add Video to Content Blocks in Design Mode](#)

Designers and site administrators can add YouTube®, Google®, Adobe® Flash®, Windows Media®, and Apple QuickTime® videos to content blocks when viewing a page in Design Mode.

Attach Hyperlinks to Text and Images in Design Mode

When viewing a page in Design Mode, designers and site administrators can create hyperlinks to external Web pages or websites, pages and assets in the site, email messages, and anchors on the page.

Add Anchors to Pages in Design Mode

An anchor is an invisible marker that identifies a particular location on a page. If you're a designer or site administrator, you can add an anchor to the page and then create a hyperlink that jumps to that specific location, which can be useful if a page is particularly long. For example, if your page has several sections, you could add links to each section at the top of the page to aid navigation.

See Also

[Add Images to Content Blocks in Design Mode](#)

[Attach Hyperlinks to Text and Images in Design Mode](#)

Understand the Content Editing Toolbar

Designers and site administrators can edit content blocks when in Design mode. Content blocks contain the site page's text, along with images, videos, and hyperlinks.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition



When in Design Mode, you can use the content editing toolbar to:

- Undo and redo your edits, and remove the formatting of text copied and pasted from Microsoft® Office, which can often include hidden formatting (1).
- Cut, copy, and paste text (2).
- Apply direct formatting (3), such as:
 - Font family and size
 - Bold, italic, underline, subscript, superscript, and strikethrough
 - Font and highlight color
- Control the text style and layout (4) by:
 - Applying paragraph and heading styles
 - Setting paragraph indentation

- Left-aligning, centering, right-aligning, or justifying text
- Inserting numbered or bulleted lists
- Insert a table, and add rows, columns, and spacing (5).
- Add [images](#), [videos](#), and special characters (6).
- Add and remove [hyperlinks](#) and [anchors](#) (7).

 **Tip** Avoid applying formatting, such as different fonts or highlighting, directly to text whenever possible. Instead, it's best practice to use the paragraph and heading styles to quickly apply consistent formatting throughout the site. Using paragraph and heading styles also ensures that all page text is updated automatically if a site administrator or designer modifies these styles.

Edit Content Blocks in Design Mode

Designers and site administrators can edit content blocks on the page. Content blocks contain the text for site pages, and can also house images, videos, and hyperlinks.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

When the page is open in Design Mode:

1. Double-click the content block on the page.
2. Add or edit text and format it using the [content editing toolbar](#).

 **Tip**

- If you copy and paste text from Microsoft® Office, highlight the text and click  to remove any hidden formatting, which can adversely affect how the text appears on the page.
- Avoid applying formatting, such as different fonts or highlighting, directly to text whenever possible. Instead, it's best practice to use the paragraph and heading styles to quickly apply consistent formatting throughout the site. Using paragraph and heading styles also ensures

that all page text is updated automatically if a site administrator or designer modifies these styles.

3. Add [images](#), [videos](#), [hyperlinks](#), or [anchors](#) as required.
4. Click **Save**.

 **Note** The content of all editable page elements on a child page or template is linked to the content of the editable elements on its parent page template. When you update the content of an editable page element on the parent template, the changes are pushed down to any child pages or page templates. However, if you modify the content of an editable page element at the child page or template level, you break the link between the elements, and any subsequent changes made to the page element on the parent template don't trickle down to its children.

Add Images to Content Blocks in Design Mode

Designers and site administrators can add images to content blocks when viewing a page in Design Mode.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

Site.com Contributor User

AND

Contributor role assigned at the site level

When the page is open:

1. Double-click the content block on the page.

2. Position your cursor where you want to insert the image and click .
3. In the Image Properties dialog box, either:
 - Enter a URL to an image in the Image URL field.
 - Select an image from your website by clicking **From Website** and selecting the image in the list that appears.
 - Upload an image from your computer by opening the Upload tab, browsing to the image, and clicking **Upload**.
4. Enter a brief description of the image in the Alternative text field. If the browser can't display the image, the description is used by screen reader users or as a substitute. It can also help with search engine optimization (SEO).
5. Optionally, preview how the image appears in relation to the text on the page and set:
 - The width and height of the image
 - How much space surrounds the image (which is controlled by the **HSpace** and **VSpace** properties)
 - How it aligns with the text on the page
 - The image border (for example, to set a dotted green border that's 10 pixels wide, you enter **10px dotted green** in the **Border** field)
6. Click **Apply**.
7. Click **Save**.

See Also

[Edit Content Blocks in Design Mode](#)

[Understand the Content Editing Toolbar](#)

[Import and Manage Assets](#)

Add Video to Content Blocks in Design Mode

Designers and site administrators can add YouTube®, Google®, Adobe® Flash®, Windows Media®, and Apple QuickTime® videos to content blocks when viewing a page in Design Mode.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

USER PERMISSIONS NEEDED

Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

Site.com Contributor User

AND

Contributor role assigned at the site level

When the page is open:

1. Double-click the content block on the page.
 2. Position your cursor where you want to insert the video and click .
 3. In the Video Properties dialog box, select the video type and either:
 - Enter the URL to the video in the Video URL text box—for example, `http://www.youtube.com/watch?v=123abc`.
 - Select a video from your website by clicking **From Website** and selecting the video in the list that appears.
 - Upload a video from your computer by opening the Upload tab, browsing to the image, and clicking **Upload**.
-  **Note** You can only select or upload Flash, Windows Media, and QuickTime videos.
4. To specify how the video is displayed on the page, you can set:
 - The width and height of the video
 - How much space surrounds the video (which is controlled by the `HSpace` and `VSpace` properties)
 - How it aligns with the text on the page

You can also preview the video.

5. Click **Apply**. The video appears as an icon in the content block.
6. Click **Save**.

 **Note** You can view all video types, other than Windows Media videos, when you [preview a page](#).

See Also

[Edit Content Blocks in Design Mode](#)
[Understand the Content Editing Toolbar](#)

Attach Hyperlinks to Text and Images in Design Mode

When viewing a page in Design Mode, designers and site administrators can create hyperlinks to external Web pages or websites, pages and assets in the site, email messages, and anchors on the page.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

When the page is open:

1. Double-click the content block on the page.
 2. Select the text or image that you want to attach a hyperlink to and click  +.
 3. Select the link type in the Link to dropdown list. To link to:
 - A Web page:
 - Select **A URL**.
 - Type the address in the URL field—for example, `http://www.externalsite.com`.
 - Go to step 4.
 - A page or item in your site:
 - Select **An item in your site**.
 - Select the item type, such as a page or image.
 - Select the item. (If you can't see the list of items, place your cursor in the URL field and press the DOWN key on your keyboard.)
 - Go to step 4.
 - **An anchor** that you previously added to the page:
 - Select **An anchor on the page**.
 - Select the anchor in the dropdown list. Alternatively, enter a new anchor name and create the anchor afterwards.
 - Go to step 5.
 - An email message:
 - Select **An email**.
 - Enter the recipient's email address and the message information.
 - Go to step 5.
-  **Note** For content blocks in a data repeater, you can use [expressions](#) to add a custom link, such as a URL query string, to an item in your site or to a Web page.
4. To select which window the item opens in, select an option in the Target dropdown list:

- **Popup window** loads the item into a window. When you select this option, you can set the title for the popup and control its appearance and size with the options that appear.
 - **New window (_blank)** loads the item into a new, unnamed browser window.
 - **Same window (_self)** loads the item into the same frame or window as the link. This setting is the default.
 - **Topmost window (_top)** loads the item into the topmost parent frameset or window of the frame that contains the link.
 - **Parent window (_parent)** loads the item into the parent frameset or window of the frame that contains the link.
5. Optionally, enter a tooltip description for the link. The tooltip displays as a pop-up when the user hovers over the link.
6. Click **Apply**.
7. Click **Save**.

To delete a hyperlink, select it and click .

See Also

[Edit Content Blocks in Design Mode](#)

[Understand the Content Editing Toolbar](#)

Add Anchors to Pages in Design Mode

An anchor is an invisible marker that identifies a particular location on a page. If you're a designer or site administrator, you can add an anchor to the page and then create a hyperlink that jumps to that specific location, which can be useful if a page is particularly long. For example, if your page has several sections, you could add links to each section at the top of the page to aid navigation.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

USER PERMISSIONS NEEDED

To edit only content in Site.com sites:

Site.com Contributor User

AND

Contributor role assigned at the site level

When the page is open in Design Mode:

1. Double-click the content block on the page.
2. Place your cursor at the beginning of the line where you want to link to and click .
3. Enter a name for the anchor and click **Apply**. Ideally, use a name that helps identify the anchor's location on the page—for example, *top*.
4. Now [create a hyperlink](#) that links to the anchor.

See Also

- [Attach Hyperlinks to Text and Images in Design Mode](#)
[Edit Content Blocks in Design Mode](#)
[Understand the Content Editing Toolbar](#)

Add Custom Code to Pages

Custom code lets you customize your site using markup, such as HTML and JavaScript.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level



Tip

- Scripts can't execute while you're editing a page in Site.com Studio. To test your code, preview the page.
 - If you are building a Site.com site from an existing HTML site, avoid using the Custom Code page element to paste large chunks of HTML from the original site. Instead, use the available page elements, such as panels, content blocks, and data tables. Using these elements lets you make future updates and design changes much more easily.
-
- Add markup to a specific location on a page using the Custom Code page element. JavaScript added using the Custom Code page element loads when that part of the page loads.
 - Add markup to the page `head`. JavaScript in the page `head` loads first.
 - Add JavaScript to the page `body`. JavaScript added to the page `body` is positioned at the end of the `body` tag and only loads when the DOM is ready.
 - Add a reference to a JavaScript file or library in the page `head` or `body`.

Add Markup Directly to the Page

1. Drag a **Custom Code** page element from the Page Element pane onto the page.
2. Enter the code in the Edit Code dialog box.
3. Click **Save and Close** to add the code directly to the page.

Add Markup to the Page `Head`

1. In the Scripts section of the Properties pane, click **Configure** in the Edit Head Markup section.
2. Enter the markup in the Edit HTML Code dialog box.
3. Click **Save and Close** to insert the markup into the page `head`.

Add JavaScript to the Page `Body`

1. In the Scripts section of the Properties pane, click **Configure** in the Edit Body Scripts section.
2. Enter the code in the Edit JavaScript Code dialog box. Don't add `<script>` tags, as they're already included.
3. Click **Save and Close** to add the code to the bottom of the page `body`.

Using JavaScript Files or Libraries

Instead of adding JavaScript code directly to a page, you can include links to imported or external JavaScript files, or to an open-source library (via the Google Libraries API).

1. In the Scripts section of the Properties pane, click  in either the Body Scripts or the Head Scripts section.
2. To link to:
 - A JavaScript file that you've imported, select **An imported script** and select the file.
 - An open-source JavaScript library, select **A Google AJAX library** and select the library.
 - An external JavaScript file, select **A URL to an external script** and enter the address.
3. Click **Apply**.

See Also

- [Displaying Data or Content Using Custom Code](#)
- [Add Site.com Page Elements](#)

About the Site Map and Page Hierarchy

The Site Pages view on the Overview tab contains the pages and site map links of your website. The Site Map folder reflects the hierarchy or tree structure of your site by housing site pages and links that are included in the site map. When you create pages or site map links, they're automatically added to this folder. The Landing Pages folder houses standalone pages that are excluded from the site map, making it ideal for temporary pages, such as promotional or competition pages.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

 **Tip** If you can't see the Site Map folder in the Site Pages view on the Overview tab, click .

When adding a navigation menu to your site, it's important to organize the hierarchy of your site pages and links accurately, because this structure is used to generate the menu. Pages and site map links are displayed in navigation menus in the order you arrange them.

In this representation of the site hierarchy, you can more clearly see the tree structure.



- 1 is the Site Map folder, which contains four site pages.
- 2 is a top-level page in the site hierarchy.
- 3 is a top-level, parent page with two child pages. A child page is a page at a lower level in the site hierarchy than its parent page.
- 4 are the two child pages.
- 5 is the Landing Pages folder, which contains a temporary page that's not part of the site map or navigation menu.

 **Tip** If a page has child pages, the  icon appears beside it indicating that you can expand the branch.

By default, when you create a menu, it's generated from the pages and site map links in the Site Map folder in the Site Pages view. However, you can also create a menu that's generated from the pages in the Landing Pages folder or from the child or sibling pages of a site page.

You can hide a page in menus, breadcrumbs, and the site map by selecting the **Hide Page** checkbox

found on the Properties pane for each page. This setting also prevents website visitors from accessing the page's direct URL. By default, all pages are visible.

Add Links to a Site Map

Pages aren't the only things you can assign to your site map. Add internal or external URLs to your site map to customize your navigation menus and breadcrumbs, and achieve greater flexibility and control.

Adding Breadcrumb Navigation to Pages

Add a Breadcrumb page element to your page to help users navigate through your site and show the page's location in the site hierarchy.

Add a Navigation Menu

By default, when you create a menu, it's generated from the pages and site map links in the Site Map folder in the Site Pages view. However, you can also create a menu that's generated from the pages in the Landing Pages folder or from the child or sibling pages of a site page.

Style Navigation Menus

Navigation menus are styled using CSS themes that you can customize to match the design of your website. When you add a navigation menu to a page, it uses a default theme to control its appearance. Choose from other existing themes in the Theme Name dropdown list in the Properties pane.

See Also

[Add a Navigation Menu](#)

[Create Site.com Pages](#)

Add Links to a Site Map

Pages aren't the only things you can assign to your site map. Add internal or external URLs to your site map to customize your navigation menus and breadcrumbs, and achieve greater flexibility and control.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

For example, let's say you have a top-level Products page that has a menu based on its child pages. You also want to include a page called Testimonials in the menu, but it's not a child page of the Products page. You can create a site map link that points to the Testimonials page and add the link under Products in the site map. Now when visitors view the Products page, they see a menu consisting of its child pages, along with a menu item that takes them directly to Testimonials.

1. On the Overview tab, click **New | Site Map Link**.
2. Enter a name for the link.
3. Enter a URL. URLs can be either relative or absolute, and are case-sensitive.

 **Note** You can't preview absolute site map links in Site.com Studio unless they include a prefix, such as `http://` or `https://`.

4. Click **Create**.
The link appears at the bottom of the site map.
5. Drag the link to the correct position in the site map.

 **Note** Site map links are automatically included in navigation menus and breadcrumbs. However, you can't set a site map link as a custom root node in a breadcrumb.

See Also

- [Adding Breadcrumb Navigation to Pages](#)
- [Add a Navigation Menu](#)
- [About the Site Map and Page Hierarchy](#)

Adding Breadcrumb Navigation to Pages

Add a Breadcrumb page element to your page to help users navigate through your site and show the page's location in the site hierarchy.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Breadcrumb navigation is based on the pages and links in the site map and usually supplements menu navigation. You can hide a page in menus, breadcrumbs, and the site map by selecting the **Hide Page** checkbox found on the Properties pane for each page. This setting also prevents website visitors from accessing the page's direct URL. By default, all pages are visible.

 **Tip** To save time and effort, add a Breadcrumb page element to a page template to automatically include it on every template-based site page. The breadcrumb on each derived page dynamically updates based on its location in your site map.

1. Open the page or page template that you want to add the breadcrumb element to.
2. Drag a **Breadcrumb** from the Page Elements pane onto the page.
3. In the Properties pane, under Root, specify a custom root node for the breadcrumb. By default, the root value is set to None, which builds the breadcrumb structure based on the page's location in the site map.
 - Select *Home Page* to set the site's home page as the first item in the breadcrumb. To find out how to set a site's home page, see [Configuring Site Properties](#).
 - Select a specific site page to set it as the first item in the breadcrumb.



Note You can't set a site map link as a custom root node in a breadcrumb.

4. In the Properties pane, under Separator, you can customize the separator used between breadcrumb nodes. By default, the separator is `>`. However, you can change it to another text symbol or insert HTML code for an image in your site, such as ``.
5. To style the breadcrumb:
 - a. Click the Style pane and ensure **Class** is selected.
 - b. Choose an option from the Style drop-down list.
 - c. Adjust the values in the [Visual tab](#) as desired.

See Also

[Add a Navigation Menu](#)

[Edit and Work with Site.com Page Elements](#)

Add a Navigation Menu

By default, when you create a menu, it's generated from the pages and site map links in the Site Map folder in the Site Pages view. However, you can also create a menu that's generated from the pages in the Landing Pages folder or from the child or sibling pages of a site page.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

1. Arrange site pages and [site map links](#) in the Site Map folder or the Landing Pages folder in the order you want them to appear in the menu by dragging them to the desired location. Drag a page or site map link onto another page or link to make it a child of that item. Drag pages or links into the Landing Pages folder to exclude them from the site map.

 **Tip** If you can't see the Site Map folder in the Site Pages view on the Overview tab, click .

2. Open the page template or site page that you want to add the navigation menu to.
3. Drag a **Menu** from the Page Elements pane onto the page.
4. In the Properties pane, under Menu Source, select the pages that you want to use for the menu. By default, the Site Map folder is used to create the menu. Any site map link in the applicable hierarchy also shows up in your menu.
 - Select **Landing Pages** to create the menu from the pages in the Landing Pages folder.
 - Select **Child Pages** to create the menu from the current page's child pages.
 - Select **Sibling Pages** to create the menu from all of the pages that share the same parent as the current page.
 - Select a specific site page to create the menu from just its child pages.

 **Note** If you add a menu element to a page template, the menu doesn't display correctly in the template if you select *Child Pages* or *Sibling Pages* as the menu source, because page templates aren't part of the site map hierarchy. However, the menu appears as expected on site pages based on the page template.

5. To alter the appearance of the menu, you can select a different theme from the Theme Name drop-down list. For example, to create a drop-down menu, select **Horizontal Drop-down**. You can [modify the style of any theme](#) to suit your needs.
6. To change the name of a page in the menu, open the associated page and update its Navigation Name field in the Properties pane.
Navigation names can include spaces and special characters.

Alternatively, to change the name of a site map link in your menu, hover over the link in the Site Pages view on the Overview tab, click **Edit**, and update the name.

 **Tip**

- When you add a new page or site map link, update a page's Navigation Name property, or rearrange pages or links, the menu updates automatically to reflect the changes.

- To automatically include a menu on every site page, add the menu to a [page template](#) and base the site pages on it.
- You can hide a page in menus, breadcrumbs, and the site map by selecting the Hide Page checkbox found on the Properties pane for each page. This setting also prevents website visitors from accessing the page's direct URL. By default, all pages are visible.

See Also

[Style Navigation Menus](#)

[About the Site Map and Page Hierarchy](#)

[Add Site.com Page Elements](#)

Style Navigation Menus

Navigation menus are styled using CSS themes that you can customize to match the design of your website. When you add a navigation menu to a page, it uses a default theme to control its appearance. Choose from other existing themes in the Theme Name dropdown list in the Properties pane.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Alternatively, to customize a theme to suit your needs:

1. Select the navigation menu on the page.
2. Select a theme to use as a base in the Theme Name dropdown list in the Properties pane. Use a theme that most closely matches your site design or select *Blank* to start with a completely blank theme.
3. Open the Style pane and ensure **Class** is selected.
4. In the Style dropdown list, select the part of the menu that you want to style. When you select an item, it's highlighted for a few seconds, so you can easily see which part you're styling.

 **Tip** If you're familiar with CSS, you can also modify the style of the menu in the site's style sheet.

5. To style the selected menu item, use the [Style pane properties](#). Your changes are immediately reflected in the menu.
6. Repeat as required for each part of the menu.

See Also

[Add a Navigation Menu](#)

[About the Site Map and Page Hierarchy](#)

Adding Custom HTML Attributes

You can add custom HTML attributes to pages and page elements, which are rendered on the HTML tag of the page element. For example, this is useful when working with third-party frameworks that render page elements differently based on certain attributes.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

When the page is open:

1. Select the relevant page or page element in the Page Structure pane.
2. In the HTML Attributes section of the Properties pane, click .
3. Enter a name and value for the HTML attribute.
4. Click **Save**.

To delete an HTML attribute, select it and click .

To change the order in which an HTML attribute is rendered, select it and click  or .

See Also

- [Changing a Page Element's HTML Tag](#)
- [HTML5 Semantic Page-Layout Tags](#)
- [Changing a Page's Doctype Property](#)

Changing a Page Element's HTML Tag

By default, panels, data repeaters, data elements, custom code, and content blocks are each defined as a `div`, but you can change this to any other HTML tag using the **HTML Tag** property. This gives you greater flexibility and control over how the page element is displayed on the page.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To edit page element properties:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned in Site.com Studio

 **Warning** The **HTML Tag** property provides a powerful way to control how page elements are displayed. However, if you change a page element's HTML tag, you may generate invalid HTML. Before publishing any changes, test the page thoroughly.

To redefine a panel, data repeater, data element, custom code, or content block:

1. Select the element on the page.
2. In the **HTML Tag** field in the Properties pane, start typing the tag name.
3. In the auto-complete list that appears, select the relevant HTML tag. Alternatively, you can define your own HTML tag—for example, if you're working with a JavaScript library or if new HTML5 tags are introduced in the future. You can also remove the HTML tag on a panel, data repeater, data element, custom code, or content block to disable its ID, class, or inline styles.

 **Note** The following tags aren't included in the auto-complete list:

- `base`

- `body`
- `doctype`
- `head`
- `html`
- `meta`
- `style`
- `title`

See Also

[HTML5 Semantic Page-Layout Tags](#)

[Adding Custom HTML Attributes](#)

HTML5 Semantic Page-Layout Tags

HTML5 defines several semantic page-layout tags that describe the content they contain. These tags make it easier for search engines and screen readers to read and organize your content.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

By default, several page elements are defined as a `div`, including panels, data repeaters, data elements, content blocks, and custom code. Using a page element's **HTML Tag** property, you can change the tag to a semantic HTML5 block tags, such as:

Option	Description
Article	A section containing an independent item of content, such as a magazine article or a forum post.
Aside	A section containing content that's only superficially related to the main page content, such as a sidebar or advertising.
Details	A section containing additional details that the user can view or hide using an interactive widget. It can also include a summary section.
Header	A section containing an introduction, or a group of

Option	Description
	navigation elements.
Footer	A footer section for the page or parent section. It typically contains information about the parent section and appears at the end of the section.
Nav	A section that contains navigation links.
Section	A generic section of the page.
Summary	A summary or caption section for a details section.

 **Tip** If you use a HTML5 semantic tag, it's good practice to also [change the page's doctype](#) to HTML5.

See Also

[Changing a Page Element's HTML Tag](#)

[Adding Custom HTML Attributes](#)

Set Up the Contributor's Studio View

Control what your contributors can do in Site.com Studio.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

The contributor's Site.com Studio view lets users browse and update website content. To enable contributors to edit the site's content, however, you must first create template-based pages with editable areas and enable several properties.

To enable your contributors to:

- Edit the text on a page, create template-based pages with editable content blocks
- Add content blocks to the page, create template-based pages with editable panels
- Create site pages based on a page template, select **Available to Contributors** in the Properties pane when the page template is open
- Add widgets to the page, select **Available to Contributors** in the Widgets view on the Overview tab or in the Properties pane when the widget is open
- Update the appearance of pages and widgets, set up branding properties

To ensure that the contributor's Site.com Studio view is set up correctly, click **View Studio as a Contributor** in the site's pull-down menu (on the top toolbar). To exit, click **Return to My Studio View**.

The contributor's view is not available by default for Site.com Community sites. However, you can use a Site.com Contributor license to grant contributor access to a specific user. See *About Feature Licenses* in the Site.com help for details. Alternatively, a user can preview the Site.com Community site as a contributor by appending `?iscontrib` to the site's URL. For example:

`MyDomainName.builder.salesforce-experience.com/?iscontrib`

See Also

[Create Site.com Page Templates](#)

[Create Widgets](#)

[Site Branding Overview](#)

[The Contributor's Page Editing View](#)

Cascading Style Sheets Overview

Cascading Style Sheets (CSS) provide a flexible way to add style to the pages of your website. This collection of formatting rules governs the appearance of your pages, and lets you define the fonts, colors, layout, and other presentation features.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

By using CSS to control your fonts, you can ensure greater consistency in the appearance and layout of your pages in multiple browsers. Some of the many text properties that CSS lets you control include font family and size, text and background color, text formatting, and link color.

Using CSS, you can also position, add color to, float text around, and set margins and borders for block-level elements. A block-level element is a standalone piece of content that's visually formatted as a block. For example, content blocks (which are equivalent to `p` tags) and panels (which are the same as `div`

tags) are both block-level elements.

Site.com supports CSS3, which is the latest specification for CSS.

About Inline Styles Versus Style Sheets

In Site.com Studio, you can:

- Apply styles directly to a selected page or page element using the **Inline** option in the Style pane ().
Inline styles apply only to the selected item.
- Add style items such as CSS classes or IDs to a style sheet, and apply the style items to the selected page or page element. This approach separates the content (your web pages) from the presentation (the style sheet).

If you're not familiar with CSS, the inline option is probably the easiest to use and understand. However, inline styles lose many of the advantages of style sheets because they mix content with presentation—the inline style is only applied to that individual element. If you need to update the style of your site, you have to update the style properties of every affected page and page element.

By contrast, although style sheets can be more difficult to understand at first, they enable you to make site-wide changes from one convenient location. When you update a style item in your style sheet, it immediately updates the style of every page or page element that uses it.

It's worth taking the time to become familiar with CSS because it:

- Saves you time and effort when building and designing your site
- Produces cleaner, more consistent site designs
- Simplifies navigation for people with accessibility issues (such as site visitors using screen readers)

For more information about using CSS and creating style sheets, go to the World Wide Web Consortium (W3C) at www.w3.org/Style/CSS. There are also many tutorials available on the Internet that provide in-depth CSS training.

About CSS Classes and IDs

When you use style sheets to style your site, you can redefine the formatting of HTML tags such as `body` or `h1`. You can also create CSS classes and IDs to define the style of particular elements, such as headers or repeating content. A CSS class lets you define and apply style properties to many elements on a page, whereas a CSS ID is ideal for targeting a single item on a page. For example, in a page's structure, IDs are often used to define the header and footer areas, as each page has only one header or footer, but classes are used to define repeating page elements, such as a blog post.

Best Practices

- Include a [CSS reset](#) in your style sheet to reset all style items to a baseline value. A CSS reset helps avoid cross-browser differences due to their built-in default style settings.

- Use CSS classes and IDs instead of inline styles wherever possible. CSS classes and IDs promote the separation of presentation and content, and makes it easier to update the site's styles.
- Use IDs when there is only one occurrence per page. When you've used the ID, you can't use it again on that page. Use classes when there are one or more occurrences per page.
- Use [groups](#) to organize your CSS logically and make it easier to maintain your style sheet.
- If you're using CSS3, ensure you preview and test your site in each browser that you want it to support, because some browsers haven't yet fully implemented CSS3 features.

[Use the Style Pane](#)

The Style pane is a visual CSS editor that lets you modify style properties, such as the background color, font size, and border style, as you work with pages and page elements. If you're using CSS classes or IDs to style your pages, you can modify or create style items directly from the Style pane, rather than opening the style sheet.

[Style Pane Properties](#)

The Style pane is a visual CSS editor that lets you modify style properties, such as the background color, font size, and border style, as you work with pages and page elements.

[Understand the Style Sheet View in Site.com](#)

When working with style sheets, you can add style items, organize them into groups, and edit the CSS code directly.

[Create and Use CSS Style Sheets](#)

A default style sheet called "Site Style Sheet" is included with every site you create. However, if you're familiar with CSS and need multiple style sheets, you can create ones to use in your site.

[Create Style Sheet Items and Groups](#)

When adding style items to style sheets, you can define CSS classes and IDs, or you can redefine the formatting of HTML tags such as `body` or `h1`. When you change the CSS style of an HTML tag, anything formatted with that tag is immediately updated.

[Use CSS Reset](#)

Every browser has set presentation defaults, but unfortunately they aren't standardized across all browser types. This means that when you use CSS to style your site, it doesn't render as expected when you view it in different browsers.

See Also

[Use the Style Pane](#)

[Understand the Style Sheet View in Site.com](#)

[Create and Use CSS Style Sheets](#)

Use the Style Pane

The Style pane is a visual CSS editor that lets you modify style properties, such as the background color, font size, and border style, as you work with pages and page elements. If you're using CSS classes or IDs to style your pages, you can modify or create style items directly from the Style pane, rather than opening the style sheet.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To apply a style to a selected page or page item:

1. Open the Style pane ().
2. To apply:
 - An inline style, select **Inline**. Inline styles affect the selected item only and aren't included in a style sheet.
 - A CSS class, select **Class** and start typing the name. If the class exists in your style sheet, select it in the list that appears. To create a class, type the name, select it, and click **Yes** to add it to the style sheet.
 - A CSS ID, select **ID** and select it in the dropdown list. To create an ID, click , enter the ID name, and click .

Menu page elements have several components, which you can [style individually](#) by selecting your preferences in the Style dropdown list that appears.

3. In the Visual tab, apply [style properties](#) as appropriate. Alternatively, in the Code tab, you can type your CSS styles directly and click **Apply**.

 **Tip** To view the style properties associated with a selected page or page element, open the Code tab of the Style pane. To remove the style properties, click **Clear**.

See Also

[Create Style Sheet Items and Groups](#)

[Create and Use CSS Style Sheets](#)

[Cascading Style Sheets Overview](#)

Style Pane Properties

The Style pane is a visual CSS editor that lets you modify style properties, such as the background color, font size, and border style, as you work with pages and page elements.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

The Background Section

Property	Description
Background Color	Sets the element's background color. Click the color box and use the color picker to select a color, or enter a specific hexadecimal code in the text box. You can also choose from a list of colors in the Background Color dropdown list.
Background Image	Adds a background image to the element. Click URL and enter the image URL, or click  to select an imported image.
Background Repeat	Tiles the element's background image. <ul style="list-style-type: none"> ensures that only one copy of the image appears. repeats the image horizontally. repeats the image vertically. tiles the image both horizontally and vertically.
Position	Specifies the position of the element's background image. <ul style="list-style-type: none">To set the horizontal position of the background image, enter a value in the X text box and select a unit of measurement. Alternatively, select Left or Right in the dropdown list.To set the vertical position of the background image, enter a value in the Y text box and select a unit of measurement. Alternatively, select Top or Bottom in the dropdown list.
Cursor	Sets the cursor type, such as crosshair or pointer.

The Font & Color Section

Property	Description
Font	Sets the font family of the selected element.
Color	Sets the font color. Click the color box and use the color picker to select a color, or enter a specific hexadecimal code in the text box. You can also choose from a list of colors in the Color dropdown list.
Size	<p>Sets the font size. Enter a value in the Size text box and select a unit of measurement such as em, point, or %. Alternatively, select a predefined value such as XX-Small. Select Inherit to use the same font size as the parent element (for example, the page or panel).</p> <p>Use relative sizes such as em or a percentage to enable your end users to resize the font size in their Web browsers.</p>
Style	<p>Formats the element's font style.</p> <ul style="list-style-type: none"> • makes the font bold. • makes the font bold and italic. • makes the font italic. • None removes existing styles.
Font Variant	Specifies whether to render the font as small capitals.
Line Height	Modifies the amount of space between lines of text. Enter a value in the text box and select a unit of measurement such as pixels, percentage, or em. Select Inherit to use the same line height as the parent page element.
Text Decoration	<p>Applies decorative effects to the element's text. For example, you could remove the underline that usually appears under hyperlinks, which is a standard CSS rule that's built in to most Web browsers.</p> <ul style="list-style-type: none"> • underlines the text. • applies strikethrough formatting. • displays a line over the text. • None removes existing text decoration.
Align	Aligns the text of the selected element.

Property	Description
	<ul style="list-style-type: none"> aligns the text to the left. aligns the text to the right. centers the text. aligns the text with both the left and right margins.
Case	<p>Changes the capitalization of the element's text.</p> <ul style="list-style-type: none"> capitalizes the first character of each word. capitalizes all characters. lowercases all characters. removes existing capitalization formatting.
Text Indent	<p>Indents the first line of text of the selected page element. Enter a value in the text box and select a unit of measurement. Select Inherit to use the same indentation as the parent page element.</p>
White Space	<p>Controls how white spaces such as spaces, tabs, and hard returns are handled inside an element.</p>

The Layout Section

Property	Description
Positioning	<p>Positions page elements outside the normal flow of the document. Usually, elements on a page are rendered in Web browsers in the order they appear in the document. Block elements such as <code>p</code> tags and <code>div</code> tags appear one beneath the other, whereas inline elements such as <code>em</code>, <code>strong</code>, and <code>span</code> tags are rendered next to text or each other.</p> <ul style="list-style-type: none"> Absolute positions the content using the settings in the Top, Bottom, Left, and Right text boxes. Relative renders the page element in the normal layout flow, but moves the element relative to its normal position depending on the values in the Top, Bottom, Left, and Right text boxes. For example, if you set an element's left position to 20 pixels, the page element is positioned 20 pixels further to the left.

Property	Description
Display	<p>Overrides a page element's default layout behavior. For example, you can hide page elements, make block elements render inline, or make inline elements render as blocks.</p> <ul style="list-style-type: none"> • None hides the page element. • Block displays the page element as a block-level page element, with a line break before and after the element. • Inline, which is the default setting, displays the page element as an inline page element without a line break before or after the element. • Inline-block renders the page element as an inline rectangle, but with content that behaves as if it's inside a block element.
Position	<p>When used in conjunction with the Absolute or Relative positioning options, these four properties place page elements outside the normal flow of the document. Enter a value in the text boxes as appropriate and select a unit of measurement in the respective dropdown lists.</p> <ul style="list-style-type: none"> • Top sets how far the top edge of an element is above or below the top edge of the parent element. • Bottom determines how far the bottom edge of an element is above or below the bottom edge of the parent element. • Right sets how far the top edge of an element is to the right or left of the right edge of the parent element. • Left defines how far the left edge of an element is to the right or left of the left edge of the parent element.
Z-index	<p>Specifies the order in which elements overlap each other when they must be rendered in the same space. An element with a greater z-index value covers an element with a lower value. The default value is 0.</p> <p>Click  and  to increase and decrease the z-index, or enter a value in the text box.</p>
Float	<p>Floats a page element to the left or right so that subsequent elements—text for example—wrap around the floating page element.</p> <ul style="list-style-type: none"> •  floats the page element to the left.

Property	Description
	<ul style="list-style-type: none"> • floats the page element to the right. • None removes an existing float setting.
Clear	<p>Specifies whether the selected page element allows floating page elements beside it.</p> <ul style="list-style-type: none"> • moves the page element below any floating page element on its left. • moves the page element below any floating page element on its right. • moves the page element below floating page elements on either side. • None removes existing float settings.
Visibility	<p>Specifies whether the selected page element is visible.</p> <ul style="list-style-type: none"> • Visible is the default value. • Hidden hides the page element and renders an invisible rectangle in its place. • Collapse is used to hide table elements. (For other page elements, it has the same result as hidden.) <p>Invisible page elements still occupy the same space in the page's layout.</p>
Overflow	<p>Specifies whether the content of a page element is clipped when it overflows its area.</p> <ul style="list-style-type: none"> • Visible doesn't clip the content. • Hidden clips the content. • Scroll clips the content, but provides scroll bars so that users can view the remaining content. • Auto is dependent on the browser, but displays a scroll bar to view the rest of the content.

The Dimensions Section

Property	Description
Width	Sets the width of the selected page element. Enter a value in the Width text box and select a unit of measurement. Select Inherit to use the width of the parent page element.
Height	Sets the height of the selected page element. Enter a value in the Height text box and select a unit of measurement. Select Inherit to use the height of the parent page element.
Margins	Sets the width of the page element's margin, which is the space between its border and outer edge. Set the margins for all four sides by entering a value in the All text box, or add margins to the top, right, bottom, or left sides as required.
Padding	Sets the width of the page element's padding, which is the space between its content and border. Set the padding for all four sides by entering a value in the All text box, or add padding to the top, right, bottom, or left sides as required.

The Borders Section

Property	Description
Type	Specifies whether to set border properties for each side separately or for all four sides.
Style	Sets the border's style such as dashed, dotted, or double.
Color	Sets the border's color. Click the color box and use the color picker to select a color, or enter a specific hexadecimal code in the text box. You can also choose from a list of colors in the Color dropdown list.
Thickness	Specifies the border's thickness. Enter a value in the Thickness text box and select a unit of measurement. Alternatively, select Thin, Medium, or Thick.

The Tables Section

Property	Description
Border Collapse	When designing tables:

Property	Description
	<ul style="list-style-type: none"> • Collapse uses a common border between cells • Separate gives each cell its own border
Horizontal Spacing	Sets the horizontal distance that separates cell borders. Enter a value in the text box and select a unit of measurement. This value is used only if Border Collapse is set to Separate.
Vertical Spacing	Sets the vertical spacing that separates cell borders. Enter a value in the text box and select a unit of measurement. This value is only used if Border Collapse is set to Separate.

See Also

[Use the Style Pane](#)

Understand the Style Sheet View in Site.com

When working with style sheets, you can add style items, organize them into groups, and edit the CSS code directly.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Open a style sheet on the Overview tab by double-clicking it or hovering over it and clicking  | **Edit**. The style sheet opens as a new tab.



- Using the toolbar (1), you can [import a style sheet](#) and edit the style sheet's CSS code directly.

- Using the style sheet pane (2), you can:
 - [Create style items and groups](#)
 - Preview, edit, and delete style items
 - Move style items and groups by dragging them to the correct location
 - [Add a CSS reset](#)
- Using the Style Preview section (3), you can preview and manually edit a selected style item.
 **Note** You can't preview at-rules, such as `@media` or `@font-face`, in the Style Preview section.
- Using the visual CSS editor (4), you can define the [CSS properties](#) for the selected style item.

See Also

[Create and Use CSS Style Sheets](#)

[Cascading Style Sheets Overview](#)

Create and Use CSS Style Sheets

A default style sheet called “Site Style Sheet” is included with every site you create. However, if you’re familiar with CSS and need multiple style sheets, you can create ones to use in your site.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To create a style sheet:



- Style sheet names can only contain alphanumeric characters, hyphens, colons, and underscores.
- You can also [import a CSS file](#) to use in your site.

1. Click **Style Sheets | New** on the Overview tab. Alternatively, click **New Style Sheet** in the Style Sheets

view.

2. Enter a name for the style sheet.
3. Click **Apply**. The style sheet opens.
4. [Add style items and groups](#) to the style sheet.

After you create a style sheet, you must attach it to a page to apply its styles to the page.

To attach a style sheet to a page:

 **Tip** If you used a [page template](#) to create your site pages, the quickest way to include the new style sheet on every page is to attach it to the template, which automatically includes a reference to the style sheet in every page that's based on the template.

1. Select the page in the Page Structure pane ().
2. In the Style Sheets section of the Properties pane () click .
3. Select the style sheet in the list that appears.
4. Attach the style sheet to the page by clicking  beside the dropdown list.

See Also

[Understand the Style Sheet View in Site.com](#)

[Cascading Style Sheets Overview](#)

Create Style Sheet Items and Groups

When adding style items to style sheets, you can define CSS classes and IDs, or you can redefine the formatting of HTML tags such as `body` or `h1`. When you change the CSS style of an HTML tag, anything formatted with that tag is immediately updated.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Creating Style Items

To open a style sheet, double-click it in the Style Sheets view of the Overview tab, or hover over it and click  | **Edit**.

If you're very familiar with CSS and prefer coding by hand, click **Edit Style Sheet Code** to edit the style sheet directly using the CSS editor. Additionally, to add at-rules (for example, `@media`), you must edit the style sheet directly.

Alternatively:

1. Select the style sheet and click  | **Insert Style Item**.
2. Enter the name of the style item:
 - To redefine the default formatting of a specific HTML tag, enter the HTML tag name—for example, `body` or `h1`.
 - To create a CSS class, enter the class name and ensure that you include a period before it—for example, `.classname`.
 - To create a CSS ID, enter the ID name preceded by `#`—for example, `#contentID`.
3. Click **Apply**.
4. Add style definitions by either:
 - Setting **style properties** in the visual style editor on the right
 - Typing CSS styles in the text box in the Style Preview section and clicking **Save**

As you modify the definition of a selected style item, you can see how your changes appear in the Style Preview section.

Tip

- A class name must begin with a period or it isn't recognized as a CSS class.
- An ID name must begin with `#` or it isn't recognized as a CSS ID.
- Use IDs when there is only one occurrence per page. When you've used the ID, you can't use it again on that page. Use classes when there are one or more occurrences per page.
- Class and ID names can contain alphanumeric characters, hyphens, and underscores only, and can't begin with a number or include spaces.

Creating Style Groups

Use groups to organize your CSS logically, which makes it easier to locate and maintain styles.

When the style sheet is open:

1. Select the style sheet and click  | **Insert Style Group**.
2. Enter a name for the group and click **Apply**.
3. To add a style to the group, select the group and click  | **Insert Style Item**. To add an existing style to the group, drag it onto the folder icon.

Assigning Style Items

After you've created styles, you can assign them to the pages and pages elements of your site.

To assign a class to a page or page element, select it and either:

- Type the class name in the **Class** field of the Properties pane ().
- Select **Class** in the Style pane (), start typing the name, and select it in the list that appears.

To assign an ID to a page or page element, either:

- Type the ID name in the **ID** field in the Properties pane ().
- Select **ID** in the Style pane () and select it in the dropdown list.

See Also

- [Create and Use CSS Style Sheets](#)
- [Understand the Style Sheet View in Site.com](#)
- [Cascading Style Sheets Overview](#)

Use CSS Reset

Every browser has set presentation defaults, but unfortunately they aren't standardized across all browser types. This means that when you use CSS to style your site, it doesn't render as expected when you view it in different browsers.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

For example, browsers differ in how they display:

- Unordered and ordered lists

- Top and bottom margins for headings
- Indentation distances
- Default line-heights

A CSS reset cancels the differences between browsers to control how browser elements are presented to the end user. You can either use Site.com's CSS reset, or you can add your own CSS reset code.

To use Site.com's CSS reset:

1. In the Style Sheets view on the Overview tab, open the style sheet by double-clicking it, or hovering over it and clicking  | **Edit**.
2. Click  | **Insert CSS Reset**.
3. Ensure the CSS reset is positioned at the top of the style sheet. To move it, drag it to the correct location in the pane on the left.

To add your own CSS reset code:

1. In the Style Sheets view on the Overview tab, open the style sheet by double-clicking it, or hovering over it and clicking  | **Edit**.
2. Click **Edit Style Sheet Code** to open the CSS editor.
3. Paste the code at the top of the style sheet code.
4. Click **Save and Close**.

See Also

[Create and Use CSS Style Sheets](#)

[Cascading Style Sheets Overview](#)

Site Branding Overview

Branding provides a flexible way for you to define different aspects of your website's brand. Once branding properties are defined, your editors can easily customize everything in one centralized place, the Branding Editor. When your website editors customize the properties, they get a preview of their branding changes immediately.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

By adding branding properties to your website, page templates, or widgets, you can easily define and change those aspects of your website's brand without needing to edit your Cascading Style Sheet.

Here are examples of aspects that you might define that are related to your website's brand:

- The website's background, page, or link colors
- Font size and color
- The logo or header background image
- The thickness and color of lines

To use branding in your website, you must:

1. Define the parts of your site that make up your brand by creating a branding property for each aspect such as background color, logo, and fonts. You complete this task from [Branding Properties](#) within Site Configuration.
2. Use [expression language](#) syntax to replace the current definitions for these properties within your Cascading Style Sheets with the new branding properties.
3. Use the [Branding Editor](#) to customize the properties and preview how your website will look.

The branding properties can also be accessed by using expression language syntax directly from within a custom code block or content block.

[Creating Branding Properties](#)

With branding , you can quickly create style properties that can be reused by editors of your site.

[Set Up Branding Properties](#)

After you create branding properties, the next step is to reference them as expressions in your Cascading Style Sheets (CSS) or Site.com page elements such as content blocks or custom code elements.

[Use the Branding Editor](#)

The Branding Editor provides a centralized place to customize and then preview the changes that you make to the branding properties in your site.

See Also

[Creating Branding Properties](#)

[Set Up Branding Properties](#)

[Use the Branding Editor](#)

Creating Branding Properties

With branding , you can quickly create style properties that can be reused by editors of your site.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

Site.com Contributor User

AND

Contributor role assigned at the site level

When creating your branding properties, you can organize properties into multiple sections to make them easier to find in the Branding Editor. You can also organize your properties within sections.

- Order properties within sections in a logical manner. For example, organize them alphabetically.
- Order sections and properties by dragging and dropping them within the Branding Properties view.

1. From within your site, click **Site Configuration | Branding Properties**.

The Branding Properties editor appears.

2. Click .

3. Enter a name for the property in the Label field.

The expression name is filled in automatically. The expression name is used in style sheets and code blocks.

4. Choose a type.

5. Set the default value.

6. To make the property required, click **Required**.

Hover over any property and use the menu  to edit or delete it. You can double-click any section name to edit it.

See Also

[Site Branding Overview](#)

[Custom Property Types](#)

[Set Up Branding Properties](#)

[Use the Branding Editor](#)

Set Up Branding Properties

After you create branding properties, the next step is to reference them as expressions in your Cascading Style Sheets (CSS) or Site.com page elements such as content blocks or custom code elements.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

Site.com Contributor User

AND

Contributor role assigned at the site level

The expression syntax for branding properties is `{!Site.branding-property-name}`. For example, if you create a color property with the name HeaderColor, you can insert the property into a CSS or code block by typing `{!Site.HeaderColor}`.

 **Note** The expression names are case-sensitive!

1. Open your CSS or code block.
2. Locate the place where you want to insert the expression.
3. Type `{!`
- and a dropdown list of available properties appears.
4. Select the property and double-click to enter it on the page.

 **Example** For example, if part of your website's brand is to use blue for the background of the header section, you normally statically set the color's HEX value in your CSS rules.

```
.site-header {  
    background-color: #3793DD;  
}
```

With branding, you can create a property that maps directly to the HEX color and then use the property's expression name in your CSS.

```
.site-header {  
    background-color: {!Site.HeaderColor};  
}
```

Now the color can be easily changed by any of your site editors by using the Branding Editor instead of editing the CSS.

See Also

- [Site Branding Overview](#)
- [Creating Branding Properties](#)
- [Use the Branding Editor](#)

Use the Branding Editor

The Branding Editor provides a centralized place to customize and then preview the changes that you make to the branding properties in your site.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

Site.com Contributor User

AND

Contributor role assigned at the site level

The Branding Editor includes two areas:

- The editor palette, on the left side, contains all of the branding properties that you can edit.
- The preview area on the right side shows a live preview of how your website will appear after you

change a style.

1. Open the page that you want to edit.
2. Click .
3. From the left pane, select the style property that you want to edit.
Your results appear immediately in the right pane.

See Also

- [Site Branding Overview](#)
- [Creating Branding Properties](#)
- [Set Up Branding Properties](#)

Custom Site Properties Overview

With custom site properties, you can define and store frequently occurring content on your site. For example, you can store your address and phone number as a custom property so that it can be reused by anyone who is editing your site. You can apply stored properties to pages, headers and footers, and widgets quickly by using expression language syntax.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

To use custom properties in your website, you must:

1. Define the custom property. You complete this task from [Custom Properties](#) within Site Configuration.
2. Use [expression language](#) syntax to replace the current definitions for these properties within your Cascading Style Sheets or custom code blocks.

After you create a new property and specify its value and type, you can access the property in custom code or content blocks by using expressions. Expressions are placeholders for data that will be replaced with the custom property when the page loads.

[Create Custom Site Properties](#)

You can quickly create Custom Properties that can be reused by editors of your site.

[Setting Up Custom Properties](#)

You can use custom properties in your website by using expression language syntax within a style sheet, code block, widget, or the HTML page editor.

[Custom Properties for Page Templates or Widgets Overview](#)

When you create a page template or a widget, you can add custom properties to it and specify the value and type of each property to achieve greater flexibility over how templates and widgets are

reused.

Add Custom Properties to Page Templates or Widgets

When you create a page template or a widget, you can add custom properties to it and specify the value and type of each property.

Custom Property Types

Property types are available when designers or site administrators create site branding properties, site custom properties, widget branding properties, and widget custom properties.

See Also

[Create Custom Site Properties](#)

[Setting Up Custom Properties](#)

Create Custom Site Properties

You can quickly create Custom Properties that can be reused by editors of your site.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

Site.com Contributor User

AND

Contributor role assigned at the site level

When creating your custom properties, you'll probably want to organize them logically.

- Organize properties into multiple sections to make them easier to manage and easy to find.
- Order properties within sections in a logical manner. For example, organize them alphabetically.
- Order sections and properties by dragging and dropping them within the Custom Properties view.

1. From within your site, click **Site Configuration | Custom Properties**.
The Custom Properties view appears.
2. Click .
3. Enter a name for the property in the Label field.
The expression name is filled in automatically. The expression name is used in style sheets and code blocks.
4. Choose a type.
5. Set the default value.
6. To make the property required, click **Required**.

To edit or delete properties, hover over any property and use the menu   . To edit section names, double-click the name make changes.

See Also

- [Custom Site Properties Overview](#)
- [Custom Property Types](#)
- [Setting Up Custom Properties](#)

Setting Up Custom Properties

You can use custom properties in your website by using expression language syntax within a style sheet, code block, widget, or the HTML page editor.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

Site.com Contributor User

AND

USER PERMISSIONS NEEDED

Contributor role assigned at the site level

To reference a site custom property in custom code or a content block, use the syntax `{!Site.expression_name}`. For example, create a custom property to store the company phone number with `PhoneNumber` as the expression name. In a content block, enter `Contact us at { !Site.PhoneNumber }`. When the page loads, it replaces the value that is represented by `{!Site.PhoneNumber}` and displays “Contact us at 1-800-667-6389” on the page.

 **Note** The expression names are case-sensitive!

1. Open a website element such as a code block or widget.
2. Locate the place where you want to insert the expression.
3. Type `{!` to see the list of available custom properties.
4. Select the property and double-click to enter it on the page.

 **Example** Setting up custom properties is great for things that might change over time such as your address or a particular product phrase. Because expressions are just placeholders, if you update the value of a property in the Custom Properties view, the value is updated automatically wherever the custom property is referenced on a page.

See Also

[Custom Site Properties Overview](#)

[Create Custom Site Properties](#)

Custom Properties for Page Templates or Widgets Overview

When you create a page template or a widget, you can add custom properties to it and specify the value and type of each property to achieve greater flexibility over how templates and widgets are reused.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Then, by adding custom code or content blocks to the page template or the widget, you can access the property values by using expressions. Expressions serve as placeholders for data that's replaced with information when the page loads.

In turn, when you or your team create a page from the template, the page is a copy or *instance* of the

template. Similarly, when you add the widget to a page, it creates an instance of the widget. You can't edit the instance, but you can update its property values.

Because expressions are just placeholders, their values are updated automatically when you update the values in the Properties pane of the page or widget.

You can also create sections to group related properties. These sections control how properties are grouped in the Properties pane.

 **Example** For example, let's say you add a content block to a template to contain the page's heading. In this case, when users create a page from the template, you want to let them replace part of the text to suit their needs, but without letting them edit the entire content block. By adding a custom property that's called **pageSubject** and specifying an initial value, you can instead use the following expression in the content block: `Learn About { !pageSubject }` This action lets team members rename any page that's derived from the template by updating the **Page Subject** property in the page's Properties pane, which automatically updates the value that's represented by the `{ !pageSubject }` expression. 

 **Example** Alternatively, let's say you want to create a YouTube widget using the following embed code:

```
<iframe width="560" height="315" src="//www.youtube.com/embed/hcUaN6XBTz4" frameborder="0" allowfullscreen></iframe>
```

However, you want users to specify which video to display when they add an instance of the widget to the page. In this case, you could create a section called **YouTube**, add a custom property labeled **Video URL** with `videoURL` as the expression name, and instead use the following code:

```
<iframe width="560" height="315" src="{ !videoURL }" frameborder="0" allowfullscreen></iframe>
```

Now, when users add the YouTube widget to the page, they can point to any video by updating the **Video URL** property in the YouTube section of the Properties pane, which automatically updates the value represented by the `{ !videoURL }` expression. 

See Also

- [Widgets Overview](#)
- [Add Custom Properties to Page Templates or Widgets](#)
- [Custom Property Types](#)
- [About Displaying Dynamic Data Using Expressions](#)

Add Custom Properties to Page Templates or Widgets

When you create a page template or a widget, you can add custom properties to it and specify the value and type of each property.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Then, by adding custom code or content blocks to the page template or widget, you can access the property values using expressions, which serve as placeholders for the values. When you or your team create a page from the template or add an instance of the widget to the page, you can update the property values to modify the instance.

When the page template or widget is open:

1. Make sure the template or widget is selected in the Page Structure pane and click **Edit Custom Properties** in the Properties pane.
2. Click **New Property**.
Base is the default section, which you can rename. Sections control how properties are grouped in the Properties pane of a widget instance or page.
3. Enter the property label, which appears in the Properties pane of a widget instance or page—for example, *Video URL*.
4. Optionally, update the expression name that was added automatically.
Expression names are case sensitive and can't include spaces.
5. Specify the property type and value.
6. Optionally, mark the property as required.
7. Add additional properties or sections as needed. Reorder items by dragging them to a new location.
8. To use a custom property, drag a **Custom Code** or a **Content Block** page element from the Page Element tab onto the page template or widget.
9. Use the syntax **{!expression_name}** to reference the expression—for example, `{!videoURL}`.
When the page loads, the expression is replaced by the property value.
10. Add any additional text you require and save your changes.
For example, `Check out our video at {!videoURL}`.

 **Tip** You can also create a custom property by first typing the name of the property in custom code or

a content block using expression language. For example, if you type `{!videoHeight}` in a content block, a new property called **videoHeight** is automatically added to the Properties pane of the template or widget, where you can then add a property value.

If you update a widget's properties, your changes aren't reflected in any widget instances. Instead, you must replace existing widget instances with the latest version.

If you delete a custom property that's being used in custom code or a content block, make sure you remove any references to the property.

See Also

[Custom Properties for Page Templates or Widgets Overview](#)

[Custom Property Types](#)

[About Displaying Dynamic Data Using Expressions](#)

[Create Widgets](#)

Custom Property Types

Property types are available when designers or site administrators create site branding properties, site custom properties, widget branding properties, and widget custom properties.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Property Type	Description
Text	Lets users enter a text value.
Checkbox	Lets users set a true (selected) or false (deselected) value.
Color	Lets users specify a color, either by selecting one from the color picker or by entering a hexadecimal value, such as #333333.
Image	Lets users specify an image, either by browsing to a site image, uploading a new image, or entering the URL to an image.
HTML	Lets users enter an HTML value.
Units	Lets users enter a measurement, such as pixels, percentage, or em—for example, 5px.

Property Type	Description
Picklist	<p>Lets users select a value from a list you define.</p> <p>To add items to the picklist, create a comma-delimited list of options, where each option has a label and a value—for example, <i>Label 1:Value 1, Label 2:Value 2</i>. If you don't add a value for an option, the label is used as the value by default.</p> <p>To define a default picklist selection, use the format <i>label:value:default</i>—for example, <i>Small:S:default, Large:L</i>.</p>
Font	Lets users select a font.

See Also

[Add Custom Properties to Page Templates or Widgets](#)

[Create Custom Site Properties](#)

[Creating Branding Properties](#)

Site.com Data Services Overview

Site.com data services combine many features that let you connect to standard and custom Salesforce objects. Retrieve data from your organization's objects and dynamically display it on your site pages, or alternatively, gather and submit data from your customers. And when you update data in your Salesforce object, the changes are reflected automatically on the live site—no site updates required!

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Here are a few ways you can use Site.com data services:

- Publish a catalog of products—List your company's products and include information, such as model numbers and prices, pulled dynamically from your organization.
- Post company press releases—Publish your company's press releases and sort by publication date.
- Create a realtor website—Display current listings filtered by city or price.
- Create a recruiting website—Post job openings to a public site and allow visitors to submit applications and resumes.

So how does it all work? Several data-bound page elements let you retrieve and display your data, or collect data from your site visitors.

[Data tables](#) connect to Salesforce objects, retrieve a dataset based on the filter criteria that you specify, and display one or more record as rows in the table.

[Data Repeaters](#) and [data elements](#) combine to let you connect to standard and custom objects, retrieve data, and dynamically display it on your site's pages. Together, the data repeater and data elements result in a "repeating template" that offers you the greatest flexibility for displaying one or more records on the page.

[Data functions](#) let you perform calculations on data retrieved from objects and display the result on the page. For example, for a particular field in an object, you can use a data function to calculate the total value or the average amount of all returned records.

[Nested repeaters](#) let you retrieve data from objects with a parent-to-child relationship.

[Forms](#) and [form fields](#) combine to let you collect data from your site visitors and submit the data to standard or custom Salesforce objects. Create web-to-lead forms, capture customer details, or gather feedback on your products or services.

Data Services Considerations

- To allow guest users to view the data in or submit data to a Salesforce object, you must [set the object's data access permissions](#).
- When working with assets, the easiest way to take advantage of Site.com data services is to import the files into your website, and store a relative URL to these assets in your standard or custom object. See [Storing Assets to Use with Salesforce Objects](#).
- If you add a data-bound page element to your site and then subsequently change a field type in the Salesforce object it's connected to—for example, changing a text field to a picklist—the data-bound page element no longer works. You must reconfigure the data-bound page element to reference the updated field.
- If you update data in an object that's connected to a data table, data repeater, or data function, the changes are reflected automatically on the live site. To control this, you can add a picklist field to the object to specify when a record is approved to go live. Then you can use the field to [filter the records](#) by approved status, so only approved records appear on the live site.

[Setting Data Access Permissions for Salesforce Objects](#)

Sites built with Site.com are publicly available, so visitors access the site via the Guest User license that's associated with the site.

[Storing Assets to Use with Salesforce Objects](#)

Because websites built with Site.com are publicly available, site visitors don't have the security privileges required to view images and documents stored in your Salesforce objects, which are available to authenticated users only.

[Dynamically Retrieve Data with Data Repeaters](#)

Use a data repeater to connect to a standard or custom Salesforce object and retrieve a dataset based on the filter criteria that you specify.

[Adding a Form to the Page](#)

Use forms to collect data from your site visitors and submit the data to standard or custom Salesforce objects. Create web-to-lead forms, capture customer details, or gather feedback on your products or services.

The Default, Error, and No Data Views

When working with data repeaters, data tables, data functions, and forms, you can customize what your site visitors see if an error occurs when connecting to the data source.

Repairing Data Connections

If you or another user modifies the object that an existing data repeater, data table, data function, or form is connected to, the data connection might break. For example, this can happen if a connected object or field is renamed or deleted, or if its permissions are changed.

See Also

[Access Data in Related Objects Overview](#)

[Add Pagination to Data Repeaters and Data Tables](#)

[The Default, Error, and No Data Views](#)

Setting Data Access Permissions for Salesforce Objects

Sites built with Site.com are publicly available, so visitors access the site via the Guest User license that's associated with the site.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To edit the guest user profile:

Site.com Publisher User field enabled on the user detail page

AND

USER PERMISSIONS NEEDED

Site administrator or designer role assigned at the site level

AND

Manage Profiles and Permission Sets

AND

Customize Application

By default, site visitors can access information made available in an active public site, such as the site's pages and assets. However, to allow guest users to view or submit data to a Salesforce object, you must modify the object's permission in the site's guest user profile. Each site has a separate Guest User license, so you can control guest access to Salesforce objects on a per-site basis.

To edit the site's guest user profile:

1. On the Overview tab of Site.com Studio, click **Site Configuration** and click **Site Name Profile**. Alternatively, if you're adding a data repeater, data table, data function, or form to the page, click **go to the guest user profile** in the item's dialog box.
2. In the site's guest user profile, enable the "Read" permission on the standard or custom objects you want to retrieve data from using data repeaters, data tables, or data functions. Enable the "Create" permission on the objects you want to submit data to using forms. All permissions that aren't set by default must be set manually.
3. If required, modify the field-level security of an object.

See Also

[Storing Assets to Use with Salesforce Objects](#)

[Site.com Data Services Overview](#)

Storing Assets to Use with Salesforce Objects

Because websites built with Site.com are publicly available, site visitors don't have the security privileges required to view images and documents stored in your Salesforce objects, which are available to authenticated users only.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Therefore, the easiest way to take advantage of Site.com data services is to import the files into Site.com, and instead store a relative URL to these assets in your standard or custom object. Alternatively, if your images or files are hosted elsewhere and readily available on the Internet, you can use an absolute URL.

For example, let's say you want to use a custom object called "Catalog" to build a Catalog site page that displays product details and images. Before you begin building the Catalog site page, you would:

1. Import the product images into your Site.com site.
2. In the Catalog object, create a field to store the relative URL of the image, such as **Image URL**.
3. For each product record, add the relative path of the image. This URL is relative to the site, so if you upload `widget.png` to your site, the relative path is `/widget.png`. URLs are case sensitive.

Then, when you add a data table, or a data repeater and data elements to the Catalog site page to display the product data, you can reference the **Image URL** field to dynamically display each product's image on the page.

See Also

[Setting Data Access Permissions for Salesforce Objects](#)

[Site.com Data Services Overview](#)

Dynamically Retrieve Data with Data Repeaters

Use a data repeater to connect to a standard or custom Salesforce object and retrieve a dataset based on the filter criteria that you specify.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To edit the guest user profile:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

AND

Manage Profiles and Permission Sets

AND

Customize Application

You can connect to standard or custom Salesforce objects, or to content lists or categories in your site. When you combine a data repeater with data elements, custom code, or content blocks, you can create a “repeating template” that displays one or more records on the page.

To add a data repeater to a page:

1. Drag a **Data Repeater** from the Page Elements pane onto the page.
2. Select the object that you want to connect to.



Note

- For Site.com users, the dropdown list only displays objects that are available to guest users because site visitors access your public site via the Guest User license. To make other objects available, go to the [guest user profile](#), enable the relevant object's “Read” permission, and refresh the list.
- For Communities users, the dropdown list displays objects that aren't always available to site visitors. For authenticated visitors, object access on public and private pages is controlled by their user profiles. For unauthenticated visitors, object access on public pages is controlled by the site's guest user profile.

3. Optionally, in Filters, select criteria to filter your dataset. If you don't select any criteria, all the data from the item is returned.
 - a. Select the field to which the filter criteria apply. The Field dropdown list displays the object's fields,

- followed by the fields of all parent objects, which use the format `parent_object_name.field_name`.
- b. Select [the operator](#) to control how results are filtered. For example, select `Equals` to return an exact match.
 - c. Select [the source](#) of the filter value. For example, to specify an explicit value, select `Fixed value`, or to use the values passed to the page via a query string, select `URL query string`.
 - d. Set the value of the filter. If you're using a query string, you can also specify what happens if the query string is missing.
 - e. Add additional filter criteria as required to narrow your results further. Each filter item is combined with an AND operator.



Note If you're using a fixed value to filter the results, you can view the returned records in the Connection Preview section. To refresh the list of records, click **Reload Preview**.

- 4. In **Sorting**, you can specify whether to sort the results by one or more fields in ascending or descending order. For example, if you're working with an object that contains user data, you could sort your results by gender first and then by name.
- 5. In **Limits**, you can limit the number of returned results. For example, if you're only interested in the top five results, enter `5` in the **Limit results to** field.
- 6. If you're adding [pagination](#), specify the number of results to display per page in the **Results per page** field.
- 7. Click **Save**.

Next, you must add either [data elements](#), [custom code](#), or [content blocks](#) to the data repeater to display the data it retrieves.

[Display Data or Content Using Data Elements](#)

You can use a data element to display the data retrieved by a page data connection or a data repeater. The data element binds to a field in the object and acts as a placeholder that's replaced with the field's data when the page loads.

[Displaying Data or Content Using Custom Code](#)

In addition to data elements and content blocks, you can also use custom code as an alternative way to display data in a data repeater or in a page data connection. It's particularly useful for displaying field data that's inline with text.

[Displaying Data Using Content Blocks](#)

In addition to data elements and custom code, you can also use content blocks as an alternative way to display data in a data repeater or in a page data connection. It's particularly useful for displaying field data that's inline with text.

[Dynamically Retrieve Data with Data Tables](#)

Use a data table to connect to a standard or custom Salesforce object, retrieve a dataset based on the filter criteria that you specify, and display one or more record as rows in the table.

[Edit Columns in a Data Table](#)

A data table's columns bind to the fields of the object it's connected to. Each column cell acts as a placeholder that is replaced with the field's data when the page loads.

[Add Pagination to Data Repeaters and Data Tables](#)

Events let you add interactive and animated effects to the pages and page elements of your website.

When using data repeaters and data tables, you can add pagination events so users can easily page through the displayed data. Pagination events are particularly useful when working with large amounts of data.

[Using Data Functions](#)

A data function lets you connect to a standard or custom Salesforce object, perform calculations on the returned results, and display the calculation on the page. For example, for a particular field in an object, you can use a data function to calculate the total value or the average amount of all returned records.

[Page Data Connections Overview](#)

Page data connections make it easy for site administrators and designers to create a detail page for a single record when working with Salesforce objects.

[Retrieving Data with Page Data Connections](#)

Use a page data connection to create a detail page for a single record when working with Salesforce objects.

[Access Data in Related Objects Overview](#)

Standard and custom objects have relationships that define how records in one object relate to records in another. For example, the Accounts object has a one-to-many relationship with the Contacts object—that is, each account can have one or more contacts associated with it. This relationship is also known as a parent-to-child or a master-detail relationship.

[Displaying Data from Related Objects Using Nested Data Repeaters](#)

You can retrieve data from any child object of a parent object using a data repeater that contains another data repeater, data table, or data function.

[Improve Performance Using Caching](#)

When working with data-bound page elements, such as data repeaters, data tables, and data functions, you can improve the performance and page rendering of your website using caching. Caching controls how often a page containing a data connection requests data from Salesforce.

[Data Filters](#)

When you add a data repeater, a data table, or a data function to a page, you don't have to limit the records it retrieves. However, if you're working with a Salesforce object that has thousands of records, you can limit the returned results using filter criteria.

[About Displaying Dynamic Data Using Expressions](#)

Site.com uses expression language to display data dynamically. Expressions serve as placeholders for data that is replaced with information when the page loads. When working with data-bound page elements or custom widget properties, you can use expressions to customize how data is displayed on the page.

[Data Filter Examples](#)

When working with data repeaters, data tables, and data functions, you can filter the data you retrieve in many ways. In this topic, we explore two options—fixed values and URL query strings—to illustrate some common filtering techniques.

See Also

- [The Default, Error, and No Data Views](#)
- [Improve Performance Using Caching](#)

[Data Filter Examples](#)
[Site.com Data Services Overview](#)

Display Data or Content Using Data Elements

You can use a data element to display the data retrieved by a page data connection or a data repeater. The data element binds to a field in the object and acts as a placeholder that's replaced with the field's data when the page loads.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

When combined with a data repeater, data elements result in a “repeating template” that displays one or more records on the page. When used with a page data connection, data elements display data from a single record.

You can use data elements to display plain text, formatted text (for dates and numbers), or images. You can also add hyperlinks to data elements to allow site visitors to navigate to another page, such as a detailed description, or to refresh the data displayed on the page or the data repeater based on their selection. See [Data Filtering Examples](#).

When the page is open:

1. Drag a **Data Element** from the Page Elements pane onto the data repeater. Alternatively, if the page has a page data connection, drag the **Data Element** page element directly onto the page canvas.
2. Select the field to display. To [customize how the field's data is displayed](#), click **Customize**.
 **Note** The object's fields are listed first, followed by the fields of all [parent objects](#), which use the format `parent_object_name.field_name`.
3. Select the display type.

Option	Description
Text	Lets you display the field's data as plain text.
Formatted text	If you're working with dates, times, or currency, lets you choose from several text display formats.
Image	<p>If the field contains an image URL, lets you display the field's data as an image. The URL can be absolute or relative to the site.</p> <p>You can also select a field to use for the alternative text or enter custom text.</p>

4. To create a hyperlink, select **Add a hyperlink**. Otherwise, go to step 8.
5. Select the link type.

Option	Description
A URL	<p>Lets you link to a Web page by:</p> <ul style="list-style-type: none"> • Choosing a field that you want to reference, such as a field that stores the relative URLs of PDFs you uploaded to your site. • Choosing a field that you want to reference and clicking Customize to add an absolute URL or to create a custom link, such as a URL query string.
An item in your site	<p>Lets you link to a page, image, or file in the site by selecting the item type and then selecting the item. (If you can't see the list of items, place your cursor in the URL field and press the DOWN key on your keyboard.)</p> <p>You can also customize the URL—for example, by creating a URL query string.</p>
An email	<p>Lets you link to an email message by entering the recipient's address, and the message subject and body.</p> <p>You can use merge fields to access the object's fields. For example, if an object has an Email field, enter the merge field, such as {!email}, in</p>

Option	Description
	the Email address text box.
	When the link is clicked, it opens a new message window in the user's email client and adds the appropriate email address to the To: field.

6. Optionally, enter a tooltip by selecting the required field or clicking Customize to add custom text. The tooltip displays as a pop-up when the user hovers over the link.
7. If you're linking to a URL or an item in your site, specify where the item must open.

Option	Description
Popup window	Loads the item into a window. When you select this option, you can set the title for the popup and control its appearance and size with the options that appear.
New window (_blank)	Loads the item into a new, unnamed browser window.
Same window (_self)	Loads the item into the same frame or window as the link. This setting is the default setting.
Topmost window (_top)	Loads the item into the topmost parent frameset or window of the frame that contains the link.
Parent window (_parent)	Loads the item into the parent frameset or window of the frame that contains the link.

8. Click **Save**.

The data element is displayed on the page as a merge field. To test the output, [preview the page](#).

See Also

- [Dynamically Retrieve Data with Data Repeaters](#)
- [Retrieving Data with Page Data Connections](#)
- [Displaying Data or Content Using Custom Code](#)
- [Displaying Data Using Content Blocks](#)

Displaying Data or Content Using Custom Code

In addition to data elements and content blocks, you can also use custom code as an alternative way to display data in a data repeater or in a page data connection. It's particularly useful for displaying field data that's inline with text.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

When the page is open:

1. Drag a **Custom Code** page element from the Page Elements pane onto the data repeater. Alternatively, if the page has a page data connection, drag the **Custom Code** page element directly onto the page canvas.
2. To access the fields of the object that the data repeater or page is connected to, type `{ !` and double-click the **expression** that you want to display.



Note The object's fields are listed first, followed by the fields of all **parent objects**, which use the format `parent_object_name.field_name`.

3. Add any additional expressions or text you require. For example:

To contact `{ !Name }`, call `{ !Phone }`.

where `{ !Name }` and `{ !Phone }` are placeholders for the values of the **Name** and **Phone** fields of each record.

4. Click **Save and Close**.

See Also

[Dynamically Retrieve Data with Data Repeaters](#)

[Retrieving Data with Page Data Connections](#)

[Displaying Data Using Content Blocks](#)

[Data Filter Examples](#)

Displaying Data Using Content Blocks

In addition to data elements and custom code, you can also use content blocks as an alternative way to

display data in a data repeater or in a page data connection. It's particularly useful for displaying field data that's inline with text.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

When the page is open:

1. Drag a **Content Block** page element from the Page Elements pane onto the data repeater. Alternatively, if the page has a page data connection, drag the **Content Block** page element directly onto the page canvas.
2. Enter the name of the field you want to display using an [expression](#). For example, `{!Name}`.

 **Note** The object's fields are listed first, followed by the fields of all [parent objects](#), which use the format `parent_object_name.field_name`.

3. Add any additional expressions or text you require. For example:

`To contact {!Name}, call {!Phone}.`

where `{!Name}` and `{!Phone}` are placeholders for the values of the **Name** and **Phone** fields of each record.

You can also use expressions if you're [adding a hyperlink](#) to the content block.

4. Click **Save**.

See Also

- [Dynamically Retrieve Data with Data Repeaters](#)
- [Retrieving Data with Page Data Connections](#)
- [Displaying Data or Content Using Custom Code](#)
- [Display Data or Content Using Data Elements](#)
- [Data Filter Examples](#)

Dynamically Retrieve Data with Data Tables

Use a data table to connect to a standard or custom Salesforce object, retrieve a dataset based on the filter criteria that you specify, and display one or more record as rows in the table.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To edit the guest user profile:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

AND

Manage Profiles and Permission Sets

AND

Customize Application

A data table's columns bind to the fields of the object it's connected to. Each column cell acts as a placeholder that is replaced with the field's data when the page loads.

To add a data table to the page:

1. Drag a **Data Table** from the Page Elements pane onto the page.
2. Select the object that you want to connect to.

**Note**

- For Site.com users, the dropdown list only displays objects that are available to guest users because site visitors access your public site via the Guest User license. To make other objects available, go to the [guest user profile](#), enable the relevant object's "Read" permission, and refresh the list.
- For Communities users, the dropdown list displays objects that aren't available to site visitors. For authenticated visitors, object access on public and private pages is controlled by their user profiles. For unauthenticated visitors, object access on public pages is controlled by the site's guest user profile.

3. Optionally, in Filters, select criteria to filter your dataset. If you don't select any criteria, all the data from the item is returned.
 - a. Select the field to which the filter criteria apply. The Field dropdown list displays the object's fields, followed by the fields of all parent objects, which use the format *parent_object_name.field_name*.
 - b. Select [the operator](#) to control how results are filtered. For example, select *Equals* to return an exact match.
 - c. Select [the source](#) of the filter value. For example, to specify an explicit value, select *Fixed value*, or to use the values passed to the page via a query string, select *URL query string*.
 - d. Set the value of the filter. If you're using a query string, you can also specify what happens if the query string is missing.
 - e. Add additional filter criteria as required to narrow your results further. Each filter item is combined with an AND operator.
4. In Sorting, you can specify whether to sort the results by one or more fields in ascending or descending order. For example, if you're working with an object that contains user data, you could sort your results by gender first and then by name.
5. In Limits, you can limit the number of results returned. For example, if you're only interested in the top five results, enter 5 in the **Limit results to** field.
6. If you're adding [pagination](#), specify the number of results to display per page in the **Results per page** field.
7. Click **Next**.
8. Add available fields to the table by double-clicking a field, or selecting it and clicking
9. Reorder the list of selected fields by clicking **Move Up** or **Move Down**.
10. Click **Save**.



Note You can't add page elements to a data table. However, you can add additional columns to a data table by selecting it and clicking **Edit**. On the Select Fields screen, select the additional fields and save your changes.

After you've added the data table to the page, you can use the Properties pane to:

- Provide a short heading or summary of the table's purpose in the **Caption** field. The caption appears above the table, and complies with W3C accessibility standards for screen reader users.
- Hide the column headings by deselecting **Show Column Headings** in the Table section.

- Make the columns sortable by selecting the jQuery Flexigrid theme in the Theme section and selecting **Enable Sorting**. The theme also changes the appearance of the table.
- Change the name of a column by selecting the column cell and updating its name in the **Column Heading** property in the Data Table section.

See Also

- [Edit Columns in a Data Table](#)
- [The Default, Error, and No Data Views](#)
- [Improve Performance Using Caching](#)
- [Data Filter Examples](#)

Edit Columns in a Data Table

A data table's columns bind to the fields of the object it's connected to. Each column cell acts as a placeholder that is replaced with the field's data when the page loads.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

You can display plain text, formatted text (for dates and numbers), or images in the column cells. You can also add hyperlinks to column cells to allow site visitors to navigate to another page, such as a detailed description, or to refresh the data displayed in the data table based on their selection. See [Data Filtering Examples](#).

To edit a column:

1. Double-click the column cell in the data table.
2. Select the field to display. To [customize how the field's data is displayed](#), click **Customize**.



Note The object's fields are listed first, followed by the fields of all [parent objects](#), which use the format *parent_object_name.field_name*.

3. Select the display type.

Option	Description
Text	Lets you display the field's data as plain text.
Formatted text	If you're working with dates, times, or currency, lets you choose from several text display formats.
Image	<p>If the field contains an image URL, lets you display the field's data as an image. The URL can be absolute or relative to the site.</p> <p>You can also select a field to use for the alternative text or enter custom text.</p>

4. To create a hyperlink, select **Add a hyperlink**. Otherwise, go to step 8.

5. Select the link type.

Option	Description
A URL	<p>Lets you link to a Web page by:</p> <ul style="list-style-type: none"> Choosing a field that you want to reference, such as a field that stores the relative URLs of PDFs you uploaded to your site. Choosing a field that you want to reference and clicking Customize to add an absolute URL or to create a custom link, such as a URL query string.
An item in your site	<p>Lets you link to a page, image, or file in the site by selecting the item type and then selecting the item. (If you can't see the list of items, place your cursor in the URL field and press the DOWN key on your keyboard.)</p> <p>You can also customize the URL—for example, by creating a URL query string.</p>
An email	<p>Lets you link to an email message by entering the recipient's address, and the message subject and body.</p> <p>You can use merge fields to access the object's</p>

Option	Description
	<p>fields. For example, if an object has an Email field, enter the merge field, such as <code>{!email}</code>, in the Email address text box.</p> <p>When the link is clicked, it opens a new message window in the user's email client and adds the appropriate email address to the To: field.</p>

6. Optionally, enter a tooltip by selecting the required field or clicking **Customize** to add custom text. The tooltip displays as a pop-up when the user hovers over the link.
7. If you're linking to a URL or an item in your site, specify where the item opens.

Option	Description
Popup window	Loads the item into a window. When you select this option, you can set the title for the popup and control its appearance and size with the options that appear.
New window (_blank)	Loads the item into a new, unnamed browser window.
Same window (_self)	Loads the item into the same frame or window as the link. This setting is the default.
Topmost window (_top)	Loads the item into the topmost parent frameset or window of the frame that contains the link.
Parent window (_parent)	Loads the item into the parent frameset or window of the frame that contains the link.

8. Click **Save**.

The column is displayed on the page as an expression. To test the output, [preview the page](#).

To change the name of a column, select the column cell and update the name in the **Column Heading** field of the Properties pane.

See Also

- [Dynamically Retrieve Data with Data Tables](#)
- [Site.com Data Services Overview](#)

[Add Pagination to Data Repeaters and Data Tables](#)

Events let you add interactive and animated effects to the pages and page elements of your website. When using data repeaters and data tables, you can add pagination events so users can easily page

through the displayed data. Pagination events are particularly useful when working with large amounts of data.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

For example, if you've added a data repeater that displays all the users in an organization, you can add pagination to help users navigate through the data. You can add three pagination events:

- Previous Page
- Next Page
- Go To Page

Create Previous and Next Pagination:

You can create previous and next buttons so users can move through the data one page at a time. The process for creating both buttons is the same.

1. Create your data repeater or data table.
2. In the data repeater or data table, be sure to specify how many records to display per page in the Limits section.
3. Drag a button to the page.
4. In the Properties pane, change the Button Name to Previous Page or Next Page as appropriate.
5. In the Events pane, select the click event.
6. When the Actions box appears, click  and select the Previous Page or Next Page action.
7. In the Target Element, select the data repeater or data table.
8. Click **Save**.

Create GoTo Pagination:

Creating GoTo navigation is similar to creating the previous and next buttons, but you must add an input field so users can specify what page they want to go to.

1. Create your data repeater or data table.
2. In the data repeater or data table, be sure to specify how many records to display per page in the Limits section.
3. Drag a **Number** field onto the page.
4. In the Properties pane, change the field's Label Name to something that makes sense. For example, Enter Page Number.
5. Drag a **Button** onto the page.
6. In the Properties pane, change the Button Name to GoTo Page.
7. In the Events pane, select the click event.
8. When the Actions box appears, click  and select the Go To Page action.
9. In the Target Element, select the data repeater or data table.
10. For Input Field ID, select the field you created in step 3.
11. Click **Save**.

See Also

[Dynamically Retrieve Data with Data Repeaters](#)

[Dynamically Retrieve Data with Data Tables](#)

[Create an Event](#)

[Available Events and Actions](#)

Using Data Functions

A data function lets you connect to a standard or custom Salesforce object, perform calculations on the returned results, and display the calculation on the page. For example, for a particular field in an object, you can use a data function to calculate the total value or the average amount of all returned records.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

USER PERMISSIONS NEEDED

To edit the guest user profile:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

AND

Manage Profiles and Permission Sets

AND

Customize Application

To add a data function to a page:

1. Drag a **Data Function** from the Page Elements pane onto the page.
2. Select the object that you want to connect to.



Note

- For Site.com users, the drop-down list only displays objects that are available to guest users because site visitors access your public site via the Guest User license. To make other objects available, go to the [guest user profile](#), enable the relevant object's "Read" permission, and refresh the list.
- For Communities users, the drop-down list displays objects that may not be available to site visitors. For authenticated visitors, object access on public and private pages is controlled by their user profiles. For unauthenticated visitors, object access on public pages is controlled by the site's guest user profile.

3. Optionally, in Filters, select criteria to filter your data set. If you don't select any criteria, all the data from the item is returned.
 - a. Select the field to which the filter criteria apply. The Field drop-down list displays the object's fields, followed by the fields of all parent objects, which use the format *parent_object_name.field_name*.
 - b. Select [the operator](#) to control how results are filtered. For example, select *Equals* to return an exact match.
 - c. Select [the source](#) of the filter value. For example, to specify an explicit value, select *Fixed value*, or to use the values passed to the page via a query string, select *URL query string*.
 - d. Set the value of the filter. If you're using a query string, you can also specify what should happen if the query string is missing.
 - e. Add additional filter criteria as required to narrow your results further. Each filter item is combined with an AND operator.



Note If you're using a fixed value to filter the results, you can view the returned records in the Connection Preview section. To refresh the list of records, click **Reload Preview**.

4. In Functions, select a function:

Option	Description
Count	Counts the number of records that contain a value for the selected field. For example, if an object contains 30 records, but only 25 records have a value in the field you specify, the result is 25.
Maximum	Returns the highest value of all the values for the selected field. Applies to numbers, strings, and dates.
Average	Calculates the average value of all records for the selected field. For example, if you have 20 records with a total value of \$20,000 in the Price field, the average is \$1,000. Only applicable to fields that contain numbers.
Minimum	Returns the lowest value of all the values for the selected field. Applies to numbers, strings, and dates.
Sum	Calculates the total value of all records for the selected field.

5. Select the field it applies to.

6. Click **Save**.

See Also

[The Default, Error, and No Data Views](#)
[Improve Performance Using Caching](#)
[Data Filter Examples](#)

Page Data Connections Overview

Page data connections make it easy for site administrators and designers to create a detail page for a single record when working with Salesforce objects.

REQUIRED EDITIONS

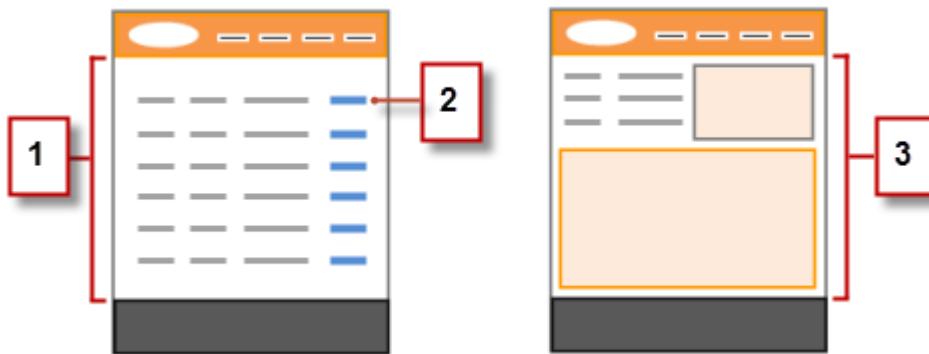
Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

When combined with repeater elements, custom code, or content blocks, page data connections let you connect to standard and custom objects, retrieve a specific record, and dynamically display the record on a site page.

 **Example** For example, let's say you want to list all of your company's accounts on a site page called Accounts. When a user clicks a **View Details** link, you want to open a site page called Account Detail to display information for the selected account. In this case, you could add a data repeater to the Accounts site page to retrieve a list of records from the Account object (1). Using a data element, you could create a **View Details** link (2) that, when clicked, uses a URL query string to pass the Account ID field as a unique identifier to the Account Detail page. As the Account Details page loads, the page data connection uses the unique identifier value to dynamically return only that record's details (3).



With page data connections, you can use expressions to access the returned data anywhere on the page, including the page's properties. For example, let's say you want to use the account name as the title of the Account Detail page. In this case, you would simply enter `{!Name}` in the Title field on the Properties pane. When the page loads, it retrieves the account name for that particular record and displays it in the browser's title bar.

See Also

- [Retrieving Data with Page Data Connections](#)
- [Site.com Data Services Overview](#)

Retrieving Data with Page Data Connections

Use a page data connection to create a detail page for a single record when working with Salesforce objects.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To edit the guest user profile:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

AND

Manage Profiles and Permission Sets

AND

Customize Application

When combined with data elements, custom code, or content blocks, page data connections let you connect to standard and custom objects, retrieve a specific record, and dynamically display the returned data anywhere on the page. You can even use expressions to access the returned data in the page's properties.

When the page is open:

1. Ensure that the page is selected in the Page Structure pane.
2. Click **Add Connection** in the Page Data Connection section of the Properties pane.
3. Select the object that you want to connect to.

 **Note**

- For Site.com users, the drop-down list only displays objects that are available to guest users because site visitors access your public site via the Guest User license. To make other objects available, go to the [guest user profile](#), enable the relevant object's "Read" permission, and refresh the list.
- For Communities users, the drop-down list displays objects that may not be available to site visitors. For authenticated visitors, object access on public and private pages is controlled by their user profiles. For unauthenticated visitors, object access on public pages is controlled by the site's guest user profile.

4. Optionally, in Filters, select criteria to filter which record is returned. If you don't select any criteria, the

first record is returned by default.

- a. Select the field to which the filter criteria apply. The Field drop-down list displays the object's fields, followed by the fields of all parent objects, which use the format *parent_object_name.field_name*.
- b. Select **the operator** to control how results are filtered. For example, select *Equals* to return an exact match.
- c. Select **the source** of the filter value. For example, to specify an explicit value, select *Fixed value*, or to use the values passed to the page via a query string, select *URL query string*.
- d. Set the value of the filter. If you're using a query string, you can also specify what should happen if the query string is missing.
- e. Add additional filter criteria as required to narrow your results further. Each filter item is combined with an AND operator.



Note If you're using a fixed value to filter the results, you can view the returned records in the Connection Preview section. To refresh the list of records, click **Reload Preview**.

5. In Sorting, you can specify whether to sort the results by one or more fields in ascending or descending order.

For example, say you want to create a page that displays the most popular movie. You could connect to a custom object that contains movie data, and instead of using filter criteria, you could sort by user rating in descending order. Because a page data connection returns a single record, only the highest rated movie is returned.

6. Click **Save**.

Next, you must add either **data elements**, **custom code**, or **content blocks** to the page to display the retrieved record data.

You can use expressions to access the returned data anywhere on the page, including the page's properties. For example, let's say you want to use the account name as the title of the Account Detail page. In this case, you would simply enter `{ !Name }` in the **Title** field on the Properties pane. When the page loads, it retrieves the account name for that particular record and displays it in the browser's title bar.

See Also

- [Page Data Connections Overview](#)
- [Improve Performance Using Caching](#)
- [About Displaying Dynamic Data Using Expressions](#)
- [Data Filter Examples](#)

Access Data in Related Objects Overview

Standard and custom objects have relationships that define how records in one object relate to records in another. For example, the Accounts object has a one-to-many relationship with the Contacts object—that is, each account can have one or more contacts associated with it. This relationship is also known as a parent-to-child or a master-detail relationship.

REQUIRED EDITIONS

! **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Data repeaters, data tables, and data functions take advantage of these relationships to let you display data from related objects on the page.

Access Data in Parent Objects:

When you add a data repeater, data table, or data function to the page and connect it to a standard or custom object, you can automatically access the fields of any parent object it's related to.

If you add [filter criteria](#) to a data repeater, data table, or data function, the Field dropdown list in the Create Data Connection dialog box displays the object's fields, followed by the fields of all of its parent objects, which use the format *parent_object_name.field_name*. This lets you filter results based on a field in the parent object. So for example, when retrieving records from the Contacts object, you could decide to return only contacts where the account name (**Account.Account Name**) is "ABC Labs."

Similarly, when you add data elements to a data repeater, or columns to a data table, you can bind them to fields in a parent object. So for example, if you add a data table that's connected to the Contact object, you can add a column that binds to its **Full Name** field and a column that binds to the Account object's **Account.Account Name** field to display the contact's name along with the name of the account it's associated with.

Access Data in Child Objects:

You can retrieve data from any child object of a parent object using a data repeater that contains another data repeater, data table, or data function. The outer or parent data repeater connects to an object, such as Accounts. In turn, the inner data repeater, data table, or data function automatically lets you connect to any child objects, such as Contacts. It's also known as a *nested data repeater*.

Let's say you want to display a list of accounts along with the names of the associated contacts, similar to this example.



You can achieve this by creating a data repeater (1) that's connected to Accounts, and adding a data element (2) to it that binds to the **Account Name** field. Then add a nested data repeater (3) that's connected to Contacts, which is a child of Accounts. Finally, add a data element (4) to the nested data repeater that binds to the Contact object's **Full Name** field.



See Also

- [Displaying Data from Related Objects Using Nested Data Repeaters](#)
- [Dynamically Retrieve Data with Data Repeaters](#)
- [Dynamically Retrieve Data with Data Tables](#)

Displaying Data from Related Objects Using Nested Data Repeaters

You can retrieve data from any child object of a parent object using a data repeater that contains another data repeater, data table, or data function.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To edit the guest user profile:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

AND

Manage Profiles and Permission Sets

AND

Customize Application

The outer or parent data repeater connects to an object, such as Accounts. In turn, the inner data repeater, data table, or data function automatically lets you connect to any child objects, such as Contacts. This is also known as a *nested data repeater*. For example, if a data repeater is connected to Accounts, you can add a nested data function to it that's connected to Contacts to return the number of contacts associated with each account.

To create a nested data repeater:

1. [Add a data repeater](#) to the page.
2. Drag another **Data Repeater**, **Data Table**, or **Data Function** from the Page Elements pane onto the data repeater.
3. Select the related Salesforce object that you want to connect to.

 **Note** You can also retrieve data from unrelated objects. However, as this can adversely affect the performance of your site, we recommend retrieving data from related objects only.

4. Optionally, in Filters, select criteria to filter your data set. If you don't select any criteria, all the data from the item is returned.
 - a. Select the field to which the filter criteria apply. The Field drop-down list displays the object's fields, followed by the fields of all parent objects, which use the format *parent_object_name.field_name*.
 - b. Select [the operator](#) to control how results are filtered. For example, select *Equals* to return an exact match.
 - c. Select [the source](#) of the filter value. For example, to specify an explicit value, select *Fixed value*, or to use the values passed to the page via a query string, select *URL query string*.
 - d. Set the value of the filter. If you're using a query string, you can also specify what should happen if the query string is missing.
 - e. Add additional filter criteria as required to narrow your results further. Each filter item is combined with an AND operator.

 **Note** If you're using a fixed value to filter the results, you can view the returned records in the Connection Preview section. To refresh the list of records, click **Reload Preview**.

5. In Sorting, you can specify whether to sort the results by one or more fields in ascending or descending order. For example, if you're working with an object that contains user data, you could sort your results by gender first and then by name.
6. In Limits, you can limit the number of returned results. For example, if you're only interested in the top five results, enter 5 in the **Limit results to** field.
7. If you're adding [pagination](#), specify the number of results to display per page in the **Results per page** field.
8. If you're working with a data table, click **Next** and add fields to the table by double-clicking a field, or selecting it and clicking .
9. Click **Save**.

You can either add [data elements](#) or [custom code](#) to the parent repeater to display its fields. Similarly, if you've nested a data repeater inside the parent data repeater, add data elements or custom code to the nested data repeater to display the child object's fields.

 **Note** You can't nest data repeaters more than one level deep.

See Also

- [Access Data in Related Objects Overview](#)
- [Dynamically Retrieve Data with Data Tables](#)
- [Using Data Functions](#)

Improve Performance Using Caching

When working with data-bound page elements, such as data repeaters, data tables, and data functions, you can improve the performance and page rendering of your website using caching. Caching controls how often a page containing a data connection requests data from Salesforce.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

Lets say 100 people visit the site page at the same time. Without caching, the page makes 100 separate requests for the same data, slowing performance considerably. However, with caching enabled, the data is requested and retrieved only once—the first time someone visits the page. Any subsequent requests for data during a set time period are returned from the cache. When the specified time period expires, the cache is refreshed.

The Cache Duration (Minutes) field in the Properties tab controls the length of time to cache retrieved data for the selected data repeater, data table, or data function.

The default value is 30 minutes. However, the appropriate number of minutes depends on your requirements. For example:

- If the data is updated frequently, such as in a commenting system or a stock ticker, you can disable caching by setting the value to zero to ensure the freshest data displays on the page.
- If the data changes infrequently, say just once a week, you can set the value to a much greater number of minutes. A longer caching period also helps ensure pages can display data even if the data source is momentarily unavailable.



Note Whenever updates to a site are published, the cache is deleted for all data connections.

Caching begins again the next time a site visitor accesses the page.

See Also

- [Dynamically Retrieve Data with Data Repeaters](#)
- [Dynamically Retrieve Data with Data Tables](#)
- [Using Data Functions](#)

Data Filters

When you add a data repeater, a data table, or a data function to a page, you don't have to limit the records it retrieves. However, if you're working with a Salesforce object that has thousands of records, you can limit the returned results using filter criteria.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

When you add filter criteria, you must specify:

- The field to which the filter criteria apply
- The operator
- [The source of the filter value](#)
- The filter value

Filter Operators

Operator	Description
Equals	Returns an exact match.
Not equal to	Returns records that don't have the value you specify.
Less than	Returns records that are less than the value you specify.
Greater than	Returns records that exceed the value you specify.
Less than or equal to	Returns records that match or are less than the value you specify.
Greater than or equal to	Returns records that match or exceed the value you specify.
Starts with	Use when you know what your value starts with, but not the exact text. For example, "california" would return California Travel, but not Surf California.
Ends with	Use when you know what your value ends with, but not the exact text.
Contains	Returns records that include your search string but can also include other information. For example, "california" would return California Travel and Surf California.
Includes	Use for field types that support multiple values to return records that contain

Operator	Description
	one or more of the comma-separated values you specify.
Includes all	Use for field types that support multiple values to return records that contain all of the comma-separated values you specify.
Excludes	Use for field types that support multiple values to return records that don't contain one or more of the comma-separated values you specify. For example, for a Locations category field, "San Francisco, Vancouver" would exclude a record containing "San Francisco, Dallas" and a record containing "Vancouver, New York."
Excludes all	Use for field types that support multiple values to return records that don't contain all of the comma-separated values you specify. For example, for a Locations category field, "San Francisco, Vancouver" would exclude only records that contain <i>both</i> terms.

Filter Value Sources

Source	Description
Fixed value	Use when you want to specify the value.
URL query string	Use when you want to pass variable content via a URL to the item when the page loads.
Global property	Use when you want to use a fixed value from the site, such as the current date or current time.
Request header	Use when you want to use a value from the browser, such as the host header or browser version.
Parent repeater	<p>Use when you want to create a query between unrelated objects, or between content lists or categories. Available only when a data repeater, data table, or data function is nested inside a parent repeater, but the parent repeater's object is unrelated to the nested item's object, and the selected data sources for both are unrelated.</p> <p>Retrieving data from unrelated objects can adversely affect the performance of your site. We recommend retrieving data from related objects only.</p>

See Also

[Dynamically Retrieve Data with Data Repeaters](#)

[Dynamically Retrieve Data with Data Tables](#)

[Using Data Functions](#)

[About Displaying Dynamic Data Using Expressions](#)

About Displaying Dynamic Data Using Expressions

Site.com uses expression language to display data dynamically. Expressions serve as placeholders for data that is replaced with information when the page loads. When working with data-bound page elements or custom widget properties, you can use expressions to customize how data is displayed on the page.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

In Site.com, expression syntax consists of an open curly brace and exclamation point, the field, custom property name, or namespace name, and a closing curly brace.

For example, if you added a custom property called **URL** to a widget, you can use the syntax `{!URL}` to add the expression to custom code or content blocks. Similarly, to add an expression for the Billing City field, you can use the syntax `{!BillingCity}`. For related objects, the field name is prefixed by the name of the master object: `{!Account.BillingCity}`.

If you're editing a [data element](#) in a data repeater or a [data table column](#), you can access the object's fields by name in a dropdown list without using expressions. However, if you want to customize how the field is displayed on the page, you can see and edit the expression when you click **Customize** in the dialog box.

If you're working with [custom code](#) or [content blocks](#) in a data repeater, you can only access the object's fields using expressions.

Expressions let you customize the output by:

- Adding text around the expression. For example, lets say you're displaying the phone number of each of your business locations. You could enter the text *Contact us at* before the `{!Phone}` expression. When the data is displayed on the page, `{!Phone}` is replaced with the field's value for each record:
Contact us at 100-200-3000.
- Formatting the output using HTML tags. For example, you could wrap `H1` tags around the expression to alter how the output is displayed on the page: `<H1>{!Phone}</H1>`.
- Creating a URL query string to pass variable information to a data repeater or data table on another page. The second page, in turn, uses the received variable to retrieve and display the relevant records. For example, you could create a hyperlink, such as `/product_details?productID={!id}`, where `{!id}` is replaced with the product ID for each record. When a particular product link is clicked, the product ID is passed to the Product Details page, which uses the ID to retrieve the record's information and display it on the page.

See Also

- [Dynamically Retrieve Data with Data Repeaters](#)
- [Dynamically Retrieve Data with Data Tables](#)
- [Using Data Functions](#)

Data Filter Examples

When working with data repeaters, data tables, and data functions, you can filter the data you retrieve in many ways. In this topic, we explore two options—fixed values and URL query strings—to illustrate some common filtering techniques.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Example 1: Use a Fixed Value to Filter Results

In our first example, we have a custom object called “News” that stores company news and events. Each news item has a **Status** picklist that can be set to either In Progress or Approved. We only want to display approved news items on our Company News page. In this case, we can use a fixed value to filter the data in the News object.

In this scenario:

1. Add a data repeater to the page and configure it as follows:
 - a. Select the News custom object.
 - b. In the Filters section, set the criteria to **Status Equals Fixed value**.
 - c. Enter *Approved* in the Value text box.

This configuration tells the data repeater to return only records where the **Status** field contains the value *Approved*.

2. To display the required fields, such as Title, Description, and Date, on the page, add data elements to the data repeater.

Example 2: Use a URL Query String to Dynamically Filter and Display Results on Another Page

In our second example, we have a custom object called “Products” that stores product information. However, some products are only available in certain locations, so we like to let customers view the products in their nearest city. In this case, we could create a Locations site page that contains links to each available city.

We want each link to open the Products site page, but only display products based on the user's selection. We can do this using URL query strings, which let us pass variable content between HTML

pages. The variable information in this case is the product location; the data repeater doesn't know which products to return until the user makes a selection.

In this scenario:

1. Add a data repeater to the Products page and configure it:
 - a. Select the Products custom object.
 - b. In the Filters section, set the criteria to *City Equals URL query string*.
 - c. In the Value text box, enter the variable name—in this case, *location*.
 - d. If query string is missing dropdown list, select **Don't apply this filter item**. This option is used when a customer wants to view all products without filtering.
 - e. Click **Save**.
2. Add data elements for the fields you want to display, such as Product Name, Description, and Price.
3. On the Locations page, add a data repeater and select the Products custom object.
4. Add a data element to the data repeater to represent the **City** field and configure it:
 - a. Select the **City** field as the field to display, because we want to use the name of the city as the hyperlink.
 - b. Select Text as the display format.
 - c. Select **Add a hyperlink** to display a URL on the page.
 - d. In the Link to dropdown list, select An item in your site.
 - e. Select Page as the type and select the Products page. (If you can't see the list of pages, place your cursor in the URL text box and press the DOWN key on your keyboard.)
 - f. Change the URL value to `/product?location={!City}`. In this case, `{!City}` is a placeholder for the value of a record's **City** field. When the page first loads, `{!City}` is replaced with the correct value, such as Dallas, which creates this URL for that record:

```
/product?location=Dallas
```

When clicked, the Products page opens with `Dallas` listed as the value of the **location** variable.

- g. For the tooltip, select the **City** field and click **Customize**.
- h. Change the value to *Show me products available in {!City}*.

Again, the value of the **City** field replaces the `{!City}` placeholder for each record when the page loads.

Now, when the Locations page loads, the data repeater displays the location of each product as a link. When a customer clicks a link, such as Dallas, the Locations page passes `location=Dallas` to the Products page. As the Products page loads, the data repeater uses this value to dynamically return only records where the **City** field contains the value `Dallas`.

See Also

- [Dynamically Retrieve Data with Data Repeaters](#)
- [Display Data or Content Using Data Elements](#)
- [Dynamically Retrieve Data with Data Tables](#)
- [Access Data in Related Objects Overview](#)

Adding a Form to the Page

Use forms to collect data from your site visitors and submit the data to standard or custom Salesforce objects. Create web-to-lead forms, capture customer details, or gather feedback on your products or services.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To add a form to the page:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned in Site.com Studio

To edit the guest user profile:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned in Site.com Studio

AND

Manage Users

AND

Customize Application

To add a form to a page:

1. Drag a **Form** from the Page Elements pane onto the page.
2. Select the Salesforce object that you want to submit data to.



Note

- For Site.com users, the drop-down list only displays objects that are available to guest users

because site visitors access your public site via the Guest User license. To make other objects available, go to the guest user profile, enable the relevant object's Create permission, and refresh the list.

- For Communities users, the drop-down list displays objects that may not be available to site visitors. For authenticated visitors, object access on public and private pages is controlled by their user profiles. For unauthenticated visitors, object access on public pages is controlled by the site's guest user profile.
3. Add available fields to the form by double-clicking a field, or selecting it and clicking .
 - All required fields are automatically added to the list of selected fields. However, you can hide required fields after you add the form to the page.
 4. Reorder the list of selected fields by clicking **Move Up** or **Move Down**.
 5. Click **Save**.

-  **Note** When adding forms to authenticated community pages in Site.com, set the current user for Salesforce objects that require the Owner ID field. Setting the current user (as opposed to the default guest user) lets you identify the authenticated user when the form is submitted. To set the current user for the Owner ID field, select the field in the form, and click **Configure**. Under Field Properties in the Properties pane, select **Global Property** as the source, and select **Current userID** as the value.

After you add a form to the page, you can't change the object it's connected to. If you need to connect to a different object, you must replace the form.

You can use the form's Properties pane to:

- See which object the form is connected to.
- Add a title to the top of the form.
- Specify what occurs when a user successfully submits the form.
- Change the appearance of the form by selecting a different theme.

Add Input Fields to Forms or Pages

Add additional input fields to an existing form. Each input field binds to a field in the object the form is connected to. Add input fields directly to a page, panel, data repeater, or data table to build your own custom features using

Editing Input Fields in a Form

After you've added a form to the page, you can edit and reorder its fields.

Input Field Types

When adding input fields to a form or page, the following field types are available. However, if the object connected to a form doesn't contain a particular field type, you can't add that input field type to the form.

Input Field Properties

When adding or editing input fields on a form or page, use the options in the Field Properties section of the Properties pane to control how the selected input field functions.

Setting the Default Value of Input Fields

You can set the default value of an input field that you add to a form or page. This automatically populates the input field with the value you specify when the page loads.

Setting a Form's Submit Behavior

When your site visitors submit a form successfully, you can either redirect them to another page or display a message indicating that they were successful.

Styling Forms

Forms are styled using CSS themes that you can customize to match the design of your website.

See Also

[Add Input Fields to Forms or Pages](#)

[Input Field Types](#)

[Editing Input Fields in a Form](#)

Add Input Fields to Forms or Pages

Add additional input fields to an existing form. Each input field binds to a field in the object the form is connected to. Add input fields directly to a page, panel, data repeater, or data table to build your own custom features using

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Add Input Fields to a Form:

You can add additional input fields to an existing form. Each input field binds to a field in the object the form is connected to.

The quickest option is to:

1. Select the form on the page.
2. Select  | Add Fields.

3. In the Add Fields list, click the fields that you want to add. The Add Fields list displays the available fields in the object that the form is connected to. When you click a field, the correct field type is automatically added to the form, such as a checkbox or picklist field.

Alternatively:

1. Select the form on the page.
2. Select  | Add Page Elements.
3. In the Add Page Elements list, click the input field type that you want to add, such as **Checkbox**.
4. Choose a field in the Add a Field dialog box and click **Save**. If no fields of that type exist in the object, you can't add a field of that type to the form.



Note

- You can't add fields to a form by clicking  | Edit Form.
- Formula, encrypted text, geolocation, and lookup field types aren't supported.
- You can't configure the default field-level error messages that appear when users enter an incorrect value.

Add Input Fields to a Page:

You can add input fields directly to a page, panel, data repeater, or data table to build your own custom features using [custom code](#). For example, let's say some of your products are only available in certain locations, and you want to let customers view the products in their nearest city. You could add a picklist input field to the page that lists the various locations. Using custom code, you could then pass the user's selection to a [data table](#) or [data repeater](#) via a [query string](#) to display a filtered product list.

To add an input field to the page, drag it from the Page Elements pane onto the page. Alternatively, select the page or container page element in the Page Structure pane, select  | Add Page Elements, and select the input field.

See Also

- [Input Field Types](#)
- [Input Field Properties](#)
- [Editing Input Fields in a Form](#)
- [Adding a Form to the Page](#)

[Editing Input Fields in a Form](#)

After you've added a form to the page, you can edit and reorder its fields.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To reorder fields, drag them to the correct position on the page or in the Page Structure pane.

Alternatively, select a field and click | **Move Up** or | **Move Down**.

To hide a field, such as required field that you don't want your site visitors to see, select **Hidden Field** in the Properties pane. When you hide a field, you can access it from the Page Structure pane.

To make a field a required field, which means the user must complete the field before submitting the form, select **Required Field** in the Properties pane. A red asterisk (*) is displayed beside the field to indicate that it's required. You can't change the **Required Field** setting for any fields that are required by the object the form is connected to.

To rename a field, replace the name in the **Label Name** field in the Properties pane.

To change the appearance of a field on the page, select a different theme in the Properties pane. If the field is in a form, you can only change the form's theme.

Note

- You can't move fields from a form onto the page. However, you can delete non-required fields, or hide both required and non-required fields.
- You can't drag fields from the page onto a form.

See Also

[Add Input Fields to Forms or Pages](#)

[Input Field Properties](#)

[Setting the Default Value of Input Fields](#)

[Adding a Form to the Page](#)

Input Field Types

When adding input fields to a form or page, the following field types are available. However, if the object

connected to a form doesn't contain a particular field type, you can't add that input field type to the form.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

Page Element	Description
Checkbox	Lets users set a true (selected) or false (deselected) value on a form or page.
Currency	Lets users enter a currency amount on a form or page.
Date	Lets users enter a date on a form or page. If the field's Date/Time Selector property is set to Popup Calendar, the user can pick the date from a popup calendar.
Date/Time	Lets users enter a date and time on a form or page. If the field's Date/Time Selector property is set to Popup Calendar, the user can pick the values from a popup calendar.
Email	Lets users enter a valid email address on a form or page.
Multi-Select Picklist	Lets users select one or more values from a list on a form or page.
Number	Lets users enter a whole number on a form or page.
Percent	Lets users enter a percent amount on a form or page.
Phone	Lets users enter a phone number on a form or page.
Picklist	Lets users select a value from a list on a form or page.
Rich Text Area	Lets users enter up to 32,768 characters on a form or page. Supports any combination of letters, numbers, or symbols. Users can format the text, and add images and hyperlinks. A rich text area field can't be used for image uploads in Site.com sites or Salesforce Sites due to security constraints.
Text	Lets users enter up to 255 characters (depending on the field limit in the Salesforce object) on a form or page. Supports any combination of letters, numbers, or symbols.
Text Area	Lets users enter up to 32,768 characters on a form or page,

Page Element	Description
	which display on separate lines. Supports any combination of letters, numbers, or symbols.
URL	Lets users enter a valid website address on a form or page.

See Also

[Add Input Fields to Forms or Pages](#)

[Input Field Properties](#)

[Adding a Form to the Page](#)

Input Field Properties

When adding or editing input fields on a form or page, use the options in the Field Properties section of the Properties pane to control how the selected input field functions.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Name	Description
Label Name	The external name of the field. This is displayed as the field name on the form or page.
Default Value	Sets the field's default value. This automatically populates the field with the value you specify when the page loads.
Required Field	When selected, makes the field mandatory, so the user can't submit a record without entering a value. A red asterisk (*) is displayed beside the field to indicate that it's required. You can't change the Required Field setting for any fields that are required by the object the form is connected to.
Hidden Field	When selected, hides the field from the form or page. For example, you may want to hide a required field from your site visitors, or set the default value for a field that you don't want users

Name	Description
	to see or edit. When you hide a field on the page canvas, you can still access it from the Page Structure pane.
Date/Time Selector	Applies to the Date/Time field only. Sets whether users can select the date and time using a popup calendar.
Picklist Values	<p>Applies to Picklist and Multi-select Picklist fields only, and only when added directly to the page. Sets the list of items to display in the picklist. (You can't modify the picklist items if the field is connected to an object.)</p> <p>Each picklist item consists of a Label and a Value field. The Label field is displayed in the picklist, whereas the Value field is an internal value. In most situations, you can use the same value in both fields, unless you want to submit a different value than the one displayed to the user.</p> <p>Click  to enter values for the Label and Value fields.</p>
Rows	Applies to Rich Text Area and Text Area fields only. Sets the number of rows of text to display.

See Also

[Input Field Types](#)

[Add Input Fields to Forms or Pages](#)

[Editing Input Fields in a Form](#)

Setting the Default Value of Input Fields

You can set the default value of an input field that you add to a form or page. This automatically populates the input field with the value you specify when the page loads.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

When used with a field's **Hidden Field** property, which hides the field on the form or page, the default value is a useful tool for submitting data that you don't want your users to see. For example, you could add a hidden field that uses the Global Property option to track when a form is submitted.

 **Note** Default field values that are already set on the object aren't transferred to Site.com.

To set the default value of a field:

1. Select the field.
2. Click **Configure** in the Properties pane.
3. Select the source of the default value.

Option	Description
Fixed Value	Use when you want to specify the value. For example, for a text field, you could add default text. Alternatively, if the field is a picklist, you can select a default value from the list.
URL Query String	Use when you want to pass variable content via a URL to the item when the page loads. For example, if you're working with a picklist field, you could pass variable information, such as location, to filter the list and display only items from that location. See Data Filtering Examples for similar scenarios using data repeaters.
Global Property	Use when you want to use a fixed value from the site, such as the current date or current time.
Request Header	Use when you want to use a value from the browser, such as the host header or browser version.

4. Set the default value.
5. Click **Save**.

See Also

- [Add Input Fields to Forms or Pages](#)
- [Editing Input Fields in a Form](#)
- [Adding a Form to the Page](#)

Setting a Form's Submit Behavior

When your site visitors submit a form successfully, you can either redirect them to another page or display a message indicating that they were successful.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

When the page is open:

1. Select the form in the Page Structure pane.
2. In the Properties pane, select an option in the When Successful drop-down list.

Option	Description
Show another page	Redirects the user to the page that you specify.
Display a message	Displays a message below the form.

3. As appropriate, specify either:
 - The page to redirect to in the Page URL field. You can enter a relative URL, such as a site page, or an absolute URL. For relative URLs, ensure you include a forward slash (/).
 - The message text in the Message Text field.

 **Tip** The form's **Submit** button uses the submit action in the [Events pane](#).

See Also

- [Adding a Form to the Page](#)

[Input Field Properties](#)
[Site.com Data Services Overview](#)

Styling Forms

Forms are styled using CSS themes that you can customize to match the design of your website.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

When you add a form to a page, the form uses a default theme to control its appearance. To change the form's theme, select an option in the Theme section in the Properties pane:

- **Default** places the field name above the text box. Required fields are denoted by a red asterisk (*) beside the field name.
- **Salesforce** places the field name to the left of the text box. Required fields are denoted by a red vertical bar (|) in front of the text box.

Alternatively, to customize a theme to suit your needs:

1. Select the form on the page.
2. Select a theme to use as a base in the Theme section of the Properties pane.
3. Open the Style pane and ensure **Class** is selected.
4. In the Style drop-down list, select the part of the form that you want to style. When you select an item, it's highlighted for a few seconds so you can easily see which part you're styling.

Option	Description
Entire form	Styles the whole form.
Field rows	Styles each field row.

Option	Description
Field labels	Styles the labels of each field.
Fields	Styles the fields (text boxes, drop-down lists, and so on). Only available for the Salesforce theme.
Required symbol	Styles the asterisk symbol (*) for required fields. Only available for the default theme.
Error message	Styles the error message that's displayed when users try to submit an incorrectly completed form. Only available for the default theme.

 **Tip** If you're familiar with CSS, you can also modify the style of the form in the site's style sheet.

- To style the selected part of the form, update the [Style pane properties](#). Your theme customizations are reflected immediately in the form, and apply to the selected form only.
- Repeat as required for each part of the form.

See Also

[Adding a Form to the Page](#)

The Default, Error, and No Data Views

When working with data repeaters, data tables, data functions, and forms, you can customize what your site visitors see if an error occurs when connecting to the data source.

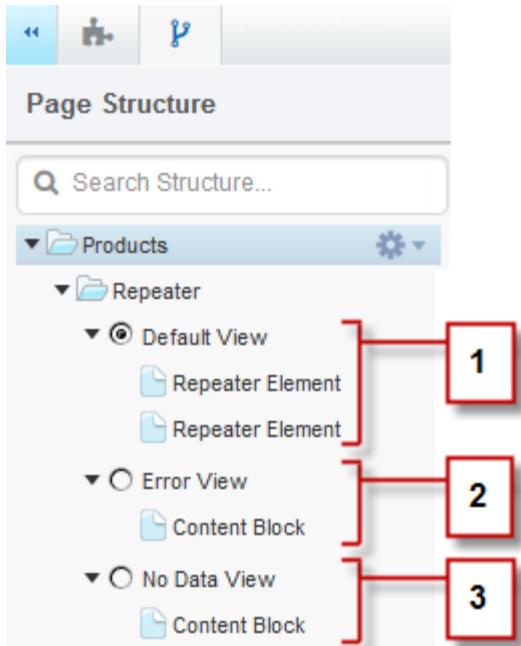
REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

Additionally, for data tables, data repeaters, data functions, you can customize what's displayed when no data exists for the current query. For example, if you set up a data repeater to dynamically filter and display results based on the user's selection, but there are no results for that selection, you can display an appropriate message to explain what happened.



- The Default View (1) is what usually appears on the page. For example, when you add a form to the page, the Default View contains the form's fields.
- The Error View (2) is displayed if an error occurs when connecting to the data source. It contains a content block with a default message that you can customize. You can also add other page elements to the view.
- The No Data View (3) is displayed when no data exists for the current query. It contains a content block with a default message that you can customize. You can also add other page elements to the view.

See Also

- [Dynamically Retrieve Data with Data Repeaters](#)
- [Dynamically Retrieve Data with Data Tables](#)
- [Using Data Functions](#)

Repairing Data Connections

If you or another user modifies the object that an existing data repeater, data table, data function, or form is connected to, the data connection might break. For example, this can happen if a connected object or field is renamed or deleted, or if its permissions are changed.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

When you open a site page containing broken data connections, a dialog box that lists the problems appears. Hover over each item's  icon to see the possible solutions.

Problem	Solution
A data repeater, data table, or data function is connected to an object that doesn't have the correct access permission, or has been renamed or deleted.	<p>Open the guest user profile and ensure the Read permission is enabled on the object.</p> <p>If that's unsuccessful, click Edit beside the listed item to open the Data Connection dialog box, and ensure the correct object is selected. If the object has been renamed, select the renamed object. If the object has been deleted, you must either choose a different object or delete the page element.</p>
A data repeater, data table, or data function is trying to filter results using fields that are longer visible, or have been renamed or deleted.	<p>Open the guest user profile and ensure the object has the correct field-level security enabled for each field.</p> <p>If that's unsuccessful, click Edit beside the item to open the Data Connection dialog box, and reset the filter criteria.</p>
A data element is connected to a field that's no longer visible, or has been deleted or renamed.	<p>Open the guest user profile and ensure the object has the correct field-level security enabled for the field.</p> <p>If that's unsuccessful, click Edit beside the item to open the Edit Data Element dialog box, and ensure the correct fields are referenced in all relevant drop-down lists and in any custom text.</p>
A form is connected to an object that doesn't have the correct access permission, or has been renamed or deleted.	<p>Open the guest user profile and ensure the Create permission is enabled on the object.</p> <p>If that's unsuccessful, you must replace the form, as you can't edit a form's data connection.</p>
A form is missing one or more required fields, which were added to the object after the form was created.	Add the missing field to the form.
A form field is pointing to a field that's no longer visible, or has been renamed or deleted.	Open the guest user profile and ensure the object

Problem	Solution
	<p>has the correct field-level security enabled for the field.</p> <p>If that's unsuccessful, remove the field from the form.</p>

 **Note** If you're a Communities user working with authenticated pages, keep in mind that object access on public and private pages is controlled by the user profile of the authenticated user. The guest user profile controls object access on public pages for unauthenticated visitors only.

See Also

[Setting Data Access Permissions for Salesforce Objects](#)

[Dynamically Retrieve Data with Data Repeaters](#)

[Add Input Fields to Forms or Pages](#)

Widgets Overview

Widgets let you save time by building custom page elements that you and your team can reuse throughout the site.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

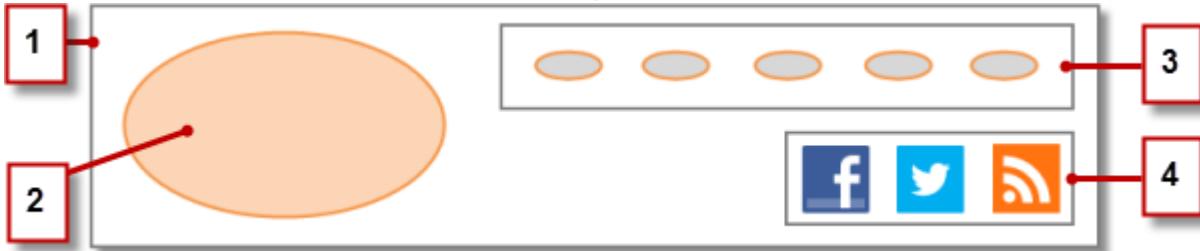
Using the existing Site.com page elements, such as panels, content blocks, custom code, or data repeaters, you can create widgets to suit your unique requirements. You can add custom properties to allow greater flexibility over how your widgets are reused. And you can even add a widget to another widget!

By using CSS to style the widget, you can ensure it always appears correctly whenever it's added to site pages or page templates. Additionally, you can let contributors add widgets to site pages and add branding properties that enable contributors to update the widget's appearance.

With widgets, you can:

- Minimize duplication in your site. You build one time, then reuse.
- Reduce maintenance overheads for you and your team. Any updates you make to a widget are automatically reflected in copies of the widget on the page.
- Improve the load time of your pages.

 **Example** For example, you could use the custom code page element to create reusable social plug-in widgets, such as **Like** or **Follow** buttons from Facebook, or a Twitter feed.  Alternatively, to ensure a consistent look and feel across all of your pages, you could create a corporate header such as this sample header widget. It consists of a panel (1) that contains a company logo (2) and a menu (3). The widget also contains another widget (4) that's composed of a panel containing custom code for Facebook, Twitter, and RSS plug-ins.



You could also use widgets to store commonly used pieces of text, such as company names, addresses, legal text, and so on. Simply create a widget that contains a content block with the relevant text.

Best Practices for Using Widgets

- Wherever possible, use widgets to reduce duplication in your site design. Reducing duplication is particularly important when working with complex designs or site elements, where maintenance can be time consuming.
- If you plan to use the same widgets across several sites, consider creating a basic site to contain all the required widgets. Then, for each new site, create a copy of that basic site. That way, each new site automatically includes all of the widgets you created.
- When using CSS to style widgets, add the CSS to the Site Style Sheet, which is the site's global style sheet. Because every page automatically references the style sheet, you can ensure that each widget appears correctly on the page.

Create Widgets

Widgets let you create custom, reusable page elements by combining existing Site.com page elements, custom code, and CSS.

Add a Widget to a Page

Add a widget that enhances your page.

Edit and Delete Widgets

Access the site's widgets in the Widgets view under All Site Content (on the Overview tab).

See Also

[Create Widgets](#)

[Custom Properties for Page Templates or Widgets Overview](#)

[Site Branding Overview](#)

Create Widgets

Widgets let you create custom, reusable page elements by combining existing Site.com page elements, custom code, and CSS.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

When you create a widget, it's added to the Widgets view on the Overview tab, where you can access and manage all of the site's widgets. If you make it available, the widget also appears in the Widgets section of the Page Elements tab, where you and your team can easily drag it onto the page. You can also let contributors add widgets to site pages.

1. Hover over **Widgets** on the Overview tab and click **New**, or click **New Widget** when the Widgets view is open.
2. Enter the widget name. This name appears in the Page Elements pane.
3. Optionally, add a description.
The description appears as a tooltip in the Page Elements pane.
4. To add a display icon for the widget, select an image from your imported assets. Use an icon that's 16 x 16 pixels in size.
5. Optionally, deselect **Available in the Page Elements Pane** if you don't want the widget to appear in the Page Elements pane.
For example, you don't want the widget to appear until you've finished building it.
6. Click **Apply**. Now you're ready to add page elements to the widget.
7. Click the widget's name to open it in a new tab.
8. Add the page elements and CSS styles you need.
9. Optionally, add custom properties or branding properties to the widget.

After you finish building the widget, ensure it's available in the Page Elements tab by selecting **Available in Page Elements Pane** on the Properties pane.

To let contributor's add a widget to a site page, select **Available to Contributors** on the Properties pane. This setting controls whether the widget appears in the contributor's Page Elements pane. You must also ensure that the site includes a template-based site page with at least one editable panel.

See Also

- [Widgets Overview](#)
- [Custom Properties for Page Templates or Widgets Overview](#)
- [Site Branding Overview](#)
- [Add a Widget to a Page](#)

Add a Widget to a Page

Add a widget that enhances your page.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Before you can add a widget to a page, you must make sure it's available for use. Either:

- Select **Available in Page Elements pane** in the Properties pane when the widget is open.
- Hover over the widget in the Widgets view on the Overview tab, select  | **Edit Properties**, and select **Available in Page Elements pane**.

When the page is open:

1. Drag the widget from the Widgets section of the Page Elements pane onto the page.
When you add a widget to a page, it creates a copy or *instance* of the widget. You can't edit this widget instance.
2. If available, update the properties in the widget instance's Properties pane.

See Also

[Widgets Overview](#)[Custom Properties for Page Templates or Widgets Overview](#)[Edit and Delete Widgets](#)[Create Widgets](#)

Edit and Delete Widgets

Access the site's widgets in the Widgets view under All Site Content (on the Overview tab).

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To update a widget, hover over it and click . From the  menu, you can:

- Edit the widget. When you edit a widget, it opens in a new tab. Any updates you make are reflected immediately in the site pages or page templates that reference it.
- Update the widget's properties, including the name, description, display icon, and availability. In the Widgets view, you can also edit the name, description, or availability inline by double-clicking the item.
- Duplicate, preview, or delete the widget. If a widget is being used on a site page, page template, or another widget, you can't delete it.

See Also

[Widgets Overview](#)[Create Widgets](#)[Add a Widget to a Page](#)

Multilingual Sites Overview

Site.com Studio lets you to create different language versions of your site. And because all languages are maintained within the site, you don't need to create and manage a separate site for each language.

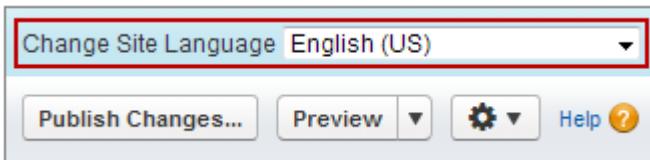
REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

As a site administrator or designer, you can add and manage the languages that you want your site to support. After you add a language to the site, you'll notice the Site.com Studio language selector on the toolbar, which lets you and your team switch between languages when editing the content of a page. This makes content editing quick and easy because you never have to leave the page to change to another language.



Contributors, designers, and site administrators can each add language-specific content to a page using the language selector.

As a site administrator or designer, you can also export site content as an .xml file and send it to your translation service. After you receive the edited file, simply import it back into your site to populate each page with the translated content.

In turn, to let your site visitors choose their preferred language from those available when viewing the site, you can add a Language Selector page element to your pages. And in case any part of the site content isn't available in their chosen language, you can specify a *fallback* language to display instead.

 **Example** For example, if a site visitor chooses French (CA) from the language selector, but there is no content for that page in French (CA), content in the fallback language—say, French (FR)—is displayed instead.

Create a Multilingual Site

Creating a multilingual site is a multistep process. It involves defining the languages you want your site to support, adding translated content for each language, and letting your site visitors choose their preferred language.

Setting the Default Language

The default language is the language in which your site initially displays. By default, it's set to English (US), and it serves as the starting point when you add new languages.

Adding Languages

Add the languages you want your site to support.

Setting Language Options

After you add site languages in the Languages view, you can define separate settings for each one.

About Editing Language Content

The default language is the language in which your site initially displays. By default, it's set to English (US), and it serves as the starting point when you add new languages. So for example, if you build your site and then add French as a site language, the French version initially contains English (US) content until you replace it with French-specific content.

Edit Language Content on the Page

The Site.com Studio language selector lets you switch between languages as you edit content on each page. Contributors, designers, and site administrators can each add language-specific content to a page using the language selector.

Language-Aware Properties

Several properties in the Properties pane are *language aware*, meaning they can store different values for each language.

Exporting Language Content

Export one or more site languages as an .xml file that you can send to your translation service.

Importing Translated Content

You can import a translated .xml file back into your site after your translation service has completed the translations.

Adding a Language Selector Page Element

The language selector lets your site visitors choose their preferred language when viewing a site.

Deleting Languages

When you delete a language, the translated content is not actually deleted—it's just no longer available to your or your team. As soon as you add the language back to the site, you can access the translated content again.

See Also

[Create a Multilingual Site](#)

[About Editing Language Content](#)

Create a Multilingual Site

Creating a multilingual site is a multistep process. It involves defining the languages you want your site to support, adding translated content for each language, and letting your site visitors choose their preferred language.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To add and manage language options:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To edit language content:

Site.com Publisher User or Contributor User field enabled on the user detail page and any role assigned at the site level

1. Set the default site language. It's important that you set a default language before you add translated content to your site.
2. Add languages to the site.
3. Set options for each language, such as the display language and fallback language.
4. Add content for each site language using one of these steps.
 - Edit the page content for each language directly.
 - Export the content for translation and then import the translated content.
5. Add a Language Selector page element to your site pages, so authenticated site users can choose their preferred language.
6. If you have a self-service community built on the Customer Service template, add the Language Selector component to your community pages. This component lets guest users (users who aren't logged in) choose their preferred language.



Tip After you add language-specific content to your site, you can share a separate preview URL for each language. Switch to the desired language in the Site.com Studio language selector and click **View Anonymous Preview**. Then copy the link to send it to your reviewers.

See Also

[Multilingual Sites Overview](#)

[About Editing Language Content](#)

Setting the Default Language

The default language is the language in which your site initially displays. By default, it's set to English (US), and it serves as the starting point when you add new languages.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To add and manage language options:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

It's important to set the default language before you add any language content to your site. This setting is not associated with the default language setting in your Salesforce organization.

1. Click **Site Configuration | Languages** on the Overview tab.
2. Select a language in the Default Site Language drop-down list.

 **Note** If you decide to change the default language after you add translated content, you must first export the translated content, then change the default language, and finally, import the content back into your site. Otherwise the translated content won't appear for the newly selected default language. For example, let's say you make English the default site language and add French as a site language. After you add content in both English and French, you decide to change the default site language to French. To preserve the French content, you must first export it. Then, select French as the default site language before importing the French content back into the site.

See Also

- [Adding Languages](#)
[Setting Language Options](#)

Adding Languages

Add the languages you want your site to support.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To add and manage language options:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

1. Click **Site Configuration | Languages** on the Overview tab.
2. Click **Add Languages**.
3. Select the languages you want to add to your site.
4. If necessary, reorder the list as you want it to appear in any language selector.
5. Save your changes.

See Also

[Setting the Default Language](#)

[Setting Language Options](#)

[Deleting Languages](#)

Setting Language Options

After you add site languages in the Languages view, you can define separate settings for each one.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To add and manage language options:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

1. Click **Site Configuration | Languages** on the Overview tab.
2. Select a language in Site Languages.

3. Update the options in Language Settings:

Option	Description
Active on Live Site	If you add a language selector element on your active site, this checkbox controls whether the language shows in the drop-down list. Use the property to hide a language until you're ready to release the associated content to your site visitors.
Fallback Language	The fallback language displays when there's no content available for the currently selected language. For example, if a site visitor chooses Japanese from the language selector, but there is no content for that page in Japanese, content in the fallback language is displayed instead.
Display Label	You can define the display label for each language. It appears in any language selectors you add to your site, and in the Site.com Studio language selector.

See Also

[Adding Languages](#)

[Setting the Default Language](#)

About Editing Language Content

The default language is the language in which your site initially displays. By default, it's set to English (US), and it serves as the starting point when you add new languages. So for example, if you build your site and then add French as a site language, the French version initially contains English (US) content until you replace it with French-specific content.

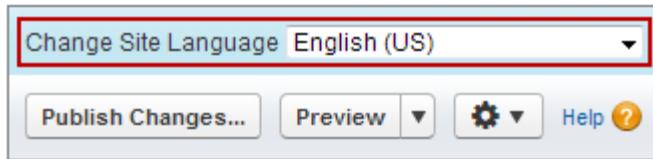
REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

After you add a language to your site, the Site.com Studio language selector appears on the toolbar. It lets you switch between languages as you edit content on each page. For example, if you add French and Spanish to your site, the language selector displays French, Spanish, and English (US).



Content blocks display icons to let you know whether content for the selected language has been replaced.

- A warning icon (⚠) indicates that the text that has not yet been replaced.
- A globe icon (🌐) indicates that the content has been replaced.

In addition, several page elements have *language aware properties*, meaning they can store different values for each language. These properties are indicated in the Properties pane by a globe icon (🌐).

You can add content for each site language one of two ways:

- [Edit the page content](#) for each language directly using the Site.com Studio language selector.
- [Export the content for translation](#) and then [import the translated content](#) back into the site.

Language Display Order

After you add language-specific content to your site, content on the page is displayed in the following order, depending on which language content is available:

1. Selected language
2. Fallback language
3. Default language

For example, if you delete the French contents of a content block, and Spanish is the fallback language, the Spanish content is displayed rather than the default language content. In turn, if you delete the Spanish content, then the default language content is displayed.

Language-Aware Page Elements

The following page elements support translated content:

- Content Block
- Custom Code
- Data Element
- Form
- Input Fields
- Image
- Language Selector

See Also

[Language-Aware Properties](#)

Edit Language Content on the Page

The Site.com Studio language selector lets you switch between languages as you edit content on each page. Contributors, designers, and site administrators can each add language-specific content to a page using the language selector.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To add and manage language options:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To edit language content:

Site.com Publisher User or Contributor User field enabled on the user detail page and any role assigned at the site level

When the page is open:

1. Select a language in the Site.com Studio language selector (the Change Site Language dropdown list on the toolbar).
2. Edit the page content.
If you're viewing a content block in a specific language:
 - A warning icon () indicates that the text that has not yet been replaced.
 - A globe icon () indicates that the content has been replaced.
3. Optionally, update the language-aware properties as required.

 **Note** Site.com Studio does not validate languages as you enter content. Take care to add the correct content for the selected language.

To revert translated content, right-click the content block and select **Revert Translation**. Text reverts to the fallback language, if specified, or else to the default site language.

See Also

[About Editing Language Content](#)

Language-Aware Properties

Several properties in the Properties pane are *language aware*, meaning they can store different values for each language.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

If you don't specify language-specific properties for a page or page element, the property values of the default languages are used instead.

Property	Applies To	Description
Alternative Text	Image	Used by screen reader users or as a substitute if the browser can't display the image. It can also help with search engine optimization (SEO).
Do Not Translate	<ul style="list-style-type: none"> • Content Block • Custom Code • Data Element • Form • Input Fields • Image • Language Selector • Page 	<p>Prevents the item's content from being translated. When selected, the globe icon () is removed from language-aware properties, indicating that they no longer support language-specific content.</p> <p>If you add language-specific content and subsequently enable this property, your language-specific content is simply hidden. If you disable the property again, the content reappears.</p>
Home Page URL	Language Selector	Redirects site visitors to a different page when they choose a language. The URL is the same for all languages.
Image Asset	Image	<p>Lets you specify the image to display for the selected language.</p> <p>For example, with English (US) selected in the Site.com Studio language selector, select the English version of an imported image. Then, to enter a French version of the same image, select French in the language selector and select the French image.</p>
Label	Language Selector	The label that appears beside the Language Selector page

Property	Applies To	Description
		<p>element when it's added to a page. "Change Site Language" is the default text.</p> <p>The text is translatable, so you can either choose a language in the Site.com Studio language selector and update the text for each language, or export all site content for translation.</p>
Navigation Name	Page	The page name that appears in a navigation menu.
Title	Page	The title that appears in the title bar of browser windows.
Visible in Live Site	Page	<p>If you add a menu to your site, controls whether the page appears in the menu for the selected language. Additionally, the page is no longer accessible via its language-specific URL.</p> <p>If Do Not Translate is also enabled, the status of Visible in Live Site applies to all languages.</p>

See Also

[Edit Language Content on the Page](#)[About Editing Language Content](#)

Exporting Language Content

Export one or more site languages as an .xml file that you can send to your translation service.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To add and manage language options:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the

USER PERMISSIONS NEEDED

site level

1. In the Languages view on the Overview tab, click | **Export Content for Translation**.
2. Select the language content that you want to export for translation. (Use CTRL+click to select multiple items.)
3. Optionally, enter a different filename. The default name is languages.xml.
4. Click **OK**.
5. If required by your browser, choose where to save the file.

After you export the .xml file, you can send it for translation.

Example The .xml file contains a time stamp attribute that records the time of the export, and encloses all translatable content in CDATA sections, as shown in this example.

```
<?xml version="1.0" encoding="utf-8"?>
<languagedata exporttimestamp="1375387451589" version="1.0">
  <component id="c5f290a5-0fb5-4679-95b7-ab27c940a9b5" context="Content Block (Content Block)">
    <field name="HTML" language="en_US" exportcrc="1113705739"><![CDATA[
      <p>This is our English content.</p>
    ]]>    </field>
    <field name="HTML" language="fr_FR" exportcrc="2453234912"><![CDATA[
      <p>This is our French content.</p>
    ]]>    </field>
  </component>
  <component id="8ef488bc-9040-49e9-9aa7-88b287cf5345" context="fffff (sites.data.vocabulary.Term)">
    <field name="TermDescription" language="en_US" exportcrc="0"><![CDATA[]]>    </field>
  </component>
</languagedata>
```

When you receive the translated .xml file, import it back into the site.

See Also

- [Importing Translated Content](#)
- [About Editing Language Content](#)

Importing Translated Content

You can import a translated .xml file back into your site after your translation service has completed the translations.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To add and manage language options:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

1. In the Languages view on the Overview tab, click  | Import Translated Content.
2. Browse to the location of the translated .xml file.
3. Select the file and click Open.
4. Decide whether to overwrite the current site content.
5. Click Import.

A message appears to indicate whether the content was imported successfully, unless the file is over 1 MB. In that case, you'll receive an email when the import process finishes.

After you import the translated content, test your pages to make sure the content displays correctly. For each page, use the Site.com Studio language selector to view the page in each supported language.

See Also

- [Exporting Language Content](#)
[About Editing Language Content](#)

Adding a Language Selector Page Element

The language selector lets your site visitors choose their preferred language when viewing a site.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To add and manage language options:

Site.com Publisher User field enabled on the user detail page

USER PERMISSIONS NEEDED

AND

Site administrator or designer role assigned at the site level

After you add languages to your site, you need to let site visitors select their preferred language from the list of languages you defined in the Languages view.

 **Tip** To save time, add the language selector to your site's page templates.

When the page is open:

1. Drag a **Language Selector** from the Page Elements pane onto the page.
Site visitors will see a Change Site Language drop-down list when they visit the page.
2. Set properties for the language selector:

Property	Description
Do Not Translate	Select this checkbox if you want the text in the Label field to remain the same for all languages.
Label	The label that appears beside the Language Selector page element when it's added to a page. "Change Site Language" is the default text. The text is translatable, so you can either choose a language in the Site.com Studio language selector and update the text for each language, or export all site content for translation.
Home Page URL	Redirects site visitors to a different page when they choose a language. The URL is the same for all languages.

See Also

[Language-Aware Properties](#)

Deleting Languages

When you delete a language, the translated content is not actually deleted—it's just no longer available to your or your team. As soon as you add the language back to the site, you can access the translated content again.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To add and manage language options:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

1. Click **Site Configuration | Languages** of the Overview tab.
2. Click  beside the language you want to delete.

See Also

[Adding Languages](#)

Content Lists and Categories Overview

Content lists let you create and store content items to use in your site, such as press releases, blog posts, and news articles. *Categories* let you create groups of terms to classify the items in your content lists.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

 **Note** Content lists and categories are currently available through a pilot program. For information on enabling content lists and categories for your organization, contact Salesforce.

Content lists and categories form part of your site's content. Combined, they integrate with [data-bound page elements](#) to let you create pages such as blogs and event lists for your site, and dynamically filter content by category or any other field.



With content lists:

- You can create custom content lists by adding the field types and field properties you need.
- You get in-context editing. There's no need to leave Site.com Studio to add or update content.
- Contributors can easily create and edit content items in the Content Lists view.
- You explicitly publish updated content lists to the live site, so changes only go live when you want them to.
- You don't need to [set access permissions](#) to allow site visitors to view content, because the content lists are part of the site.
- You don't have to [store relative URLs](#). Simply access the site's images and assets directly using [rich text fields](#).

[Creating Content Lists](#)

Create content lists to store site content such as blogs or articles that you can display dynamically on your site pages.

[Understanding the Content List's Editing View](#)

When editing a content list, you can add and edit fields, and create and manage records all in one place.

[Editing, Deleting, and Duplicating Content Lists](#)

In the Content Lists view on the Overview tab, you can manage your site's content lists.

[Adding and Editing Content List Fields](#)

As a site administrator or designer, you can add fields to a content list and edit field properties to suit your needs.

[Content List Field Types](#)

These field types are available to you when you create or edit a content list. You can add any combination of fields to a content list.

[Content List Field Properties](#)

When adding or editing fields in a content list, use the options in the Properties pane to control how the selected field functions.

[Add Records to a Content List](#)

After you've created a content list and added fields to it, you're ready to add content items.

[Editing, Deleting, and Duplicating Content List Records](#)

Each content item in a content list is stored as a record that you can edit, delete, and duplicate.

[Importing Records into Content Lists](#)

Import records from existing press releases, articles, or blogs into your site's content lists. For example, if you have a blog that's hosted elsewhere, you can export the blog's posts and then import the contents to use in your site.

[Content List XML Format](#)

To import records into a content list in Site.com Studio, you must create an XML file that consists of these elements.

[Sample Content List XML File](#)

This XML code makes up a sample XML file for importing records into a content list. In this case, the content list has two text fields called Title and Author, and a number field called Rank.

Create Categories and Terms

Categories let you create groups of terms that you can use to classify the items in your site's content lists.

Editing and Deleting Categories and Terms

In the Categories view on the Overview tab, you can update and delete your site's categories and the terms they contain.

See Also

[Creating Content Lists](#)

[Understanding the Content List's Editing View](#)

Creating Content Lists

Create content lists to store site content such as blogs or articles that you can display dynamically on your site pages.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

 **Note** Content lists and categories are currently available through a pilot program. For information on enabling content lists and categories for your organization, contact Salesforce.

1. On the Overview tab, open the Content List view and click **New Content List**.
2. Select the content list type:

Option	Description
Event List	A predefined content list that contains the following default fields:

Option	Description
	<ul style="list-style-type: none"> • Title (Text)* • Creator (Email)* • Location (Text)* • StartTime (Time)* • EndTime (Time) • Description (Text Area) <p>*Required</p>
Press Release	<p>A predefined content list that contains the following default fields:</p> <ul style="list-style-type: none"> • Title (Text)* • Author (Email)* • Body (Text Area (Rich))* • Date (Date)* <p>*Required</p>
Custom List	<p>A blank content list that you can customize to suit your needs.</p>

3. Enter a name for the content list. You can also add an optional description.
4. Click **Apply**. The content list opens in the [fields view](#).

Next, [add or edit the content list's fields](#), or add articles or posts by [creating records](#).

See Also

[Understanding the Content List's Editing View](#)

[Importing Records into Content Lists](#)

[Editing, Deleting, and Duplicating Content Lists](#)

Understanding the Content List's Editing View

When editing a content list, you can add and edit fields, and create and manage records all in one place.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

When you create or edit a content list, it opens as a new tab. Depending on whether you select  | **Edit Fields** or  | **Edit Records** on the Overview tab, the content list opens in the [fields view](#) or the [records view](#). If the content list doesn't yet have any fields, it opens in the fields view by default.

The Fields View



In the fields view, you can:

- Use the Available Fields pane (1) to [add fields to the content list](#).
- Rearrange fields on the page canvas (2) by dragging them to the correct location.
- Use the Properties pane (3) to [modify the properties](#) of a selected field.
- Use the toolbar (4) to switch to the records view. You can't open the records view until you add fields to the content list.

The Records View



In the records view, you can:

- To [create records](#), [import records](#), and switch to the fields view, use the toolbar (1).
- To delete multiple selected records, use the **Bulk Actions** menu (2).
- To [edit](#), [delete](#), or [duplicate the record](#), use the record's  menu (3).

See Also

- [Editing, Deleting, and Duplicating Content Lists](#)
- [Content List Field Types](#)

[Content Lists and Categories Overview](#)

Editing, Deleting, and Duplicating Content Lists

In the Content Lists view on the Overview tab, you can manage your site's content lists.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

 **Note** Content lists and categories are currently available through a pilot program. For information on enabling content lists and categories for your organization, contact Salesforce.

Hover over a content list and click  to:

- Edit the content list's records.
- Edit the content list's fields.
- Change the name or description properties.
- Duplicate the content list.
- Delete the content list. The content list and any records it contains are placed in the Trash Can on the Overview tab.

 **Warning** If you delete a content list that's being used by a repeater, data table, or data function, the data-bound page element will no longer work.

See Also

[Creating Content Lists](#)

[Understanding the Content List's Editing View](#)

[Content Lists and Categories Overview](#)

Adding and Editing Content List Fields

As a site administrator or designer, you can add fields to a content list and edit field properties to suit your needs.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

To add fields to a content list:

1. In the Content List view on the Overview tab, open the content list by hovering over it and clicking | **Edit Fields**. Alternatively, if the content list is open in the records view, click **Edit Fields**.
2. Add a field from the Available Fields pane by clicking it or dragging it onto the canvas. You can't add a Category field to a content list unless you've already [added categories](#) to the site.
3. Set the field's properties using the [options in the Properties pane](#).

After you add fields to the page canvas, you can:

- Rearrange the field order by selecting fields and dragging them to the correct position.
- Delete a field by selecting it and clicking .
- Open the records view by clicking **Edit Records**.

See Also

[Content List Field Types](#)

[Add Records to a Content List](#)

[Importing Records into Content Lists](#)
[Understanding the Content List's Editing View](#)

Content List Field Types

These field types are available to you when you create or edit a content list. You can add any combination of fields to a content list.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

 **Note** Content lists and categories are currently available through a pilot program. For information on enabling content lists and categories for your organization, contact Salesforce.

Type	Description
Checkbox	Lets users set a true (selected) or false (deselected) value. To import records that have a checkbox field, you can use the following values: <ul style="list-style-type: none"> • True values (case insensitive): <code>yes</code>, <code>y</code>, <code>true</code>, <code>on</code>, <code>1</code> • False values (case insensitive): <code>no</code>, <code>n</code>, <code>false</code>, <code>off</code>, <code>0</code>
Date	Lets users pick a date from a popup calendar, or enter a date in the format <i>MM/DD/YYYY</i> .
Date and Time	Lets users pick a date and time from a popup calendar, or enter a date and time in the format <i>MM/DD/YYYY HH:MM:SS</i> .
Time	Lets users pick a time from a popup calendar, or enter a time in the format <i>HH:MM:SS</i> .
Email	Lets users enter a valid email address.
Number	Lets users enter any number. Any leading zeros are removed.
Phone	Lets users enter a phone number using the following formats: <ul style="list-style-type: none"> • 1555-555-5555 • 1-555-555-5555

Type	Description
	<ul style="list-style-type: none"> • 15555555555 • (555)-555-5555 • 555-555-5555 • 5555555555
Picklist	Lets users select a value from a list you define in the Properties pane . To import records that have picklist fields, separate picklist values by a comma.
Multi-select Picklist	Lets users select one or more values from a list you define in the Properties pane . To import records that have multi-select picklist fields, separate picklist values by a comma.
Text	Lets user enter any combination of letters, numbers, or symbols, up to 255 characters.
Text Area	Lets users enter up to 32,768 characters that display on separate lines.
Rich Text Area	Lets users format text, create bulleted and numbered lists, change paragraph indentation, and add images and hyperlinks. The maximum field size is 32,768 characters, inclusive of all formatting.
URL	Lets users enter up to 255 characters of any valid website address. Must begin with <code>http://</code> or <code>https://</code> .
Category	If you've created categories , lets users select one or more terms from the category you specify. To import records that have a category field, separate category terms by a comma.

 **Note** By default, every content list also includes an ID field that contains a unique, automatically-generated ID for each record. The ID field is only visible when working with data-bound page elements, such as data repeaters or data tables.

See Also

- [Adding and Editing Content List Fields](#)
- [Add Records to a Content List](#)
- [Understanding the Content List's Editing View](#)

Content List Field Properties

When adding or editing fields in a content list, use the options in the Properties pane to control how the selected field functions.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

 **Note** Content lists and categories are currently available through a pilot program. For information on enabling content lists and categories for your organization, contact Salesforce.

Name	Description
Name	<p>The unique internal name of the field. Supports alphanumeric characters and underscores, but no spaces or special characters.</p> <p>If you're importing records into a content list, the field name in the XML file must match this name.</p> <p>If you update the name of a category field, it doesn't change the category's name in the Categories view on the Overview tab.</p>
Default Value	<p>Sets the field's default value, which is automatically entered when the user creates a new record, unless the user overwrites the value.</p> <p>If you update an existing field's default value, existing records are not updated.</p>
Hidden Field	When selected, hides the field when the user is editing a record. For example, you may want to set the default value for a field that you don't want users to see or edit.
Label	The external name of the field. This is displayed as the column heading in the content list.
Read Only	When selected, disables the field so that the user can't modify it when editing a record. For example, you may want to set a default value for a field that you don't want users to edit, but that you do want them to see.
Required	When selected, makes the field mandatory, so the user can't submit a record without entering a

Name	Description
	value.
Unique Value	When selected, requires the value of the field to be unique for each record.
Picklist Values	<p>Applies to Picklist and Multi-select Picklist fields only. Sets the list of items to display in the picklist.</p> <p>Each picklist item consists of a Label and a Value field. The Label field is displayed in the picklist, whereas the Value field is the internal value used by the content list. In most situations, you can use the same value in both fields, unless you want to submit a different value than the one displayed to the user.</p> <p>Click  to enter values for the Label and Value fields.</p>
Allow Multiple Values	Applies to Category fields only. When selected, allows the user select more than one category term when editing a record.

See Also

[Content List Field Types](#)

[Adding and Editing Content List Fields](#)

[Understanding the Content List's Editing View](#)

Add Records to a Content List

After you've created a content list and added fields to it, you're ready to add content items.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user

USER PERMISSIONS NEEDED

detail page

AND

Site administrator or designer role assigned at the site level

 **Note** Content lists and categories are currently available through a pilot program. For information on enabling content lists and categories for your organization, contact Salesforce.

1. In the Content List view on the Overview tab, open the content list by double-clicking it, or by hovering over it and clicking  | **Edit Records**.
2. To add a content item, click **Create a Record**.
3. Complete the fields. All required fields are marked with an asterisk (*).
4. Click **Save**.

Next, display the records on a page using a [data repeater](#) or [data table](#).

 **Note** New articles or blog posts don't appear on the live site until the content list is [published](#).

See Also

- [Importing Records into Content Lists](#)
- [Editing, Deleting, and Duplicating Content List Records](#)
- [Content Lists and Categories Overview](#)

Editing, Deleting, and Duplicating Content List Records

Each content item in a content list is stored as a record that you can edit, delete, and duplicate.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

USER PERMISSIONS NEEDED

AND

Site administrator or designer role assigned at the site level

 **Note** Content lists and categories are currently available through a pilot program. For information on enabling content lists and categories for your organization, contact Salesforce.

When the content list is [open](#), hover over a record and click  to:

- Edit the record. Alternatively, double-click the record.
- Delete the record.
- Duplicate the record.

To delete multiple records, select the checkbox beside each record that you want to remove, select Delete in the Bulk Actions drop-down list and click **Apply**.

 **Note** If you delete a record, it's deleted permanently from the site. However, if you delete an entire content list, the content list and its records are placed in the Trash Can on the Overview tab.

See Also

[Understanding the Content List's Editing View](#)

[Add Records to a Content List](#)

[Creating Content Lists](#)

Importing Records into Content Lists

Import records from existing press releases, articles, or blogs into your site's content lists. For example, if you have a blog that's hosted elsewhere, you can export the blog's posts and then import the contents to use in your site.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user

USER PERMISSIONS NEEDED

detail page

AND

Site administrator or designer role assigned at the site level

-  **Note** Content lists and categories are currently available through a pilot program. For information on enabling content lists and categories for your organization, contact Salesforce.

To import records into a content list, you must first create an XML file that conforms to the format described in [Content List XML Format](#).

 **Note**

- Ensure that the field names in the XML file correspond to the field names in the content list, and that the field values correspond to the [content list's field types](#). For example, if your content list has a required number field called Employee ID, the XML file must contain a corresponding `Employee ID` entry with a number value, such as `<field name="Employee ID">5228</field>`.
- The maximum file size you can upload is 50 MB.

To import an XML file:

1. On the Overview tab, open the content list that you want to import records into.
2. Click **Import Records**.
3. Browse to the XML file and click **Open**.
4. Click **Import**. A message is displayed to indicate whether the import was successful. Common reasons for failure include mismatched field types or special characters.



Tip Special characters such as `<` and `&` are illegal in XML elements. To ensure any text values don't cause a parser error, you can wrap them inside a `<! [CDATA[Your text value]]>` section. For example:

```
<field name="Title"><! [CDATA[Company News & Highlights]]></field>
```

See Also

[Sample Content List XML File](#)

[Content List Field Properties](#)

[Add Records to a Content List](#)

Content List XML Format

To import records into a content list in Site.com Studio, you must create an XML file that consists of these elements.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

 **Note** Content lists and categories are currently available through a pilot program. For information on enabling content lists and categories for your organization, contact Salesforce.

records

This element defines of a collection of records for a single content list. A `<records>` element consists of one or more `<record>` elements.

record

This element represents a grouping of related fields that comprise a single record in a content list. A `<record>` element belongs to a `<records>` element, and consists of one or more `<field>` elements. You can have multiple `<record>` elements in a `<records>` element.

field

This element represents a single field in a record. You can have multiple `<field>` elements in a `<record>` element.

Attribute:

Name	Type	Required	Description
Name	String	Yes	The internal name of the field as defined in the content list. (Not the label.) The name must match the field name in the content list exactly.

Value:

The field value must be placed between the opening and closing tags of the `<field>` element, and the format must match the field type as defined in the content list. For example, if your content list has a required number field called **Employee ID**, the XML file must contain a corresponding `Employee ID` entry with a number value, such as `<field name="Employee`

`ID">5228</field>`. See [Content List Field Types](#).

See Also

[Sample Content List XML File](#)

[Importing Records into Content Lists](#)

[Content List Field Properties](#)

Sample Content List XML File

This XML code makes up a sample XML file for importing records into a content list. In this case, the content list has two text fields called Title and Author, and a number field called Rank.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

 **Note** Content lists and categories are currently available through a pilot program. For information on enabling content lists and categories for your organization, contact Salesforce.

For more information about the required XML format, see [Content List XML Format](#).

```
<?xml version="1.0" encoding="UTF-8"?>
<records>
    <record>
        <field name="Title">Latest Company News</field>
        <field name="Author">Madison Rigsby</field>
        <field name="Rank">1</field>
    </record>
    <record>
        <field name="Title">Upcoming Developer Sessions</field>
        <field name="Author">Ian Thompson</field>
        <field name="Rank">2</field>
    </record>
</records>
```

 **Tip** Special characters such as `<` and `&` are illegal in XML elements. To ensure any text values don't cause a parser error, you can wrap them inside a `<! [CDATA[Your text value]]>` section. For example:

```
<field name="Title"><! [CDATA[Company News & Highlights]]></field>
```

See Also

- [Importing Records into Content Lists](#)
- [Content List Field Properties](#)

Create Categories and Terms

Categories let you create groups of terms that you can use to classify the items in your site's content lists.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

 **Note** Content lists and categories are currently available through a pilot program. For information on enabling content lists and categories for your organization, contact Salesforce.

You can also use terms to filter the content you display on the page. Let's say you create a category called "Cuisine" that contains the terms "Italian," Mexican," and "Japanese." Using a data-bound page element, such as a [repeater](#) or [data table](#), you can filter the results by term to display only content classified as "Italian," for example.

To create a category and add terms to it:

1. On the Overview tab, open the Categories view and click **New Category**.
2. Enter a name and description, and click **Save**.
3. In the Terms section, enter a name for the term and click **Save** or press **Enter**. If you can't see the Name field, click **New Term**.
4. Repeat for each required term.

To use the category to classify items in a content list, [add a category field](#) to the content list and select the new category. Then, when editing or creating a content list record, select the appropriate terms.

See Also

- [Editing and Deleting Categories and Terms](#)
- [Creating Content Lists](#)
- [Content Lists and Categories Overview](#)

Editing and Deleting Categories and Terms

In the Categories view on the Overview tab, you can update and delete your site's categories and the terms they contain.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

 **Note** Content lists and categories are currently available through a pilot program. For information on enabling content lists and categories for your organization, contact Salesforce.

- Hover over a category and click  to:
 - Edit the category's name and description.
 - Delete the category. The category and any terms it contains are placed in the Trash Can on the Overview tab. If the category is used by a content list, repeater, data table, or data function, you can't remove the category until you remove all references to it.
 - Add a term.
- Hover over a term and click  to:
 - Edit the term's name.
 - Delete the term. If you delete an individual term, it's removed permanently from the site. Only categories are placed in the Trash Can.

See Also

- [Create Categories and Terms](#)
- [Creating Content Lists](#)
- [Content Lists and Categories Overview](#)

Events Overview

Events enable you to add interactive and animated effects to the pages and page elements of your website.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

When an event occurs—say, when a user clicks an element on the page or when the page loads—you can specify what action (or series of actions) it should trigger. For example, when your home page loads, you could display a news bulletin popup that fades away after several seconds. Or when the user clicks a panel, you could expand the panel to reveal additional information or alter its style.

The Events pane lists many common actions that you can trigger when an event occurs. Using the options here, you can specify that:

When *[this event]* occurs, trigger *[this action]*.

[Create an Event](#)

Use events to add interactive and animated effects to the pages and page elements of your website.

[Available Events and Actions](#)

Choose from several event triggers and actions when you create an event.

See Also

- [Create an Event](#)
- [Available Events and Actions](#)
- [Using Site.com Studio as a Site Administrator or Designer](#)

Create an Event

Use events to add interactive and animated effects to the pages and page elements of your website.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Tip

- When you add an event to a page or page element, an asterisk (*) appears beside the event in the Events pane.
- If you hover over  on a selected page element, a tooltip appears indicating which events are associated with the element. You can also click the icon to quickly open the Events tab.

1. Select the relevant page or page element.
2. Select an event in the Events pane (). See [Available Events and Actions](#).
3. Click  and select an action in the Choose an Action list that appears.
4. Set the other available properties for the action, such as:
 - Target Element—specifies the page element that the action affects
 - Effect—specifies how the action is animated, such as fade or slide.
 - Speed—sets the speed of the animation to fast, normal, slow, or very slow
 - Chained—allows you to chain actions so they occur sequentially. For example, to create a Delay action that delays the action that *follows* it, select the Delay action's **Chained** checkbox. This indents the subsequent action underneath the Delay action, indicating that it's tied to the Delay action.
Applies to the Animate, Delay, Hide Element, Repeat, Show Element, and Toggle Element actions.
5. Click **Save**.
6. Add more actions if necessary.

To delete an action, select it and click .

To change the order in which an action occurs, select it and click  or .

See Also

[Add Pagination to Data Repeaters and Data Tables](#)

[Events Overview](#)

Available Events and Actions

Choose from several event triggers and actions when you create an event.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

When This Event Occurs...

Event	Occurs When...
Double click	The user double-clicks the page element.
Click	The user clicks the page element.
Focus	The focus moves to the page element.
Load	The page or page element is loaded in a browser window.
Blur	The focus moves from the page element.
Mouse in	The user moves the mouse pointer over the page element.
Mouse out	The user moves the mouse pointer out of the page element.

Trigger This Action...

Action	Description
Add CSS Class	Dynamically adds a CSS class to style the targeted item. For example, to alter the appearance of a page element, you can add a CSS class to it.
Alert	Displays a popup browser alert message.
Animate	Animates CSS properties, such as Top, Left, Width, and Height, which you specify by entering appropriate values in the CSS field. For example, if targeting an image, you can enter values such as <i>opacity: "0.4", width: "70%"</i> , which changes the image's appearance according to the speed and effect you set.
Delay	Adds a delay (measured in milliseconds) before the action that follows. (Ensure

Action	Description
	you select the Chained checkbox to tie it to the subsequent action.)
Execute JavaScript	Runs custom JavaScript code, which you enter by clicking Edit Script to open the Custom Code Editor.
Go To Page	Goes to the designated page number in data repeaters and data tables. See Adding Pagination to Data Repeaters and Data Tables .
Hide Element	Hides the targeted item according to the speed and effect you set.
Next Page	Goes to the next page in data repeaters and data tables. See Adding Pagination to Data Repeaters and Data Tables .
Previous Page	Goes to the previous page in data repeaters and data tables. See Adding Pagination to Data Repeaters and Data Tables .
Remove CSS Class	Removes a CSS class from the targeted item to dynamically remove its style. For example, to alter the appearance of a page element, you could remove the CSS class associated with it and replace it with another.
Repeat	Repeats the action that follows by the specified number of times, with the specified delay between each occurrence. (Ensure you select the Chained checkbox in the Properties pane to tie it to the following action.)
Set Element Attribute	Dynamically sets the specified attribute value of the targeted item. For example, if targeting an image, you could change the image source by entering <code>src</code> in the Attribute Name field and entering the image URL in the Attribute Value field. You can also add custom name/value pairs for advanced coding purposes.
Show Element	Reveals the targeted item according to the speed and effect you set.
Submit	Submits the selected form's data.
Toggle Element	Switches the visibility of the targeted element according to the speed and effect you set.

See Also

[Add Pagination to Data Repeaters and Data Tables](#)

[Events Overview](#)

The Contributor's Page Editing View

Use the Overview tab to import assets, preview the page, and update the appearance of the page.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To edit only content in Site.com sites:

Site.com Contributor User field enabled on the user detail page

AND

Contributor role assigned at the site level

Open a site page on the Overview tab by double-clicking the page or hovering over it and clicking | **Edit**. The page opens as a new tab.



- Using the toolbar (1), you can:
 - Undo and redo your actions.
 - Import assets, such as images and files.
 - Preview the page in a browser window.
 - Update the appearance of the page using the Branding Editor.
- Using the Page Elements pane (2), you can drag content blocks and widgets (if available) to editable areas of the page.
- On the page canvas (3), you can edit page text and add images. If editable areas are available, you can drag page elements to the page.
- Using the live mode options (4), you can see how the page appears on various devices when the page is live.

[Create Site Pages as a Site.com Contributor](#)

If your site administrator or designer has enabled page creation, you can add pages to your site.

[Edit Content Blocks as a Contributor](#)

Content blocks contain the text of your website pages, and can also house images and hyperlinks. As a contributor, you can edit the text in editable content blocks using an inline editor. Because you're editing the text inline, you always know exactly how the finished page looks when it's live.

[Understand the Inline Editing Toolbar](#)

As a contributor, you can use the inline editor to edit any editable text areas on the page. Editable areas display a gray border when you hover over the text.

[Add Images to Text as a Contributor](#)

As a contributor, you can quickly add images to any editable text areas on the page using an inline editor.

[Attach Hyperlinks to Text and Images as a Contributor](#)

As a contributor, you can use the inline editor to quickly add hyperlinks to text or images in any

editable text areas on the page.

Add Page Elements to Pages as a Contributor

As a contributor, you can add page elements to any editable areas of a page.

See Also

[Create Site Pages as a Site.com Contributor](#)

[Using Site.com Studio as a Contributor](#)

[Understand the Inline Editing Toolbar](#)

Create Site Pages as a Site.com Contributor

If your site administrator or designer has enabled page creation, you can add pages to your site.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To edit only content in Site.com sites:

Site.com Contributor User field enabled on the user detail page

AND

Contributor role assigned at the site level

1. In the Site Pages view on the Overview tab, click **New Site Page**.
2. Enter the page name.
3. Select a template for the page.
4. Click **Create**. The site page opens.

See Also

[Using Site.com Studio as a Contributor](#)

Edit Content Blocks as a Contributor

Content blocks contain the text of your website pages, and can also house images and hyperlinks. As a contributor, you can edit the text in editable content blocks using an inline editor. Because you're editing the text inline, you always know exactly how the finished page looks when it's live.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To edit only content in Site.com sites:

Site.com Contributor User field enabled on the user detail page

AND

Contributor role assigned at the site level

Editable content blocks display a gray border when you hover over the text. You can edit the text in these areas only.

When the page is open:

1. Double-click the text area you want to edit.
The inline editor appears.
2. Add or edit text, and format it using the [inline editor](#).
Avoid applying formatting, such as different fonts or highlighting, directly to text whenever possible. Instead, it's best practice to use the paragraph and heading styles to quickly apply consistent formatting throughout the site. Using paragraph and heading styles also ensures that all page text is updated automatically if a site administrator or designer modifies these styles.
3. Add [images](#) or [hyperlinks](#) as required.

Your changes are saved automatically when you click anywhere outside the text area.

See Also

[Preview How Pages Appear on Mobile Devices](#)

[The Contributor's Page Editing View](#)

Understand the Inline Editing Toolbar

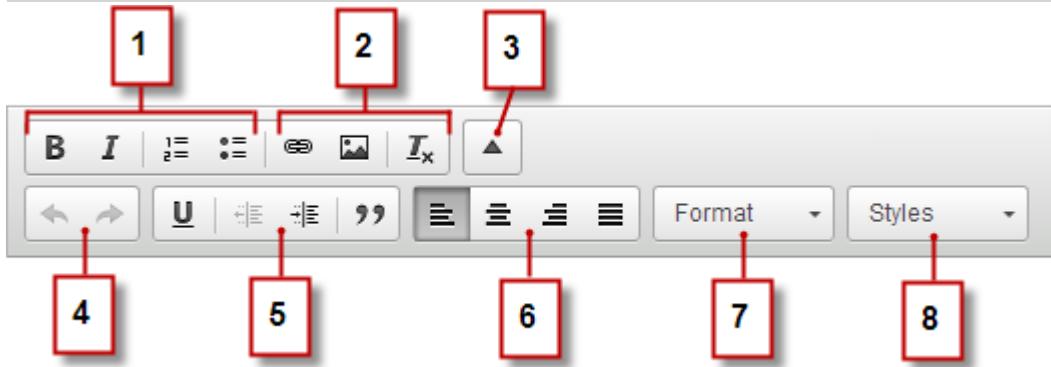
As a contributor, you can use the inline editor to edit any editable text areas on the page. Editable areas display a gray border when you hover over the text.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**



The inline editor lets you:

- Control the text style and layout (1) by:
 - Applying bold or italic formatting
 - Inserting numbered or bulleted lists
- Add [images](#) and [hyperlinks](#), and remove unwanted formatting (2).
- Expand the toolbar (3) to access additional options (4–8).
- Undo and redo your edits (4).
- Control the text style and layout (5) by:
 - Applying underline formatting
 - Setting text alignment
 - Applyingblockquote formatting
- Set paragraph indentation (6).
- Apply paragraph and heading styles (7).
- Apply additional styles, such as highlighting (8).



Tip Avoid applying formatting, such as different fonts or highlighting, directly to text whenever possible. Instead, it's best practice to use the paragraph and heading styles to quickly apply consistent formatting throughout the site. Using paragraph and heading styles also ensures that all page text is updated automatically if a site administrator or designer modifies these styles.

See Also

- [Edit Content Blocks as a Contributor](#)
- [Preview How Pages Appear on Mobile Devices](#)
- [The Contributor's Page Editing View](#)

Add Images to Text as a Contributor

As a contributor, you can quickly add images to any editable text areas on the page using an inline editor.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To edit only content in Site.com sites:

Site.com Contributor User field enabled on the user detail page

AND

Contributor role assigned at the site level

Editable text areas display a gray border when you hover over the text. You can add images to these areas only.

When the page is open:

1. Double-click the text area you want to edit.
The inline editor appears.
2. Position your cursor where you want to insert the image and click .
3. In the Image Properties dialog box, enter a URL to an image in the URL field.
For example, to add an image that you uploaded to the site, type / followed by the image filename, such as /myimage.png.
4. Enter a brief description of the image in the Alternative text field. If the browser can't display the image, the description is used by screen reader users or as a substitute. It can also help with search engine optimization (SEO).
5. Optionally, preview how the image appears in relation to the text on the page and set:
 - The width and height of the image
 - The image border (for example, to set a border that's 10 pixels wide, enter 10 in the Border field)
 - How much space surrounds the image (which is controlled by the HSpace and VSpace properties)
 - How the image aligns with the text on the page
6. Click **OK**.

Your changes are saved automatically when you click anywhere outside the text area.

See Also

- [Edit Content Blocks as a Contributor](#)
- [Understand the Inline Editing Toolbar](#)
- [Preview How Pages Appear on Mobile Devices](#)
- [The Contributor's Page Editing View](#)

Attach Hyperlinks to Text and Images as a Contributor

As a contributor, you can use the inline editor to quickly add hyperlinks to text or images in any editable text areas on the page.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

Site.com Contributor User

AND

Contributor role assigned at the site level

Editable text areas display a gray border when you hover over the text. You can add hyperlinks in these areas only.

When the page is open:

1. Double-click the text area you want to edit. Editable text areas display a gray border when you hover over the text. The inline editor appears.
2. Select the text or image that you want to attach a hyperlink to and click .
3. Select the link type.
 - To link to a page or item in your site, or to an external page:
 - Select **URL**.
 - Select the protocol.
 - In the URL field, select an item in your site, or type the address to an external page—for example, `http://www.externalsite.com`. (If you can't see the list of site items, place your cursor in the URL field and press the DOWN key on your keyboard.)
 - To link to an anchor that was previously added to the page, select **Link to anchor in the text** and select the anchor in the dropdown list.

- To link to an email message, select **An email**, and enter the recipient's email address and the message information.
4. Click **OK**.

Your changes are saved automatically when you click anywhere outside the text area.

See Also

- [Edit Content Blocks as a Contributor](#)
- [Understand the Inline Editing Toolbar](#)
- [Preview How Pages Appear on Mobile Devices](#)
- [The Contributor's Page Editing View](#)

Add Page Elements to Pages as a Contributor

As a contributor, you can add page elements to any editable areas of a page.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To edit only content in Site.com sites:

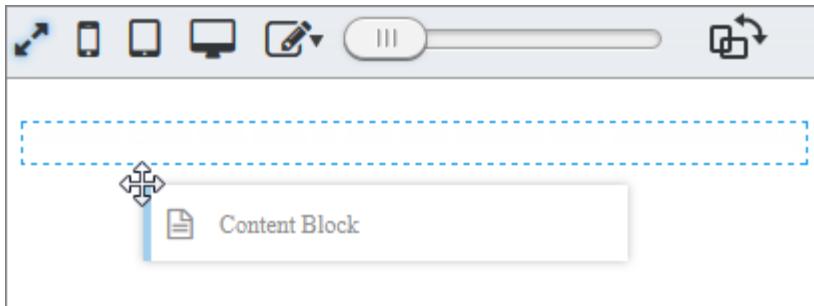
Site.com Contributor User field enabled on the user detail page

AND

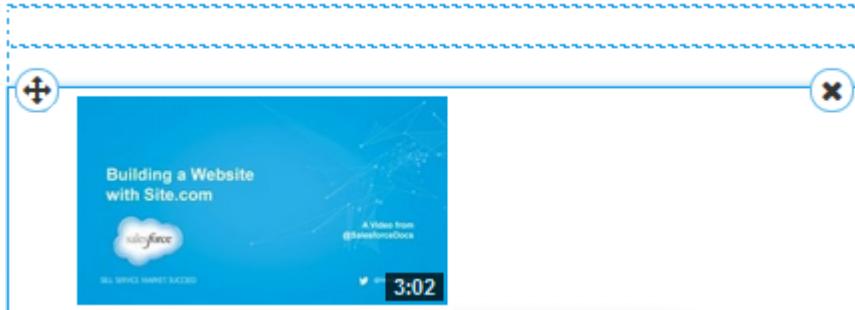
Contributor role assigned at the site level

Page elements are the building blocks of site pages. Content blocks contain the text of your website pages and can house images and hyperlinks. Widgets are custom page elements that are created by your designer or site administrator. If you can't see widgets, they're not available to you.

- To add a page element to an open page, drag the element from the Page Elements pane to an editable area. When you drag a page element to an editable page area, the area is highlighted with a blue border. If you can't see a blue border, the page doesn't have any editable areas.



- To move a page element on the page, drag the page element to another editable area.



- To delete a page element, click .

After you add a content block to a page, double-click the content block to edit it.

When you add a widget to the page, you need to specify properties for the widget in a dialog box, which automatically saves your changes. To dismiss the dialog box, click another area of the screen. For more information about the properties, contact your designer or site administrator.

See Also

- [Edit Content Blocks as a Contributor](#)
- [The Contributor's Page Editing View](#)
- [Preview How Pages Appear on Mobile Devices](#)

Preview How Pages Appear on Mobile Devices

With live mode, site administrators, designers, and contributors can preview how pages and templates appear on devices such as mobile phones and tablets.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

Site.com contributors are automatically placed in live mode for all content editing. They can edit the content blocks they've been given access to by the site administrators.

Site administrators and designers can switch between live and design modes by clicking **Live** or **Design** on the Site.com Studio toolbar. Design mode is the default view for site administrators and designers.

- To preview how the page looks:
 - On mobile phones, click .
 - On tablets, click .
 - On computer screens, click .
 - At 100% width and height, click .
- To manually adjust the width of the screen, move the slider .
- To resize the width and height of the screen, drag the resizing handles at the edge of the frame to the required size.
- To store frequently used custom screen sizes, click  and enter the custom measurements.
- To rotate the screen orientation from portrait to landscape, click .

See Also

[Edit Content Blocks as a Contributor](#)

[The Contributor's Page Editing View](#)

[Edit Site.com Pages as a Designer or Site Administrator](#)

Preview Site.com Sites

Contributors, designers, and site administrators can preview site pages to see how they look when rendered in a browser window. It's always a good idea to make sure your changes are displayed correctly, as this preview is how the pages appear on the live site.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To build, edit, and manage Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator or designer role assigned at the site level

To edit only content in Site.com sites:

Site.com Contributor User

AND

Contributor role assigned at the site level

If you're a site administrator or designer, you can also create an anonymous preview URL that allows other users to review the site before it goes live. The URL is always valid (unless you disable it) and shows the latest work in progress. It's only available to the people you send it to, and can't be found by search engines.

1. Click **Preview Page** on the toolbar when editing a page.
2. Hover over a page in the Site Pages view of the Overview tab and click | **Preview** to view a single page.
3. Click **Preview** on the toolbar of the Overview tab to view the entire site. From the **Preview** menu, you can also:
 - Click **Preview Site in a New Tab** to view the site in a new tab in the existing browser window.
 - Click **Preview Site at 1024 x 768** to view the site as it appears to laptop users.
 - Click **Enable Anonymous Preview**, if you're a site administrator or designer, to create a URL that allows other users to preview the site before it goes live. Click the **View Anonymous Preview** option that appears in the **Preview** menu to access the preview URL, which you can copy and send to other users to review and test your changes. **Enable Anonymous Preview** is also available in the [Site Configuration view](#).
4. Click **Preview** beside a site on the Site.com home page to view the entire site.

When you preview pages, all browser-related functions work, too.

Note During preview only, style sheets are rendered as inline styles.

See Also

- [Using Site.com Studio as a Site Administrator or Designer](#)
- [Using Site.com Studio as a Contributor](#)

Site.com IP Restrictions Overview

Every computer has a unique IP address that it uses to identify itself. Using IP restrictions, you can define

a range of permitted IP addresses for the pages, folders, and assets in your site to control visitors' access.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

For example, let's say you have a site page that lists all employees by department. You don't want people outside your organization to view this sensitive information. By restricting the permitted IP addresses to your organization's IP range—say, 112.122.0.1 to 112.122.0.123—you ensure that no other site visitors can view the page.

When you define IP restrictions for:

- A parent page template, all child page templates and template-based pages inherit the restrictions.
- A folder, all subfolders and assets within the folder inherit the restrictions.
- The entire site, all site items inherit the restrictions.

For any item that inherits IP restrictions, you can add additional IP restrictions to further narrow the permitted IP range.

And if a user is denied access to a page, you can redirect them to another page, such as a user-friendly error page.

[Add IP Restrictions in Site.com](#)

Control site visitors' access to the pages, page templates, folders, and assets in your site by setting the range of permitted IP addresses.

[Editing, Disabling, and Deleting IP Restrictions in Site.com](#)

After you create an IP restriction for a Site.com site, you can edit the address range, briefly disable the IP restriction (say, to allow temporary access to a page), or delete the restriction entirely.

See Also

[Add IP Restrictions in Site.com](#)

[Editing, Disabling, and Deleting IP Restrictions in Site.com](#)

Add IP Restrictions in Site.com

Control site visitors' access to the pages, page templates, folders, and assets in your site by setting the range of permitted IP addresses.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To add and edit IP restrictions in Site.com:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator role assigned at the site level

Set IP restrictions to manage access your Site.com content.

1. On the Overview tab, either:
 - Hover over the page, page template, folder, or asset in the All Site Content view and select  | **Add IP Restrictions**.
 - Select **Site Configuration | IP Restrictions** and select **Add IP Restrictions**.
2. If not already selected, choose the item you want to restrict access to.
3. Enter the first and last IP addresses of the permitted IP range—for example, 112.122.0.1 to 112.122.0.123. To enter a single IP address, complete the **Start Address** field.
Both IP addresses in a range must be either IPv4 or IPv6. In ranges, IPv4 addresses exist in the IPv4-mapped IPv6 address space ::ffff:0:0 to ::ffff:ffff:ffff, where ::ffff:0:0 is 0.0.0.0 and ::ffff:ffff:ffff is 255.255.255.255. A range can't include IP addresses inside of the IPv4-mapped IPv6 address space if it also includes IP addresses outside of the IPv4-mapped IPv6 address space. Ranges such as 255.255.255.255 to ::1:0:0:0 or :: to ::1:0:0:0 aren't allowed.
4. Select **Add IP Range** to add additional ranges.
5. To redirect users with an invalid IP address to an alternative page, such as a user-friendly error page, specify the page in Access Denied Page. If you set this in the IP Restrictions view, the page is the default for all IP restrictions unless you override it at the item level.

You can redirect users to a page in your site or to an external site. Always use a prefix such as `http://` when entering an external URL. If you don't set an access denied page, users see a blank page that displays a default "Access to this resource is denied" message.

6. Select **Save**.

To test the IP restrictions of a page template or site page, select **Preview** when the page is open. When you're happy with your updates, [publish the site](#) to enable the restrictions.

 **Note**

- If an item inherits IP restrictions—for example, an asset in an IP-restricted folder—you can add

additional restrictions to further narrow the range. Although the item doesn't display the inherited values anywhere, the inherited IP range values do apply to the child item and only site visitors with valid IP addresses can access it.

- If you select a site page in the Access Denied Page dropdown list, users with an invalid IP address can view that page even if the entire site is restricted.
- Caching is disabled for any item that has IP restrictions. Additionally, if you update the IP restrictions of an asset, folder, or page, the system updates its URL in case proxy servers already cached the item.

See Also

[Site.com IP Restrictions Overview](#)

[Editing, Disabling, and Deleting IP Restrictions in Site.com](#)

Editing, Disabling, and Deleting IP Restrictions in Site.com

After you create an IP restriction for a Site.com site, you can edit the address range, briefly disable the IP restriction (say, to allow temporary access to a page), or delete the restriction entirely.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To add and edit IP restrictions in Site.com:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator role assigned at the site level

To view all the IP restrictions on a site, open the IP Restrictions view under Site Configuration (on the Overview tab). Alternatively, to view the IP restriction on a single item, hover over it in the All Site Content view and click  | **Edit IP Restrictions**.

When either the IP Restrictions section or the IP restrictions dialog box is open, you can:

- Edit the range of IP addresses. In the IP Restrictions view, double-click the IP address to edit the values inline.
- Disable an IP restriction by deactivating it. You can reactivate the restriction at any time. If you

- deactivate an IP restriction on an item that has several restrictions, the item's other restrictions are also deactivated.
- Delete an IP restriction. This removes the restriction entirely and allows all site visitors to access the item.

After you update the site's IP restrictions, [publish the site](#) to enable your changes.

See Also

- [Site.com IP Restrictions Overview](#)
- [Add IP Restrictions in Site.com](#)

Manage Domains in Site.com

Before you can publish your site to the Internet, you must set the site's domain information.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited Editions**

Available (with limitations) in: **Developer Edition**

USER PERMISSIONS NEEDED

To manage domains and publish Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator role assigned at the site level

To add a domain to the site's domains list, Salesforce must verify that you own the domain name. The verification method you choose depends on whether the domain name is in use.

- [Add domain names using CNAME records](#) if you're creating a domain name or using a domain name that's not currently in use.
- [Add domain names using TXT records](#) if the domain name is already in use on another website and you want to reuse it. The existing website must also use domain name system (DNS) A records.

[Add Domains Using CNAME Records in Site.com](#)

To publish a site to the Internet, you must first set its domain information. Add a domain name using a CNAME record if you're creating a domain name, or if you're adding a domain name that's currently not in use.

[Add Domains Using TXT Records in Site.com](#)

To publish a site to the Internet, you must first set its domain information. Add a domain name using a

TXT record if you're reusing a domain name that's currently in use. For example, if you're migrating from an existing site to a new Site.com site, you can transition seamlessly to the new site when you're ready to go live. This method is suitable only for existing sites that use DNS A records.

[Publish and Manage Live Sites](#)

When you publish a site in Site.com, you make its pages and assets live on the Internet so site visitors can access them.

[Publish Site Changes](#)

When you *publish* a site in Site.com, you make its pages and assets live on the Internet so site visitors can access them.

[Take a Site Offline](#)

Unpublish and delete a Site.com site.

See Also

[Publish Site Changes](#)

[Take a Site Offline](#)

Add Domains Using CNAME Records in Site.com

To publish a site to the Internet, you must first set its domain information. Add a domain name using a CNAME record if you're creating a domain name, or if you're adding a domain name that's currently not in use.

REQUIRED EDITIONS

Available in: **Salesforce Classic**

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To manage domains and publish Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator role assigned at the site level

If you don't already have a branded, custom Web address, such as `www.mycompany.com`, create one by registering through a domain name registrar.

Each domain name in Site.com must be unique. Salesforce verifies your domain name when you add it to the site's list of domains, and again when you publish the site.

 **Tip** It can take up to 48 hours for domain changes to become available on the Internet. However, you can reduce this time by lowering the time to live (TTL) value in the account management settings of your DNS provider. Because the current remaining time must expire before your new setting takes effect, you must update this value a few days before going live.

You can also add a custom path to your domain name to create a custom URL. A custom URL consists of the domain and a custom path. The same path name can be used on more than one domain, but it can't be used more than one time within the same domain. When adding a domain, the / is required for the path. It indicates the root. You can add another custom name after the /, but you must at least use the / to indicate the root. For example, if the domain name is `https://oursite.com` and the path is `/products`, the site URL is `https://oursite.com/products`. If you added the custom URL to the root, the URL is `https://oursite.com`.

 **Note** You can also manage domains and paths in Setup using Domain Management.

1. In the account management settings of your DNS provider, create a CNAME record. CNAME records must include your domain name, your 18-character organization ID, and the suffix `live.siteforce.com`. For example, if your domain name is `www.mycompany.com` and your organization ID is `00dx0000000001aaa`, then the CNAME must be `www.mycompany.com.00dx0000000001aaa.live.siteforce.com`. You can find the organization ID on the new domain page in Domain Management within Setup.
2. When your CNAME record is available on the Internet, open Site.com Studio and click **Site Configuration | Domains** on the Overview tab.
3. Enter the domain name in the text box provided—for example, `www.mycompany.com`.
4. Add an optional path name.
5. click **Add**.
6. Repeat for any additional domain names. For example, you want to include common misspellings of the domain name in case users make typing mistakes. You must create a corresponding unique CNAME record for each domain name. You can add up to ten domain names.
7. When you're ready to go live, [publish the site](#).

 **Note**

- If you have an MX record set up for your domain's mail service, the domain assigned to the MX record can't be the same as the domain you assign to your CNAME record. If you create a CNAME record using the same domain as your MX record, your mail service is disabled. For example, if you have `mydomain.com` assigned to your MX record, and you want to use it for your CNAME record, we recommend assigning `www.mydomain.com` to your CNAME record instead, and then working with your DNS provider to redirect `mydomain.com` to `www.mydomain.com`.
- When you update existing domain information, you must publish your changes for them to take effect. If you see a message that there are no changes to publish, first update a page in your site and then publish your changes.
- A records aren't supported.

See Also

- [Add Domains Using TXT Records in Site.com](#)
- [Manage Domains in Site.com](#)

Add Domains Using TXT Records in Site.com

To publish a site to the Internet, you must first set its domain information. Add a domain name using a TXT record if you're reusing a domain name that's currently in use. For example, if you're migrating from an existing site to a new Site.com site, you can transition seamlessly to the new site when you're ready to go live. This method is suitable only for existing sites that use DNS A records.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To manage domains and publish Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator role assigned at the site level

Each domain name in Site.com must be unique. Salesforce verifies your domain name when you add it to the site's list of domains, and again when you publish the site.

 **Tip** It can take up to 48 hours for domain changes to become available on the Internet. However, you can reduce this time by lowering the time to live (TTL) value in the account management settings of your DNS provider. Because the current remaining time must expire before your new setting takes effect, you must update this value a few days before going live.

1. In the account management settings of your DNS provider, create a TXT record that contains your organization's ID. To find the ID, open Site.com Studio, click **Site Configuration | Domains** on the Overview tab, and copy the organization ID displayed there.
Contact your DNS provider if you're unsure how to create a TXT record.
2. When your TXT record is available on the Internet, enter the domain name in the text box provided in the Domains view—for example, `www.mycompany.com`—and click **Add**.
3. Repeat for any additional domain names. For example, you want to include common misspellings of the domain name in case users make typing mistakes.
You can add up to ten domain names.

4. [Publish](#) your Site.com site when it's ready. It isn't yet live on the Internet.
5. To go live, in your DNS account management settings:
 - a. Create a CNAME that meets the following criteria. CNAME records must include your domain name, your 18-character organization ID, and the suffix `live.siteforce.com`. For example, if your domain name is `www.mycompany.com` and your organization ID is `00dx00000000001aaa`, then the CNAME must be `www.mycompany.com.00dx0000000001aaa.live.siteforce.com`. You can find the organization ID on the new domain page in Domain Management within Setup.
 - b. Delete the old A record, and also the TXT record that you created in step 1.

See Also

[Add Domains Using CNAME Records in Site.com](#)

[Manage Domains in Site.com](#)

Publish and Manage Live Sites

When you publish a site in Site.com, you make its pages and assets live on the Internet so site visitors can access them.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To manage domains and publish Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator role assigned at the site level

When you *publish* a site in Site.com, you make its pages and assets live on the Internet so site visitors can access them.

Before you publish your site for the first time, you must [set the site's domain information](#). If you don't set up your domain, you're prompted to do so during the publishing process. After your domain is set up, you can publish your entire site or just parts of it.

When working with a site, you can:

- [Publish site changes](#).
- Review the change history by clicking  | [View Details](#) on the Overview tab.

- Take the site offline to remove it from public view.

 **Note** You can't publish your site from the sandbox.

See Also

[Using Site.com Studio as a Site Administrator or Designer](#)

Publish Site Changes

When you *publish* a site in Site.com, you make its pages and assets live on the Internet so site visitors can access them.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise, Performance, and Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To manage domains and publish Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator role assigned at the site level

Before you publish your site for the first time, you must [set the site's domain information](#). If you don't set up your domain, you're prompted to do so during the publishing process.

You can publish your entire site or just specific items.

1. Click **Publish Changes....**
2. To publish:
 - All recent changes, ensure **Site-wide changes** is selected.
 - Specific items, select **Only selected items** and select the relevant items.
-  **Note** If you select an item that has dependencies, the dependent items are also selected. For example, if you select a page, but you changed the page and the style sheet that it relies on, the style sheet is also selected. Click **View** to see the list of dependencies. You can't deselect an item if it's a critical dependency for another selected item.
3. Click **Next** and then review the list of items to be published.
4. Click **Next** and then add a publishing note, if necessary. The note appears in the Description column of the Change History view.

5. Click Publish.

You receive an email notification when your site changes go live.



Note You can't publish your site from the sandbox. Site.com Studio's publish feature doesn't support Experience Builder sites.

See Also

[Publish and Manage Live Sites](#)

[Take a Site Offline](#)

Take a Site Offline

Unpublish and delete a Site.com site.

REQUIRED EDITIONS

Available in: Salesforce Classic

Available for purchase in: **Enterprise**, **Performance**, and **Unlimited** Editions

Available (with limitations) in: **Developer** Edition

USER PERMISSIONS NEEDED

To manage domains and publish Site.com sites:

Site.com Publisher User field enabled on the user detail page

AND

Site administrator role assigned at the site level

1. On the [Overview tab](#), click **Site Configuration | Domains**.

2. Delete all listed domains.

When you remove the last domain, the site goes offline immediately. Site visitors see a “Server not found” message or similar in their browser when they attempt to view the site.

3. In the window that appears:

- Click **OK** to unpublish the site. Unpublishing the site releases the Site.com Published Site license associated with it, so you can use it to publish another site.
- Click **Cancel** if you don't want to release the Site.com Published Site license—for example, if you plan to make the site publicly available again at a later date.

If you change your mind after clicking **Cancel**, you can release the license by clicking **Unpublish** on the Overview tab. After a site is unpublished, the **Unpublish** button is replaced with the **Publish** button.

4. If you don't plan to reuse the domain with Site.com, remove any CNAME records that point to `your_domain_name.orgID.live.siteforce.com` in the account management settings of your DNS provider.

After you unpublish a site, you can delete it from the list of sites on the [Site.com tab](#).

See Also

[Manage Domains in Site.com](#)

[Publish Site Changes](#)

Salesforce Sites

Salesforce Sites enables you to create public websites and applications that are directly integrated with your Salesforce organization—without requiring users to log in with a username and password. You can publicly expose any information stored in your organization through a branded URL of your choice. And you can make the site's pages match the look and feel of your company's brand.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

Salesforce organizations contain valuable information about partners, solutions, products, users, ideas, and other business data. Some of this information would be useful to people outside your organization, but only users with the right access and permissions can view and use it. In the past, to make this data available to the general public, you had to set up a web server, create custom web pages (JSP, PHP, or other), and perform API integration between your site and your organization. Also, if you wanted to collect information using a web form, you had to program your pages to perform data validation.

With Salesforce Sites, you no longer have to do any of those things. Salesforce Sites enables you to create public websites and applications that are directly integrated with your Salesforce organization—without requiring users to log in with a username and password. You can publicly expose any information stored in your organization through a branded URL of your choice. You can also make the site's pages match the look and feel of your company's brand. Because sites are hosted on Lightning Platform servers, there are no data integration issues. And because sites are built on native Visualforce pages, data validation on collected information is performed automatically. You can also enable users to register for or log in to an associated portal seamlessly from your public site.

 **Note** Salesforce Sites is subject to these additional [Terms of Use](#).

The following examples illustrate a few ways that you can use sites:

- Create an ideas site—Use sites to host a public community forum for sharing and voting on ideas about your company, services, or products. Ideas websites can be made public using sites.
- Publish a support FAQ—Provide helpful information on a public website where customers can search

for solutions to their issues.

- Create a store locator tool—Add a public tool to your portal that helps customers find stores in their area.
- Publish an employee directory—Add an employee directory to your company's intranet by creating a site restricted by IP range.
- Create a recruiting website—Post job openings to a public site and allow visitors to submit applications and resumes online.
- Publish a catalog of products—List all of your company's products on a public website, with model numbers, current prices, and product images pulled dynamically from your organization.

Because Salesforce Sites are served directly from the Salesforce organization, a site's availability is directly related to the organization's availability. During your organization's maintenance window for major releases, your sites are unavailable; users who try to access a site see a Lightning Platform-branded maintenance page or your custom [Service Not Available Page](#). It's a good idea to inform your site users of the release maintenance windows and related sites unavailability in advance. You can view specific maintenance windows, listed by instance, at trust.salesforce.com/trust/status/#maint.

The Salesforce Sites Domain

For each of your sites, you determine the URL of the site by establishing the site's domain name. You can choose one of the following domain options.

- Use your Salesforce Sites domain. With this option, your Salesforce Sites domain name is used for all the sites that you create. For example, your company could create one public site for partners, another for developers, and a third for support. If your company's Salesforce Sites domain is <https://MyDomainName.my.salesforce-sites.com>, those three sites can have the following URLs:
 - <https://MyDomainName.my.salesforce-sites.com/partners>
 - <https://MyDomainName.my.salesforce-sites.com/developers>
 - <https://MyDomainName.my.salesforce-sites.com/support>



Note The format of the secure URLs for your Salesforce Sites depends on the org type, edition, and whether partitioned domains are enabled. The subdomain in Developer edition orgs ends in `-dev-ed`, and sandbox URLs contain the sandbox name and the word “sandbox”. Your org's secure URL is displayed on the Login Settings page. The URL is case-sensitive. Here are the formats for the most common org types. For formats for other org types, such as demo orgs and Trailhead playgrounds, see [Partitioned Domains](#) in Salesforce Help.

Organization Type	Secure URL
Developer Edition	<code>MyDomainName-dev-ed.develop.my.salesforce-sites.com</code>
Sandbox	<code>MyDomainName--SandboxName.sandbox.my.salesforce-sites.com</code>
Production	<code>MyDomainName.my.salesforce-sites.com</code>

- After you set up your Salesforce Sites, you can serve your site content on a branded domain that you own, such as <https://www.example.com>.



Note CNAME records must include your domain name, your 18-character organization ID, and the suffix `live.siteforce.com`. For example, if your domain name is `www.mycompany.com` and your organization ID is `00dx0000000001aaa`, then the CNAME must be `www.mycompany.com.00dx0000000001aaa.live.siteforce.com`. You can find the organization ID on the new domain page in Domain Management within Setup.

Manage Salesforce Sites

Review the high-level steps to create and manage a Salesforce Site.

Enable Salesforce Sites

To use Salesforce Sites with or without a custom domain, you must first enable Salesforce Sites and register your Salesforce Sites domain. Your company's Salesforce Sites domain is used for all the sites that you create.

Set Up Salesforce Sites

Enable Salesforce Sites and create public websites and applications that are directly integrated with your Salesforce org—without requiring users to log in with a username and password.

Configure Site Caching

Caching your Salesforce Site pages, attachments, and static resources can improve page load times and site performance. It can also help you avoid reaching bandwidth and service request time limits.

Use Workflow for Salesforce Sites

With workflow for sites, you can create workflow rules that trigger email alerts when certain site-related criteria are met. For example, create a rule that sends your site administrator an email alert when bandwidth usage reaches 80% of the daily bandwidth limit or when a site's status changes from the active state.

Track Your Salesforce Sites with Google Analytics

Salesforce Sites provides out-of-the-box integration with Google Analytics. To track the usage of your sites and site pages, use Google Analytics. You can track the number of visits, the number of page views, the average time spent on the site, and more.

View Salesforce Sites History

To review the changes made to your site, view the Site History related list.

View 24-Hour Salesforce Sites Usage History

You can avoid exceeding your site usage limits. Monitor the bandwidth and request time usage for each of your sites by viewing the usage data tracked on the 24-Hour Usage History related list.

Report on Salesforce Sites

To track your site activity and usage, take advantage of the Sites Usage Reporting managed package. To help you avoid reaching monthly and daily limits for individual sites and for your org, this package includes site usage reports. You can use those reports to analyze your monthly page views, daily bandwidth, and daily service request time.

Salesforce Sites Security

Review how the Sites and Domains settings related to secure connections affect what your users see when accessing your org.

Salesforce Sites-Related Apex Methods and Expressions

Lightning Platform provides Apex methods and Salesforce Sites-related expressions to help execute flow and transaction control statements.

Salesforce Sites Considerations

Before you create and use Salesforce Sites, review considerations around App Exchange packaging, access and authentication, and potential URL conflicts between sites.

Salesforce Sites Usage and Billing

Salesforce Sites usage is governed by monthly and daily allocations. Understanding these allocations is important to the success of your sites.

Can I use the same domain name for my Salesforce Sites and my Experience Cloud Sites?

With enhanced domains, your Salesforce org's My Domain name is the subdomain for Salesforce Sites and Experience Cloud sites. If enhanced domains aren't deployed in your org, you can't use the same domain name for Salesforce Sites and Experience Cloud sites.

See Also

[Set Up Salesforce Sites](#)

[Manage Salesforce Sites](#)

[Salesforce Sites Usage and Billing](#)

Manage Salesforce Sites

Review the high-level steps to create and manage a Salesforce Site.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit Salesforce Sites:

Customize Application

OR

Create and Set Up Experiences

To get started using Salesforce Sites, from Setup, enter *Sites* in the **Quick Find** box, then select **Sites**. From this page, you can:

- Register your Salesforce Sites domain, if you have not yet done so.
- View all sites hosted on your Salesforce Sites domain.
- To create a site, click **New**.
- To change an existing site, click **Edit**.
- To change the active status of your site, click **Activate** or **Deactivate**.



Warning Be careful not to expose any information that you do not want to make public.

- To view the site's details, click the site's label.
- To set a default owner of records created by guest users, select Assign new records created by Salesforce Sites guest users to a default owner.
- To open a site in a browser, click the site's URL.

See Also

[Salesforce Sites](#)

[Create and Edit Salesforce Sites](#)

[Configure Salesforce Sites](#)

Enable Salesforce Sites

To use Salesforce Sites with or without a custom domain, you must first enable Salesforce Sites and register your Salesforce Sites domain. Your company's Salesforce Sites domain is used for all the sites that you create.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit Salesforce Sites:

Customize Application

OR

Create and Set Up Experiences

To enable Salesforces Sites and register your Salesforce Sites domain, take these steps.

1. From Setup, enter *Sites* in the **Quick Find** box, then select **Sites**.

Your Sites domain includes your org's My Domain in the format *MyDomainName.my.salesforce-sites.com*.

2. Read and accept the Sites Terms of Use by selecting the checkbox.
3. Click **Register My Salesforce Site Domain**.

After you accept the Terms of Use and register your Salesforce Sites domain, the changes related to site creation are tracked in your organization's Setup Audit Trail and the Site History related list. It may take up to 48 hours for your registration to take effect.

See Also

[Set Up Salesforce Sites](#)

[Salesforce Sites Considerations](#)[Enhanced Domains](#)

Set Up Salesforce Sites

Enable Salesforce Sites and create public websites and applications that are directly integrated with your Salesforce org—without requiring users to log in with a username and password.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit Salesforce Sites: Customize Application

To edit public access settings for Salesforce Sites: Manage Users

1. From Setup, in the Quick Find box, enter *Sites*, and then select **Sites**.
2. [Enable Salesforce Sites](#).
3. [Create and configure](#) your site.
4. Develop and use [Visualforce pages](#) for your site, use or hide [standard pages](#), and customize or replace [out-of-box error pages](#). Associate these pages with your site.
5. Choose a site template. Use the lookup field to find a template that you developed, or use the provided template. The site template provides the page layout and style sheet for your site and overrides any formatting inherited from the associated portal.
6. [Enable a portal](#) for login or self-registration, then associate it with your site.
7. Optionally, modify [public access settings](#).
8. When you're ready to make your site public, click **Activate** on the Site Details page. You can also activate your site from the Site Edit and Sites pages.

After you have set up a site, you can:

- Run reports and dashboards on your site.
- [Create workflow rules](#) that trigger email alerts when site-related criteria are met.

[Create and Edit Salesforce Sites](#)

After you enable Salesforce Sites, create a Salesforce Site. Or, update an existing site.

[Configure Salesforce Sites](#)

After you create your site, configure public access and login settings, then activate the site. Optionally, you can also create a syndication feed for your site, and assign error pages for standard errors.

[Use Administrator Preview Mode to Troubleshoot Salesforce Sites](#)

If you see errors on site pages and you can't determine the cause, use administrator preview mode to look at them in context and in greater detail.

See Also

[Manage Salesforce Sites](#)
[Configure Salesforce Sites](#)

Create and Edit Salesforce Sites

After you enable Salesforce Sites, create a Salesforce Site. Or, update an existing site.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited Editions**

USER PERMISSIONS NEEDED

To create and edit Salesforce Sites:

Customize Application

OR

Create and Set Up Experiences

 **Note** You define custom URLs from Domain Management in Setup.

1. From Setup, enter *Sites* in the Quick Find box, and select **Sites**.
2. Click **New**, or click **Edit** to modify an existing site.
3. On the Site Edit page, configure the following settings.

 **Note** Site Label, Site Name, Site Contact, and Active Site Home Page are required fields.

Field	Description
Site Label	The name of the site as it appears in the user interface.
Site Name	The name used when referencing the site in the SOAP API. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
Site Description	An optional description of the site.
Site Contact	The user who receives site-related communications from the site's visitors and Salesforce.
Default Record Owner	The user who owns all new records that unauthenticated guest users create. For considerations, especially when updating sites created before Summer '20, see Best Practices and Considerations

Field	Description
	for Using the Guest Record Default Owner in Salesforce Help.
Default Web Address	<p>The unique Salesforce Site URL for this site. Salesforce Sites provide the first part of the URL based on the domain name that you registered. You must add a unique site name to the end. Unless you configure a custom address through a domain name registrar, this URL is the one that the public uses to access your site.</p>
Active	<p>Select when you're ready to make your site available to the public. You can also activate your site from the Sites and Site Details pages. When the site is inactive, users see the page specified in the Inactive Site Home Page field.</p>
Active Site Home Page	<p>The landing page that users are directed to when this site is active. Use the lookup field to find and select the Visualforce page that you developed for your site's home page. Choose the UnderConstruction page if you want to test your site.</p> <p>For ideas sites, you must use the IdeasHome page. For answers sites, use the AnswersHome page. If you don't use a site template, your site inherits its look and feel from its associated portal.</p> <p>If you deployed a site before the Summer '10 release, you can set AnswersHome as your default landing page if you create a Visualforce page using <code><apex:page action="{!!URLFOR ('/answers/answersHome.apexp') !!}></code>.</p>
Inactive Site Home Page	<p>The landing page that users are directed to when this site is inactive. Use the lookup field to find the page that you want to use. You can, for example, select a page to indicate that the site is under construction or down for maintenance.</p>
Site Template	<p>The template that provides the page layout and style sheet for your site. The site template overrides the formatting inherited from the associated portal. Use the lookup field to find a template that you've developed, or use the provided template.</p> <p>The site template specified here is used only for Visualforce pages using the <code>\$\$Site.Template</code> expression.</p>
Site Robots.txt	<p>A file that determines which parts of your public site that web spiders and other web robots can access. Search engines often</p>

Field	Description
	<p>use web robots to categorize and archive websites. HTML tags are not allowed because they are not compatible with <code>robots.txt</code> spiders. For Visualforce pages, add <code>contentType="text/plain"</code> to the <code><apex:page></code> tag.</p> <p>This example disallows all robots from indexing all pages.</p> <pre data-bbox="649 544 1432 734"><apex:page contentType="text/plain"> User-agent: * # Applies to all robots Disallow: / # Disallow indexing of all pages </apex:page></pre>
	<p>This example allows all robots to index all pages.</p> <pre data-bbox="649 872 1432 1104"><apex:page contentType="text/plain"> User-agent: * # Applies to all robots Disallow: # Allow indexing of all pages </apex:page></pre>
Site Favorite Icon	<p>The icon that appears in the browser's address field when visiting the site. Use this field to set the favorite icon for your entire site instead of for each page. Due to caching, changes are reflected on your site one day after you make them.</p>
Analytics Tracking Code	<p>The tracking code associated with your site. Services such as Google Analytics can use this code to track page request data for your site.</p>
URL Rewriter Class	<p>An Apex class to use for rewriting URLs for your site, from Salesforce URLs to user-friendly URLs. With this class, you can make rules to rewrite site URL requests typed into the address bar, launched from bookmarks, or linked from external websites. You can also create rules to rewrite the URLs for links within site pages.</p>
Enable Feeds	<p>The option that displays the Syndication Feeds related list, where you can create and manage syndication feeds for users on your public sites. This field is visible only if you have the feature enabled for your organization.</p>

Field	Description
Clickjack Protection Level	<p>You can set the clickjack protection for a site to one of these levels:</p> <ul style="list-style-type: none"> • Allow framing by any page (no protection): The least secure level. • Allow framing of site pages on external domains (good protection): Allows framing of your site pages by pages on external domains that are added to the Trusted Domains for Inline Frames list. • Allow framing by the same origin only (recommended): The default level for sites. Allows framing of site pages by pages with the same domain name and protocol security. • Don't allow framing by any page (most protection): The most secure level, but it can cause certain pages to appear as blank pages. To avoid this issue, use the default setting instead. <p>If you select Allow framing of site or community pages on external domains (good protection), select Add Domain in the Trusted Domains for Inline Frames section, enter the domain you want to allow iframes on, and select Save.</p>
Lightning Features for Guest Users	<p>Determines whether unauthenticated guest users can view features available only in Lightning. If this setting is disabled, Lightning features don't load for those users.</p> <p>If you disable this setting in a Lightning site, unauthenticated users can't access any Lightning pages, including Login pages, Error pages, and surveys. Replace those pages with custom Visualforce pages before you disable this setting.</p>
Enable Content Sniffing Protection	<p>Prevents the browser from inferring the MIME type from the document content. It also prevents malicious files from being executed as dynamic content (JavaScript, style sheet).</p>
Enable Browser Cross-Site Scripting Protection	<p>Protects against reflected cross-site scripting attacks. When a reflected cross-site scripting attack is detected, the browser renders a blank page with no content.</p>
Referrer URL Protection	<p>When loading pages, the referrer header shows only Salesforce.com rather than the entire URL. This feature eliminates the potential for a referrer header to reveal sensitive information that could be present in a full URL, such as an org ID. This feature is supported only for Chrome and Firefox.</p>

Field	Description
Allow only required cookies for this site	The option to only allow required Salesforce-supplied cookies within a Salesforce Site. When this setting is turned off, we allow all cookie types: required, functional, and advertising.
Redirect to custom domain	<p>If an HTTPS custom domain, such as <code>https://www.example.com</code>, serves this site, redirects requests from the site's system-managed URLs to that custom domain. System-managed site URLs end in <code>*.force.com</code>, <code>*.my.salesforce-sites.com</code>, or <code>*.my.site.com</code>.</p> <p>If multiple custom domains serve this site, requests are routed to the site's primary custom URL only if it's an HTTPS custom domain. Otherwise, requests are redirected to the first HTTPS custom domain associated with this site, in alphanumeric order. If no HTTPS custom domain serves this site, this option has no effect.</p>
Cache public Visualforce pages	When this option is enabled, proxy servers cache the sites' publicly available pages only for unauthenticated guest users. When this setting is disabled, all of this site's Visualforce pages can be cached in the web browser for both authenticated and unauthenticated users, and each Visualforce page's <code>cache</code> attribute controls whether the page is cached in the end user's web browser. For more information, see Configure Site Caching .
Guest Access to the Support API	When this option is enabled for a Salesforce site or Experience Cloud site, unauthenticated users are allowed to access the Support API.

4. Click **Save**.

See Also

[Set Up Salesforce Sites](#)
[Configure Salesforce Sites](#)

Configure Salesforce Sites

After you create your site, configure public access and login settings, then activate the site. Optionally, you can also create a syndication feed for your site, and assign error pages for standard errors.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit Salesforce Sites: Customize Application

To edit public access settings for Salesforce Sites: Manage Users

To access this page, from Setup, enter *Sites* in the **Quick Find** box, then select **Sites**, and then in the **Sites** list, click the site name. You can do the following:

1. From Setup, in the Quick Find box, enter *Sites*, and then select **Sites**.
2. In the **Sites** list, click the site name.
3. To update the site, click **Edit**.
4. To select the pages available for your site, click **Edit** in the Site Visualforce Pages or Site Standard Pages related lists.
All pages associated with the site must be enabled.
5. To assign error pages for standard errors, such as “Authorization Required (401)” and “Page Not Found (404),” click **Page Assignment**. You can override or edit the default pages that are provided.
6. To view or edit the site’s security settings, click **Public Access Settings**. Security settings include permissions, page layouts, and more
7. To configure the login and registration settings for your site, click **Login Settings**.
Built-in login and registration logic allows users to quickly register for and log in to your portal from your public site.



Note For Experience Cloud sites, this link opens the Experience Cloud site’s Login Page.

8. Optionally, to allow your users to access your site content through a third-party reader, create a syndication feed for your site.
 - a. To enable this option, click **Enable Feeds** in the Site Detail list.
Feeds are added as a related list for your site.
 - b. In the Feeds related list, click **New**.
9. To view your site in administrator preview mode, click **Preview as Admin**.
To help you troubleshoot site issues, administrator preview mode shows the errors on each site page in context and in greater detail.



Note This feature doesn’t appear for community organizations.

10. To see existing page redirects for your site, click **URL Redirects**.
11. To activate your site, click **Activate**.
12. To deactivate an active site, click **Deactivate**
13. To see current bandwidth and service request time usage, the daily limits, and the percentage used, view the **24-Hour Usage History** related list.
14. To see the configuration changes that have been tracked for your site, view the **Site History** related list.

Define Syndication Feeds

Syndication feeds give users the ability to subscribe to changes within Salesforce Sites and receive updates in external news readers. Simply by defining a SOQL query and mapping, you can syndicate changes to public data to your end users. You can create one or more syndication feeds for your

organization's public sites or any Visualforce page. The syndication feed displays the records specified by a SOQL query. Users can subscribe to the feed and view the feed when they visit the site or page.

[Manage Salesforce Sites Login and Registration Settings](#)

Let users register for and log in to your portal from your public Salesforce Site. For example, users browsing through an ideas site can register and login directly from that site. Then, as authenticated users, those users can vote, add comments, and participate in the ideas community. When users successfully log in, they leave the public site and enter the associated portal seamlessly.

[Public Access Settings for Salesforce Sites](#)

Control what public users can do on each Salesforce Sites site.

[Salesforce Sites URL Redirects](#)

If you move or reorganize pages on your Salesforce Site, search engines can have trouble finding the new page locations. To avoid this issue, set up site URL redirects to inform users and search engines that site content has moved.

[Associate a Portal with Salesforce Sites](#)

Allow your users to register for or log into an associated portal from your site

[Manage Salesforce Site Visualforce Pages](#)

Salesforce Sites use Visualforce pages for all site and error pages. To expose a Visualforce page on your site, enable that page for your site.

[Manage Salesforce Sites Standard Pages](#)

Control the standard pages users see for your site. Salesforce Sites uses Visualforce pages for all site and error pages. Lightning Platform also provides some standard pages that you can use. Before you can use a page for your site, enable the page for the site. If a page isn't listed under Site Standard Pages, Salesforce displays an authorization required error.

[Assign Salesforce Site Error Pages](#)

Salesforce sites use Visualforce pages for site and error pages. Sample error pages use the SiteSamples static resource for their style sheet and images.

See Also

[Manage Salesforce Sites Login and Registration Settings](#)

Define Syndication Feeds

Syndication feeds give users the ability to subscribe to changes within Salesforce Sites and receive updates in external news readers. Simply by defining a SOQL query and mapping, you can syndicate changes to public data to your end users. You can create one or more syndication feeds for your organization's public sites or any Visualforce page. The syndication feed displays the records specified by a SOQL query. Users can subscribe to the feed and view the feed when they visit the site or page.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

Create, edit, or delete a feed definition:	Modify All Data
Subscribe to a feed	No special user permission required

Define a syndication feed, including what records are returned, and which data from the records is displayed:

- Name—A descriptive name for this feed, which distinguishes it from other feeds you may create. Use only letters, numbers, or the underscore character “_”. Do not use more than one underscore character in a row.
- Description—Describe the feed. For example, “Account first name, last name, and region for the last ten accounts created or edited.”
- Query—The SOQL query that defines which records are returned to the syndication feed. To ensure fast performance, some limitations on the SOQL query are imposed.
 - If the SOQL query does not specify a limit, then no more than 20 records are returned.
 - Query limits can't exceed 200 results. If you make a query with a limit beyond this number, only the first 200 records are returned.
 - If the SOQL query does not have an ORDER BY value specified, records are ordered by the **LastModifiedDate** value if there is one, or by **SystemModstamp** value if not.
 - **COUNT** is not supported.
 - Aggregate queries are not supported. For example, this query cannot be used in a syndication feeds SOQL query:

```
SELECT Name, (SELECT CreatedBy.Name FROM Notes) FROM Account
```

- You can use *bind variables*, a variable whose value is supplied in the URL.



Note The guest user must have appropriate sharing and field-level security access or you cannot save your query, because the Lightning platform verifies access and sharing before saving.

- Mapping—Because syndication feeds use the ATOM web publishing protocol, you must provide a mapping value that matches objects and fields on the returned records with ATOM constructs. Note that all values must be string literals.
You can also use *bind variables*, a variable whose value is supplied in the URL. For more information, see [ATOM-Object Mapping](#)
- Max Cache Age Seconds—Because many users may access a feed at the same time, Salesforce caches the feed data, for 3600 seconds by default. You can modify this to a minimum of 300 seconds, or for as long as you wish. Query results that are older than the time specified here are dropped, and a new query is run on the next request for that information, that is, the next time a user opens a page containing a feed that they have subscribed to.
- Active—Select this checkbox to make the feed available for subscription. Once a feed is active, users have the option of subscribing to it.

ATOM-Object Mapping

You must specify a mapping in the syndication feed definition. The mapping relates ATOM constructs such as entry title to the corresponding value in a record, for example, "Account Name." A full set of mappings represents a news feed, and the query represents the content of each news item in a feed. For example, Lead records created today or Contacts with updated Account information.

A *feed element* is the envelope for each part of a news item, and an *entry element* is the contents of the envelope.

Mapping also allows you to apply short labels to various fields for display purposes.

The following table lists each ATOM and object element and explains what values should be provided:

Feed Element	Entry Element	Description
<code>fa</code>		Required only if <code>ea</code> (entry author) is not specified. Feed author. For example, <code>fa:"Acme Feed Author Admin Mary"</code> shows the feed author as Admin Mary.
<code>fid</code>		Optional (because default value is supplied). Id of the feed. By default, this value is the public site URL. If you specify a value, it must be a valid internationalized resource identifier (<i>IRI</i>). An IRI is a URL generalized to allow the use of Unicode.
<code>fl</code>		Optional (because default value is supplied). Feed link. For example, <code>fl:"https://www.salesforce.com"</code> . News readers usually interpret this element by linking the feed title to this URL.
<code>fst</code>		Optional. Feed subtitle. For example, <code>&map=ft:"Newest Opportunities",fst:"Western Division"</code> shows the feed title Newest Opportunities and subtitle Western Division.
<code>ft</code>		Required. Feed title. For example, <code>ft:"Newest Opportunities"</code> .
	<code>ea</code>	Required only if <code>fa</code> (feed author) is not specified. Entry author. For example, <code>ea:"Account created by: " + Account.CreatedBy</code> .
	<code>ec</code>	Required. Entry content. For example, <code>ec:"description for " Name "
" Description</code> shows the value of the Name field with additional text. The output of a feed for this example resembles the following: <div style="border: 1px solid #ccc; padding: 10px; width: fit-content;">description for Ajax Industries Description</div>

Feed Element	Entry Element	Description
	ect	Optional. Entry content of type <code>text</code> , <code>html</code> , or <code>xhtml</code> . For example, <code>ect: html</code> for HTML content. Default is <code>text</code> .
	el	Optional. Entry link. Must be a valid URI. This value is usually a link to another representation of the content for the entry. For example, the link could be to a record in the Salesforce user interface. News readers usually interpret this element by linking the entry title to this URL For example, <code>el:"Account.URL"</code> .
	es	<p>Optional. Entry summary. An optional summary of the entry content. For example, <code>et: Account.Name, es: Account.Name + "'s account number, website, and description", ec: Account.AccountNumber + " " + Account.Website + " " + Account.Description</code></p> <p>If not specified, news readers display the content defined using the <code>ec</code> element.</p>
	est	Optional. Entry summary of type <code>text</code> , <code>html</code> , or <code>xhtml</code> . For example, <code>est: html</code> for HTML content. Default is <code>text</code> . Do not specify a value unless <code>es</code> has been specified.
	et	Required. Entry title, a field name. For example, <code>et:Name</code> .
	eu	<p>Optional. By default, the required ATOM element <code><updated></code> value is automatically provided by the LastModifiedDate of the main entity being queried; usually the object in the main <code>FROM</code> clause of the SOQL query. This value indicates the last time an entry or feed was modified. If you wish to change this default behavior, you can specify a different object or field's LastModifiedDate be used. For example:</p> <ul style="list-style-type: none"> Query: <code>SELECT Id, Name, MyDate__c FROM Account</code> Mapping Parameter: <code>eu: MyDate__c</code> Query: <code>SELECT Id, Lastname, Account.LastModifiedDate FROM Contact</code> Mapping Parameter: <code>eu: Account.LastModifiedDate</code>

The following example shows a valid mapping values for a syndication feed:

```
ft: "Harry Potter", et: Name, ec: "description for " Name "<br>" Description,
el: "/" Id, ect: html
```

Feeds are displayed in the guest user context of the public site where they are displayed. Therefore, if you have used custom labels to manage internationalization, and specified labels in your mapping, users see those labels displayed in the language of the guest user.

You can only use string literals in feed mapping. You cannot use, for example, date literals such as `TODAY` or `LAST_WEEK`.

After you have defined a feed, you should test it, and then make the feed active by selecting the **Active** checkbox as described above. For more information about testing, see [Testing Syndication Feeds](#).

Using Bind Variables for Queries and Mapping

You can use *bind variables* in the `WHERE` clause of the SOQL query. Bind variables must be in the following form:

```
{ !var_name }
```

The following query uses a bind variable named `accountID`.

```
SELECT Name, Description  
FROM Account  
WHERE Id = { !accountID }
```

Note that this is not the literal name of the field, but an arbitrary name. At run time, the value for `accountID` is passed to the query from the URL. This feed is accessed from the following URL and passes in the account ID as part of the query string parameter:

```
site_URL/services/xml/My'Account'Feed?accountId=0013000000BmP4x
```

You can also use bind variables for mapping values.

The following implementation details apply to the use of bind variables for queries:

- You cannot use more than 20 bind variables in a feed definition, queries and mapping combined.
- The bind variable name cannot be more than 100 characters.
- You can use a bind variable only on the right side of a filter operation to represent part of a string. Because it represents part of a string, it must be in quotes. For example, the following is a valid query:

```
SELECT Id, Name FROM Account WHERE Name = '{!myBindVariable}'
```

The following queries are not valid, because the bind variable is not in a valid position, and is not functioning as the variable for a literal string:

```
SELECT Id, Name FROM Account WHERE {!myFieldName} = 'Joe'
```

```
SELECT Id, {!myFieldName} FROM Account WHERE IsDeleted = false
```

- You cannot use a bind variable to represent a field name. This means a bind variable cannot be used on the left side of a filter operation.
- You cannot use a bind variable to change the meaning or structure of a query for security reasons. Any special characters you specify in the bind replacement value are interpreted as literal characters when the query is evaluated.

Custom Labels and Feeds

For feeds that need to be localized into different languages, you can use custom labels to define the string in multiple languages. Then in the mapping definition, you simply refer to the custom label. When a request comes in, the custom label inspects the guest user language and returns the translated text, which is used in the mapping.

Custom labels can be specified in a field with the following syntax:

```
map_element_name: "{!$LABEL.custom_label_name}"
```

To specify a custom label in a feed, build the label via Custom Labels in Setup. For easy identification, you can name the custom label after the mapping element that takes its value, for example `feedTitle` for the `ft` element. Then specify the custom label in the feed mapping.

For example, assume that you create a feed containing information on all the houses your company is trying to sell. For English users, the title of the feed should be “The Houses,” but for Spanish users, the title of the feed should be “Las Casas.” You would create a custom label, for example, `feedTitle`. In English, its value is “The Houses,” and the Spanish value is “Las Casas.” Then, in the feed mapping definition, specify the following for the feed title `fc:`:

```
ft: "{!$LABEL.feedTitle}"
```

Visualforce and Feeds

To add a feed to a Visualforce page, use the Visualforce standard HTML features. For example, assuming the Visualforce page is located in the base directory of the site, it can contain a tag like the following:

```
<A HREF="" /services/xml/theFeedName">My feed</A>
```

The text `My feed` links to the feed.

If you want to link the feed from an image, include an inline image tag similar to the following:

```
<A HREF="/services/xml/theFeedName"></A>
```

You must upload your own image.

To add the icon to the address bar, add the link tag to the `<head>` tag of the Visualforce page:

```
<link href='URI of feed'
      type='application/x.atom+xml'
      rel='feed'
      title='A nice descriptive title' />
```

About Syndication Feeds

Syndication feeds give users the ability to subscribe to changes within Salesforce Sites and receive updates in external news readers. Simply by defining a SOQL query and mapping, you can syndicate changes to public data to your end users. You can create one or more syndication feeds for your organization's public sites or any Visualforce page. The syndication feed displays the records specified by a SOQL query. Users can subscribe to the feed and view the feed when they visit the site or page.

[View Syndication Feeds](#)

View the syndication feed definition, including what records are returned, and which data from the records is displayed.

[Test Syndication Feeds](#)

Before you enable a feed for customers, test the feed definition.

About Syndication Feeds

Syndication feeds give users the ability to subscribe to changes within Salesforce Sites and receive updates in external news readers. Simply by defining a SOQL query and mapping, you can syndicate changes to public data to your end users. You can create one or more syndication feeds for your organization's public sites or any Visualforce page. The syndication feed displays the records specified by a SOQL query. Users can subscribe to the feed and view the feed when they visit the site or page.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

Create, edit, or delete a feed definition: Modify All Data

Subscribe to a feed No special user permission required

Validating Feeds Security

When a user subscribes to a feed, the information is run in a guest user context. You must ensure the guest user has access to all and only the records appropriate for a guest use before defining any queries.

To validate feeds security:

- Edit the public access setting for the site to make sure the guest user has the correct object permissions and field-level security settings.
- Create sharing rules to control what records the guest user has access to.

After adjusting public access and field-level security settings to ensure the objects you wish to include in a feed are available to the guest user, perform any of these feeds-related tasks.

- To create a feed, click **New**.
- To view the definition of an existing feed, click the feed name.
- To edit an existing feed, click **Edit**.
- To delete an existing feed, click **Delete**.
- To test the validity of a feed, click **Run Test**. If any errors exist in the query definition or mapping, error messages are displayed.

See Also

[View Syndication Feeds](#)

[Test Syndication Feeds](#)

View Syndication Feeds

View the syndication feed definition, including what records are returned, and which data from the records is displayed.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited Editions**

USER PERMISSIONS NEEDED

Create, edit, or delete a feed definition: Modify All Data

Subscribe to a feed No special user permission required

Here are the fields in the syndication field.

- Name—A descriptive name for this feed, which distinguishes it from other feeds you can create.
- Description—Describes the feed. For example, “Account first name, last name, and region for the last ten accounts created or edited.”

- Query—The SOQL query that defines which records are returned to the syndication feed. To ensure fast performance, some limitations on the SOQL query are imposed. For more information, see [Defining Syndication Feeds](#).
- Mapping—Because syndication feeds use the ATOM web publishing protocol, you must provide a mapping value that matches objects and fields on the returned records with ATOM constructs. All values must be string literals. For more information about mapping elements, see [Defining Syndication Feeds](#).
- Max Cache Age Seconds—Because many users can access a feed at the same time, Salesforce caches the feed data for 3600 seconds by default. This value can be a minimum of 300 seconds, or for as long as you wish. Query results that are older than the time specified here are dropped, and a new query is run on the next request for that information the next time a user opens a page containing a feed that they subscribe to.
- Active—Indicates whether the feed is available for subscription. After a feed is active, users can subscribe to it.

See Also

[About Syndication Feeds](#)

[Test Syndication Feeds](#)

Test Syndication Feeds

Before you enable a feed for customers, test the feed definition.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

Create, edit, or delete a feed definition:	Modify All Data
--	-----------------

Subscribe to a feed	No special user permission required
---------------------	-------------------------------------

1. After you create a feed, from Setup, in the Quick Find box, enter *Sites*, and then select **Sites**.
2. Click the site for which you've defined the feed.
Alternatively, you can navigate to the feed detail page any number of ways, including clicking the feed name from the site detail page.
3. Click **Preview** for the feed you wish to test.
4. If one or more bind variables have been used in the feed, a dialog appears. Enter a test value for each bind variable.
5. A dialog appears allowing you to create a bookmark for the feed with the bind variable values you specified. You can save the bookmark, or cancel the dialog.
6. The values returned by your feed are displayed. Verify that the results are what you expected.

See Also

[About Syndication Feeds](#)
[View Syndication Feeds](#)

Manage Salesforce Sites Login and Registration Settings

Let users register for and log in to your portal from your public Salesforce Site. For example, users browsing through an ideas site can register and login directly from that site. Then, as authenticated users, those users can vote, add comments, and participate in the ideas community. When users successfully log in, they leave the public site and enter the associated portal seamlessly.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit Salesforce Sites:

Customize Application

OR

Create and Set Up Experiences

 **Note** Only Customer Portals can be used for self-registration. Partner portals do not support self-registration. The Authenticated Website high-volume portal user license is specifically designed to be used with Salesforce sites. Because it's designed for high volumes, it should be a cost-effective option to use with Salesforce sites.

Salesforce Sites provides built-in registration and login logic. Default Lightning Platform-branded Visualforce pages are associated with registration, login, forgot password, and password changes. You can modify these pages or replace them with your own.

The following login and registration pages are provided by default:

Page Name	Description
SiteLogin	Default login page. Used to log users in to the associated portal from your Salesforce Site.
SiteRegister	Default registration page. Used to register new users for the associated Customer Portal from your Salesforce Site.
SiteRegisterConfirm	Default registration confirmation page. The page that users see on successful registration to the associated Customer Portal.

The built-in login process:

- Checks to see whether the site is enabled for logins
- Checks to see whether the user is a valid user for the site
- Allows users to reset expired passwords

The built-in registration process:

- Checks new user information against existing users for the Customer Portal associated with the site
- Checks to see if a contact already exists for the new user
- If a contact doesn't already exist, create one and associate it with the account for the site.

! **Important** When you associate a contact with a the account for the site, you must update the SiteRegisterController with the Account ID.

- Enables the Customer Portal for the new user and sends an email confirmation message
- Optionally, allows users to create passwords on the registration page, avoiding the standard email confirmation process

Note To create and enable a person account as a Customer Portal user, use the `createPersonAccountPortalUser` Apex method. To create a person account using either the default record type defined on the guest user's profile or a specified record type, use `createPersonAccountPortalUser`, then enable the person account for the site's portal. Person Accounts can only be enabled as high-volume portal users.

1. After you associate a contact with your site for the built-in registration process, update the SiteRegisterController with the Account ID.
 - a. From Setup, in the Quick Find box, enter *Apex Classes*, and then select **Apex Classes**.
 - b. Click **Edit** next to SiteRegisterController.
 - c. Find the `private static Id PORTAL_ACCOUNT_ID = '<Account_ID>';` line and insert the ID for the account that you want to associate with new users.
The line should look similar to this:

```
private static Id PORTAL_ACCOUNT_ID = '001DoooooolQpyk';
```
 - d. Save your changes.
2. Enable public login and registration for your portal.
 - a. From Setup, in the Quick Find box, enter *Sites*, and then select **Sites**.
 - b. Click the name of the site.
 - c. Click **Login Settings**.
 - d. Click **Edit**.
 - e. From the Enable Login For list, select a portal to associate with your site. The portal you choose must have the Login Enabled option selected. For Customer Portals, you must also select the Self-Registration Enabled option. Salesforce Sites leverages the following portal settings:
 - Logout URL is used if you want to take users to a specific page on logout. If this value is left blank, users are taken to the page specified in the Active Site Home Page field for your site.
 - Lost Password Template is used for the forgot password process.

- Header, Footer, Logo, and Login Message are used for the look and feel on IdeasHome and AnswersHome pages.
 - For Customer Portals:
 - New User Template is used on self-registration if a password is not provided.
 - Default New User License, Default New User Role, and Default New User Profile are used for self-registration.
- f. Select a Change Password Page.
A default page is provided, but you can select your own page instead, using the lookup field.
- g. To let authenticated users access standard Salesforce pages as allowed by their access controls, select **Allow Access to Standard Salesforce Pages**.
This setting is enabled by default. If this setting is disabled, authenticated users can't access standard Salesforce pages, even if their access controls allow it.
- h. The Secure Web Address field shows the unique Salesforce Sites URL for this site when using SSL.
- i. Save your work.

You can also enable Sites to use your identity provider for single sign-on.

See Also

[Configure Salesforce Sites](#)

Public Access Settings for Salesforce Sites

Control what public users can do on each Salesforce Sites site.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit Salesforce Sites: Customize Application

To edit public access settings for Salesforce Sites: Manage Users

To set the public access settings for your site:

1. From Setup, in the Quick Find box, enter *Sites*, and then select **Sites**.
2. Click the name of the site.
3. To open the Profile page for your site profile, click **Public Access Settings**.

From the profile page, you can view and edit profile permissions and settings. However, you can't clone or delete the profile.

From this page, you can:

- Set the object permissions for your site. You can grant Read and Create permissions on all standard

objects except products, price books, and ideas. You can also grant Read, Create, Edit, and Delete on all custom objects. All permissions that aren't set by default must be set manually.



Warning We recommend setting the default external access to Private for the objects on which you grant Read access for your site on the Sharing Settings Setup page. This approach ensures that users accessing your site can view and edit only the data related to your site. We also recommend securing the visibility of all list views. Set the visibility of your list views to **Visible to certain groups of users**, and specify the groups that you want to view this level of access. List views with visibility set to **Visible to all users** can be visible to public users of your site. To share a list view with public users, create a new public group for those users and give them visibility. If the object's sharing is set to private, public users can't see those records, regardless of list view visibility.

- Control the visibility of custom apps. If you want to expose a custom app and its associated tabs to public users, make only that app visible and make it the default to avoid exposing other pages. If any of your site pages use standard Salesforce headers, public users can see other visible applications.
- Set the login hours during which users can access the site.
- Restrict the IP address ranges from which you can access the site.



Note To set restrictions based on IP or login hours, HTTPS is required. All authenticated access requires HTTPS. Users logging into a site with a non-secure (HTTP) site URL are redirected to a secure (HTTPS) URL. The IP addresses in a range must be either IPv4 or IPv6. In ranges, IPv4 addresses exist in the IPv4-mapped IPv6 address space `::ffff:0:0` to `::ffff:ffff:ffff`, where `::ffff:0:0` is `0.0.0.0` and `::ffff:ffff:ffff` is `255.255.255.255`. A range can't include IP addresses both inside and outside of the IPv4-mapped IPv6 address space. Ranges like `255.255.255.255` to `::1:0:0:0` or `::` to `::1:0:0:0` aren't allowed.

- Enable Apex controllers and methods for your site. Controllers and methods that are already associated with your site's Visualforce pages are enabled by default.
- Enable Visualforce pages for your site. Changes made here are reflected on the Site Visualforce Pages related list on the Site Details page, and vice versa.

Salesforce Sites URL Redirects

If you move or reorganize pages on your Salesforce Site, search engines can have trouble finding the new page locations. To avoid this issue, set up site URL redirects to inform users and search engines that site content has moved.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit site URL redirects: Customize Application

To view site URL redirects: View Setup and Configuration

Consider the following while implementing site URL redirects:

- You can't redirect error pages or CSS files (files with a .css extension).
- You can have a maximum of 6,000 redirect rules across all sites.
- Query parameters in site URL redirects are matched exactly. However, you can't redirect any URLs that include the `lastMod` parameter.
- If you have URL rewriting enabled on your site, it runs after any site page redirects.
- To redirect an Experience Cloud site home page to its companion Site.com home page set the Source URL to /. That path represents the home page for the Experience Cloud site. Then set the Target URL to s, which represents the home page for the Site.com site.

To assign a redirect to a site page:

1. From Setup, in the Quick Find box, enter *Sites*, and then select **Sites**.
2. Click a site label.
3. On the site detail page, click **URL Redirects**.
4. For Source URL, specify the former page location.
The source URL must be a relative URL, and it
 - It must be a relative URL.
 - It can't contain anchors, such as #target in /siteprefix/page.html#target.
 - It can have any valid extension type, such as .html or .php.
5.  **Note** If you use sites with prefixes, manually add the prefix to the Source URL and Target URL fields. Also, if you have a root-level site and one with a prefix, and you want to redirect a page in your prefixed site, include the prefix in the redirect rule. Otherwise, Salesforce looks for the rule in your root site, resulting in a 404 error.
6. Specify the Redirect Type.
 - Permanent (301)—Select this option when you want users and search engines to update the URL in their systems when visiting the page. Users visiting a page with this redirect type are sent to the new page. With this option, your URLs retain their search engine popularity ratings. It also instructs search engines to index the new page and to remove the obsolete source URL from their indexes.
 - Temporary (302)—Select this option when you want users and search engines to keep using the original URL for the page. Search engines interpret a 302 redirect as one that could change again at any time. Although search engines index and serve the content on the new target page, they also keep the source URL in their indexes.
7. In the Target URL field, specify the new page location.
The target URL can be a relative URL or a fully qualified URL with an https:// prefix. Unlike source URLs, target URLs can contain anchors.
8. Save your changes.

The Redirect Rules section displays all URL redirect rules you've created for your site. In this section you can:

- Edit an assigned redirect rule.
- Activate or deactivate a redirect rule.
- Delete a redirect rule.

- Sort the list in ascending or descending order by clicking the corresponding column heading.

See Also

[Salesforce Sites](#)

Associate a Portal with Salesforce Sites

Allow your users to register for or log into an associated portal from your site

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit Salesforce Sites:

Customize Application

OR

Create and Set Up Experiences

 **Note** Only Customer Portals can be used for self-registration. Partner portals don't support self-registration. The Authenticated Website high-volume portal user license is designed to be used with Salesforce Sites. Because it's designed for high volumes, it can be a cost-effective option to use with Salesforce sites.

1. Enable the portal for login.
 - a. From Setup, in the Quick Find box, enter *Customer Portal Settings*, and then select **Customer Portal Settings**. Or from Setup, in the Quick Find box, enter *Partners*, and then select **Settings**.
 - b. If you have not enabled your portal, select **Enable Customer Portal** or **Enable Partner Relationship Management** and click **Save**.
 - c. Click **Edit** for the portal you want to modify.
 - d. Select the **Login Enabled** checkbox.
 - e. Select a user for the **Administrator** field.
 - f. Optionally, set the **Logout URL**.
When the Logout URL isn't set, users are taken to the site home page on logout.
 - g. Save your changes.
2. Optionally, if you use a Customer Portal, you can allow self-registration.
 - a. From Setup, in the Quick Find box, enter *Customer Portal Settings*, and then select **Customer Portal Settings**.
 - b. Click **Edit** for the portal you want to associate with your Salesforce Site.
 - c. Select **Self-Registration Enabled**.
 - d. For both the Default New User License and Default New User Profile fields, select **Customer Portal**

User.

Depending on your portal license, you may want to select a different profile for the Default New User Profile field.

- e. For the Default New User Role field, select **User**.
- f. Save your changes.



Note Consider the following when allowing self-registration: Salesforce Sites doesn't support the use of Person Accounts for self-registration. On self-registration through a site:

- Validation rules are enforced on user creation.
- Validation rules are ignored on contact creation.

3. Associate the site pages with the default portal users.

- a. From Setup, in the Quick Find box, enter *Customer Portal Settings*, and then select **Customer Portal Settings**. Or from Setup, in the Quick Find box, enter *Partners*, and then select **Settings**.
- b. Click the name of the portal that you want to associate with your site.
- c. Edit each profile associated with your portal users. Scroll down to the Enabled Visualforce Page Access section and click **Edit**. Then add the appropriate public site pages to the **Enabled Visualforce Pages** list, and save your changes.

This step allows portal users with that profile to view these pages.



Note By default, portal users can see all pages enabled for the associated public site, so you only have to enable the pages that require authentication.

4. Associate your site with the login-enabled portal:

- a. From Setup, in the Quick Find box, enter *Sites*, and then select **Sites**.
- b. Click the site label of the site you want to configure.
- c. Click **Login Settings**.
- d. Click **Edit**.
- e. From the Enable Login For dropdown list, select the name of the portal where you want to allow login.
- f. Select the Change Password Page.
- g. Save your changes.

5. For sites with Ideas, Answers, Chatter Answers, make the zone visible in the portal and enable the IdeasHome or AnswersHome page for the site.

- a. From Setup, in the Quick Find Box, search for the feature for which you want to make the zone visible, and then click that Setup page.
 - **Ideas Zones**
 - **Chatter Answers Zones**
 - **Answers Zones**
- b. Click **Edit** next to the zone you want to make public.
- c. From the Portal dropdown list, select the portal to use for this zone.
You can choose to show the zone in all portals.



Note For ideas to work with sites, the organization must have an active portal associated with that zone. Otherwise, users encounter errors.

Manage Salesforce Site Visualforce Pages

Salesforce Sites use Visualforce pages for all site and error pages. To expose a Visualforce page on your site, enable that page for your site.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit Salesforce Sites:

Customize Application

OR

Create and Set Up Experiences

Sample error pages use the SiteSamples static resource for their style sheet and images.

 **Warning** To avoid errors, don't rename or delete SiteSamples.

1. To control the pages available to all your site visitors, add or remove the page from the site.
 - a. From Setup, in the **Quick Find** box, enter *Sites*, and then select **Sites**.
 - b. Click the name of the site you want to modify.
 - c. Click **Edit** on the Site Visualforce Pages related list.
 - d. Optionally, to enable the My Profile page, use the **Add** button to add that page to your site.

The My Profile page is a Visualforce page associated with a Customer Portal or site user's profile. The My Profile page enables users logged into either your Salesforce site, or your Customer Portal from Salesforce sites, to update their own contact information. When they update that page, the corresponding portal user and contact records are updated.

The My Profile page can be enabled either for your entire site or in a more restricted way by assigning it to the site guest user profile.

The My Profile page is also delivered as part of the Customer Portal Welcome component on your home page layout.

- e. Use the **Add** and **Remove** buttons to enable or disable other Visualforce pages for your site, and then save your changes.

If a page isn't listed under Site Visualforce Pages, an authentication or page-not-found error is displayed based on the existence of the page.



Note If you select a Visualforce page for these items, that page is automatically enabled for your site: any of the lookup fields on the Site Detail page, any of the error pages, or the **Change Password Page** under login settings. If you remove a page from this list, but it's still selected in one of these places, public users can access that page. To completely remove pages from your site, disable them here and make sure that the page isn't selected in any lookup fields for your site.

2. If you don't want to enable a Visualforce page for your entire site, you can also enable pages for specific profiles.
 - a. From Setup, in the Quick Find box, enter *Profiles*, and then select **Profiles**.
 - b. Click the name of the profile that you want to edit.
 - c. In the Enabled Visualforce Page Access related list, click **Edit**.
 - d. Use the **Add** and **Remove** buttons to enable or disable Visualforce pages for this profile, and then save your changes.



Note When you name Visualforce pages hosted on force.com sites or Classic Experience Cloud sites, choose a name that is different from standard platform URLs.

Manage Salesforce Sites Standard Pages

Control the standard pages users see for your site. Salesforce Sites uses Visualforce pages for all site and error pages. Lightning Platform also provides some standard pages that you can use. Before you can use a page for your site, enable the page for the site. If a page isn't listed under Site Standard Pages, Salesforce displays an authorization required error.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit Salesforce Sites:

Customize Application

OR

Create and Set Up Experiences

1. From Setup, in the **Quick Find** box, enter *Sites*, and then select **Sites**.
2. Click the name of the site you want to view.
3. Click **Edit** on the Site Standard Pages related list.
4. Use the **Add** and **Remove** buttons to enable or disable the following standard pages for your site.
 - Home Page—The standard page associated with the Home tab (/home/home.jsp).
 - Ideas Pages—The standard pages associated with ideas. If you want to use default ideas pages (for example, IdeasHome), enable these pages.

- Answers Pages—The standard pages associated with answers. If you want to use default answers pages (for example, AnswersHome), enable these pages.
- Search Pages—The standard Salesforce search pages. To allow public users to perform standard searches, enable these pages.
- Lookup Pages—The standard Salesforce lookup pages. These pages are the windows associated with lookup fields on Visualforce pages.



Note Disable any pages that you aren't actively using in your site. Otherwise, those pages can be accessible to public users. Also, make sure to set up private sharing to restrict search and lookup access for public users.

5. Save your changes.

Assign Salesforce Site Error Pages

Salesforce sites use Visualforce pages for site and error pages. Sample error pages use the SiteSamples static resource for their style sheet and images.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit Salesforce Sites:

Customize Application

OR

Create and Set Up Experiences



Warning To avoid errors, don't rename or delete SiteSamples.

To set the error pages for your site:

1. From Setup, enter *sites* in the **Quick Find** box, then select **Sites**.
2. Click the name of the site you want to modify.
3. Click **Page Assignment** on the Error Pages related list.
4. Using the lookup fields, assign a Visualforce page or static resource for each of the standard error conditions listed:
 - Authorization Required Page—The page users see when trying to access pages for which they don't have authorization.
 - Limit Exceeded Page—The page users see when your site has exceeded its bandwidth limits.
 - Maintenance Page—The page users see when your site is down for maintenance.
 - Service Not Available—No longer applicable. This custom page was previously used when Salesforce servers were unavailable for HTTP-only requests, which are no longer supported. When Salesforce

servers are unavailable for a Salesforce Site, the Maintenance page is displayed.

 **Tip** To display a custom page when Salesforce servers are unavailable, use Experience Cloud sites.

- **Page Not Found Page**—The page users see when trying to access a page that can't be found. You can use the action attribute on an `<apex:page>` component to redirect the Page Not Found error page. Using this kind of redirect on any other error pages will redirect users to the Maintenance page.
- **Generic Error Page**—The page users see when encountering generic exceptions.

 **Note** When using static resources in a custom error page—such as a large image file or a large CSS file contained in a static resource .zip file—each individual resource must be no larger than 50 KB. Otherwise, a 404 not found error is returned for that resource.

5. Save your changes.
6. To view the associated page as it appears in a browser, on the Site Details page, click **Preview**.

 **Tip** Add the `<site:previewAsAdmin />` component right before the closing `</apex:page>` tag in your custom Visualforce error pages to view detailed site error messages in **administrator preview mode**.

For inactive sites, the default error page is the Under Construction page, and can't be overridden using page assignments. You can override the default error page by assigning a simple HTML custom Visualforce page in the Inactive Site Home Page field.

What Happened to My Custom Error Page?

In certain circumstances, when you configure your site to display a custom error page, your site displays the standard Service Not Available page instead. This change happens by design when you switch to using the Salesforce content delivery network (CDN). This change also happens for sites that switch to using enhanced domains, because the Salesforce CDN is enabled by default for enhanced domains.

If you want to use a custom-branded error page, you have two options.

- Use Experience Builder to [customize the standard Service Not Available page](#) to fit your brand (recommended).
- Use your [custom Visualforce error page](#) by uploading it as a static resource. Make sure that the static resource contains all the resources your custom error page requires, and then select it under **Workspaces | Administration | Pages | Service Not Available**.

Interested in the technical details? When you enable the Salesforce CDN, your site traffic goes through our CDN partner. When our CDN partner receives 500, 502, 503, or 504 response codes from Salesforce, it responds with a 503 error code and displays the default Service Not Available page.

Use Administrator Preview Mode to Troubleshoot Salesforce Sites

If you see errors on site pages and you can't determine the cause, use administrator preview mode to look at them in context and in greater detail.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit Salesforce Sites: Customize Application

 **Note** Administrator preview mode is available for all active sites, including those with a [branded custom Web address](#).

To access administrator preview mode:

1. From Setup, enter *Sites* in the **Quick Find** box, then select **Sites**.
2. Click the name of the site you want to preview.
3. In the Site Detail section, click the **Preview as Admin** link. A new browser window opens with a preview of your site, and the enhanced error message appears at the bottom of the page.
4. Click **Logout of Administrator Preview Mode** to clear the administrator cookie and be redirected to the site's home page.

The detailed error messages in administrator preview mode are driven by the `<site:previewAsAdmin />` component in your Visualforce error pages. Starting with Summer '10, new organizations include the `<site:previewAsAdmin />` component by default in [standard error pages](#). You must add the component manually to all custom error pages and pages from older organizations. We recommend that you add it right before the closing `</apex:page>` tag, like this:

```
<site:previewAsAdmin />
</apex:page>
```

 **Note** The `<site:previewAsAdmin />` component contains the `<apex:messages />` tag, so if you have that tag elsewhere on your error pages, you will see the detailed error message twice.

See Also

[Create and Edit Salesforce Sites](#)

Configure Site Caching

Caching your Salesforce Site pages, attachments, and static resources can improve page load times and site performance. It can also help you avoid reaching bandwidth and service request time limits.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited Editions**

To optimize content delivery to your end users, you can enable or disable caching and set the cache duration for each of your site's pages, attachments, and static resources.

Caching Behavior for Static Site Resources

Static site resources such as images, style sheets, and scripts are cached based on the [Cache Control](#) attribute on the resource. For more information, see [Define Static Resources](#) in Salesforce Help.

Caching Behavior for Attachments

For attachments stored in Experience Cloud sites and Salesforce Sites, the caching behavior varies based on the type of attachment.

Attachment Type	Caching Behavior	Caching Location
Documents and attachments	The cache expires according to maxage parameter, in seconds. For example: /servlet/servlet.FileDownload?file=<FileID>&maxage=600	End user's web browser
Images in a rich text area	The cache expires in 45 days.	Unauthenticated guest users without IP restrictions: Proxy server Guest users with IP restrictions and authenticated users: end user's web browser

Site-Level Caching Option for Public Visualforce Pages

To control whether a site's public Visualforce pages are cached, set the site-level setting **Cache public Visualforce pages**. This setting is enabled by default for new sites.

When that site-level setting is enabled:

- Proxy servers cache the publicly accessible Visualforce page responses only during page visits by unauthenticated guest users.
- The caching behavior and caching location for publicly accessible Visualforce pages differ based upon whether IP range restrictions or login hours restrictions are defined for the site's guest user.
- If caching occurs at the network level, when a page is cacheable for guest users, an unauthenticated version of that page can be served to authenticated users. Examples of options that can cache your site pages at the network level include content delivery networks (CDNs) and intercepting proxy

servers, such as a data loss prevention (DLP) proxy. This behavior can also occur when a custom domain uses the [Salesforce CDN](#) or [your HTTPS certificate](#) to serve your site.

- You can disable caching for an individual publicly accessible Visualforce page. To disable that caching, set the page's boolean `cache` attribute to `false`.

When the site-level setting is disabled:

- The proxy server doesn't cache any of the site's Visualforce pages. Instead, the site's cache-enabled Visualforce pages are cached in the web browser for both authenticated and unauthenticated users. This caching behavior matches caching for Visualforce pages served by Salesforce outside of a site.
- You can enable caching for an individual publicly accessible Visualforce page. To enable that caching, set the page's boolean `cache` attribute to `true`.

Page-Level Caching Options for Public Visualforce Pages

To control the caching behavior for your site's Visualforce pages, set the boolean `cache` attribute and the integer `expires` attribute on each page.

For example, a Visualforce page whose cache is set to expire in 15 minutes looks like this:

```
<apex:page cache="true" expires="900">
```

Caching Behavior for Visualforce Pages When Site-Level Caching Is Enabled

Here are the rules for caching when the site-level Cache public Visualforce pages setting is enabled.

- The page is cached only when the page-level `cache` attribute is `true` or when that attribute isn't set.
- The caching behavior and location also differ based upon whether IP range restrictions or login hours restrictions are defined for the site's guest user.
- When caching occurs in these cases, the cache expires based on the page's `expires` attribute. If the page's `expires` attribute isn't set, the cache expires in 600 seconds (10 minutes).

Event	Page-Level <code>cache</code> Attribute	Site Guest User Restrictions	Caching Location
An unauthenticated user visits a login-enabled site.	<code>true</code> , <code>false</code> , or not set	Yes	Not cached
	<code>true</code> or not set	No	Proxy server and end user's web browser
	<code>false</code>	No	Not cached
An authenticated user visits a login-enabled site.	<code>true</code> , <code>false</code> , or not set	Yes or No	Not cached

Event	Page-Level <code>cache</code> Attribute	Site Guest User Restrictions	Caching Location
A user visits a Salesforce Site that isn't login-enabled.	<code>true</code> or not set	Yes	End user's web browser
		No	Proxy server and end user's web browser
	<code>false</code>	Yes or No	Not cached

Caching Behavior for Visualforce Pages When Site-Level Caching Is Disabled

When the site-level Cache public Visualforce pages setting is disabled, you can enable caching for specific pages via the page-level cache attribute. When that attribute is `true`, the Visualforce page is cached in the end user's web browser and the cache expires according to the page's expires attribute. If the page's `expires` attribute isn't set, the cache expires in 0 seconds.

This behavior applies to both authenticated and unauthenticated users regardless of whether the site is login-enabled. Also, the presence of IP range restrictions or work hour restrictions on the site's guest user has no impact on this caching behavior.

Event	Page-Level <code>cache</code> Attribute	Caching Location
A user visits a site.	<code>true</code>	End user's web browser
	<code>false</code> or not set	Not cached

See Also

[Salesforce Sites Usage and Billing](#)

[Define Static Resources](#)

Use Workflow for Salesforce Sites

With workflow for sites, you can create workflow rules that trigger email alerts when certain site-related criteria are met. For example, create a rule that sends your site administrator an email alert when bandwidth usage reaches 80% of the daily bandwidth limit or when a site's status changes from the active state.

REQUIRED EDITIONS

Available in: Lightning Experience and Salesforce Classic

Available in: **Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To create or change workflow rules:

Customize Application

USER PERMISSIONS NEEDED

To create and edit Salesforce Sites: Customize Application

Site usage workflow rules can help you keep your sites from exceeding rolling 24-hour limits for bandwidth and service request time, and monthly limits for page views and logins. Workflow rules on the Site object are evaluated every hour for all sites within the organization, unless your criteria is limited to certain sites. Workflow rules that are created on the Organization and User License objects are evaluated every three hours.

Only email alert actions are supported for site usage workflow rules. Other workflow actions, such as field updates, tasks, and outbound messages, aren't available.

1. Optionally, create custom email templates using Site merge fields.
 - a. From Setup, in the Quick Find Box, enter *Email Templates*, and then select **Email Templates**.
 - b. Click **New Template**.
 - c. To use Site merge fields in your template, in the Available Merge Fields section, select **Site Fields** in the Select Field Type dropdown list .
2. [Create a workflow rule](#) for one of these objects.
 - a. For monthly site page views allowed and monthly site page views used fields, select **Organization**.
 - b. For site detail, daily bandwidth and request time, monthly page views allowed, and other fields, select **Site**.
 - c. For the monthly logins allowed and monthly logins used fields, select **User License**.
The Organization and Site objects are only available if Salesforce Sites is enabled for your organization.
The User License object isn't dependent on sites, and is only available if you have Customer Portals or partner portals enabled for your organization.
3. For the evaluation criteria, select one of these options.
 - To specify the filter criteria that a site must meet to trigger the rule, select **criteria are met**.
For example, to trigger the rule every time the active status changes for a site, set the filter to *Site Status not equal to Active*. To add more rows or to set up Boolean conditions, click **Add Filter Logic**.
 - To enter a formula, select **formula evaluates to true**.
For example, this formula triggers the rule when bandwidth usage reaches 80 percent of the daily bandwidth limit.

```
DailyBandwidthUsed >= 0.8 * DailyBandwidthLimit
```

In this example, the formula triggers the rule when time usage reaches 80 percent of the daily time limit.

```
DailyRequestTimeUsed >= 0.8* DailyRequestTimeLimit
```

4. Add an [email alert action](#) as an automated action on your workflow rule.

See Also

[Salesforce Sites](#)

Track Your Salesforce Sites with Google Analytics

Salesforce Sites provides out-of-the-box integration with Google Analytics. To track the usage of your sites and site pages, use Google Analytics. You can track the number of visits, the number of page views, the average time spent on the site, and more.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit Salesforce Sites:

Customize Application

OR

Create and Set Up Experiences

 **Note** The `<site:googleAnalyticsTracking/>` component only works on pages used in a Salesforce Site. Sites must be enabled for your organization and the **Analytics Tracking Code** field must be populated. To get a tracking code, go to the [Google Analytics](#) website.

1. Sign up for an account at [Google Analytics](#).
2. Add a profile in Google Analytics and enter the domain or full URL for the site you want to track.
3. Copy the **Web Property ID** from Google's tracking status information. Then paste that value into the **Analytics Tracking Code** field on the Site Edit page for the site that you want to track.
The Web property ID starts with the letters UA followed by your account and profile numbers. For example, UA-9049246-2.
4. Save your changes.
5. To track the Visualforce pages associated with your site, enter the following tag in the site template for those pages, or in the individual pages themselves.

```
<site:googleAnalyticsTracking/>
```

For a page to be tracked, this tag is required either in the page or the associated page template. The default site template already contains the tag, so all pages that use that template are tracked—including certain default pages.

 **Note** Google recommends adding the component at the bottom of the page to avoid increasing page load time.

6. Go to the Google Analytics site and follow their instructions for completing the process.
After you sign up, it can take up to 24 hours before you see initial tracking results in Google Analytics.

 **Tip** To track multiple sites separately, create separate profiles using the full site URLs and enter a different Web property ID in the **Analytics Tracking Code** field for each site.

See Also

- [Manage Salesforce Sites](#)
- [Create and Edit Salesforce Sites](#)
- [Report on Salesforce Sites](#)

View Salesforce Sites History

To review the changes made to your site, view the Site History related list.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit Salesforce Sites:

Customize Application

OR

Create and Set Up Experiences

1. From Setup, in the Quick Find box, enter *Sites*, and then select **Sites**.
2. Click the name of the site you want to view.
3. View the Site History related list.

The Site History related list tracks and displays the changes made to your site. These events are tracked in the site history, along with the user who made the change and the time that the change occurred.

Event	Description
Site Creation	Logs when each site was created.
Site Detail Changes	Changes to these site values are logged. <ul style="list-style-type: none">• Site Label• Site Name• Site Description• Site Contact• Default Record Owner• Default Web Address

Event	Description
	<ul style="list-style-type: none"> • Redirect to custom domain • Custom Web Address • Active Site Home Page • Inactive Site Home Page • Site Template • Site Robots.txt • Site Favorite Icon • Analytics Tracking Code • Clickjack Protection Level • URL Rewriter Class <p>Changes to these settings are logged.</p> <ul style="list-style-type: none"> • Active Status • Enable Feeds • Lightning Features for Guest Users • Enable Content Sniffing Protection • Enable Browser Cross Site Scripting Protection • Referrer URL Protection • Allow only required cookies for this site • Cache public Visualforce pages • Guest Access to the Payments API • Guest Access to the Support API
Site Standard Pages	Logs when any standard page is enabled or disabled.
Site Error Pages	Logs when any error page assignment is changed.
Login Settings Changes	<p>Changes to these login settings are logged.</p> <ul style="list-style-type: none"> • Portal • Change Password Page • Require Non-Secure Connections (HTTP) <p>This setting is removed in Winter '21 and later.</p>
URL Redirect Changes	Logs when any URL redirect is created, deleted, enabled, disabled, or changed.

See Also

[Salesforce Sites](#)

[Configure Salesforce Sites](#)

View 24-Hour Salesforce Sites Usage History

You can avoid exceeding your site usage limits. Monitor the bandwidth and request time usage for each of your sites by viewing the usage data tracked on the 24-Hour Usage History related list.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit Salesforce Sites:

Customize Application

OR

Create and Set Up Experiences

For more information on limits for your Salesforce Sites, see [Salesforce Sites Usage and Billing](#).

1. From Setup, in the Quick Find box, enter *Sites*, and then select **Sites**.
2. Click the name of the site you want to view.
3. View the 24-Hour Usage History related list.

Processing time can cause a delay of several minutes for usage information.

The 24-Hour Usage History related list tracks and displays these usage metrics for your site.

Metric	How It's Calculated
Origin Bandwidth	Bandwidth is calculated as the number of megabytes served and received from the site's origin server. The Daily Limit applies to a rolling 24-hour period.
Request Time	"Service request time" is calculated as the total server time in minutes required to generate pages for the site. The Daily Limit applies to a rolling 24-hour period.

Origin server refers to the web server that hosts your site. Rolling 24-hour period refers to the 24 hours immediately preceding the current time.

For each metric, the related list displays Current Usage, Daily Limit, and the Percent Used.

Report on Salesforce Sites

To track your site activity and usage, take advantage of the Sites Usage Reporting managed package. To help you avoid reaching monthly and daily limits for individual sites and for your org, this package includes site usage reports. You can use those reports to analyze your monthly page views, daily

bandwidth, and daily service request time.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited Editions**

USER PERMISSIONS NEEDED

To install packages: Download AppExchange Packages

To run reports: Run Reports
AND

Read on the records included in the reports

To create, edit, save, and delete reports: Run Reports and Read on the records included in the reports
AND

Create and Customize Reports

To create, edit, and delete dashboards: Run Reports
AND

Manage Dashboards

Get Started

Before you can report on the usage for your site, install the Sites Usage Reporting managed package. This managed package, available on AppExchange, contains reports and a dashboard for monitoring sites usage.

1. To find the Sites Usage Reporting managed package, go to AppExchange and search for “sites reporting,” or go to <https://appexchange.salesforce.com/listingDetail?listingId=a0N30000001SUEwEAO>.
2. Install the Sites Usage Reporting managed package.

Use Packaged Reports to Analyze Site Usage

The Sites Usage Reporting managed package contains the following reports for the sites in your organization.

You can find these reports in the Site Usage Reports folder under All Reports in the Reports tab. You can also select **Site Usage Reports** in the **Folder** drop-down list, then click **Go**.

Report	Description
Current Period Page Views	Shows the total page views for the current period (calendar month), measured against page views allowed. Page views are broken down by site and by day. The current period limit applies to all sites within the organization.
Daily Total Bandwidth Usage	Shows the total bandwidth usage over the last 30 days, broken down by site, by day, and by origin and cache servers.
Daily Total Page Views	Shows the total page views over the last 30 days, broken down site, by day, and by origin and cache servers.
Site Daily Origin Bandwidth Usage	Shows the total origin bandwidth usage over the last 30 days, broken down by site and by day.
Site Daily Request Time Usage	Shows the total origin service request time over the last 30 days, broken down by site and by day.
Top Bandwidth Consuming Sites	Shows the sites that consumed the most bandwidth during the current period.
Top Resource Consuming Sites	Shows the sites that consumed the most service request time during the current period.
Top Sites by Page Views	Shows the sites that generated the most page views during the current period.

-  **Note** Site usage data is aggregated at midnight, GMT, so the current day's page view counts may not be accurately reflected in reports, depending on your time zone. Cache server page views may take a day or more to be reflected in reports.

Create a Custom Report

To create a custom report for site usage, use the Site Usage Reports custom report type.

1. From the Reports tab, click **New Report**.
2. For the report type, select **Administrative Reports**, then **Site Usage Reports**.

To see the Site Usage Reports custom report type, ensure that sites are enabled and the Sites Usage Reporting managed package is installed.

3. Click **Create**.
Fields related to your sites, such as **Site Name**, **Site Status**, **Daily Bandwidth Limit**, and **Daily Request Time Limit** can all be used in your custom report.

 **Note** For custom reports that use the Site Usage Reports custom report type, the **Origin Bandwidth** column is measured in bytes, and the **Request Time** column is measured in milliseconds. Account for the difference in units when comparing these columns to the **Daily Bandwidth Limit** and **Daily Request Time Limit** columns, which are measured in megabytes and minutes, respectively. For the reports included with the managed package, bandwidth is measured in megabytes and request time is measured in minutes.

Use the Site Usage Dashboard to Monitor Sites

To help you monitor the sites in your organization at a glance, the Sites Usage Reporting managed package contains the Site Usage Dashboard. The dashboard contains a component for each of the reports provided in the managed package.

1. To access the dashboard, from the Dashboards tab, use the **View Dashboard** field. Or, click **Go to Dashboard List** and select **Site Usage Dashboard** from the dashboard list.
2. To modify the dashboard, click **Edit**.
You can also create your own custom dashboard that includes any custom reports you created.
Consider adding the Site Usage Dashboard as the dashboard snapshot on your home page.

See Also

[Track Your Salesforce Sites with Google Analytics](#)

Salesforce Sites Security

Review how the Sites and Domains settings related to secure connections affect what your users see when accessing your org.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited** Editions

All authenticated access requires HTTPS. Users logging into a site with a non-secure (HTTP) site URL are redirected to a secure (HTTPS) URL. To set restrictions based on IP or login hours, HTTPS is required.

Salesforce requires HTTPS connections to all sites and automatically upgrades HTTP requests. If you're using a custom domain to serve a site, to ensure connectivity, we recommend that you select one of the options to serve your domain over HTTPS. Only select the Use a temporary non-HTTPS domain option if you're configuring your domain before it can be secured with HTTPS. For example, to configure DNS or add a subdomain whose CNAME points to another service.

These behaviors and sharing settings affect users accessing sites.



- We recommend setting the default external access to Private for the objects on which you grant “Read” access for your site on the Sharing Settings Setup page. This ensures that users accessing your site can view and edit only the data related to your site.
 - We also recommend securing the visibility of all list views. Set the visibility of your list views to **Visible to certain groups of users**, and specify the groups to share to. List views whose visibility is set to **Visible to all users** may be visible to public users of your site. To share a list view with public users, create a new public group for those users and give them visibility. If the object's sharing is set to private, public users won't be able to see those records, regardless of list view visibility.
-
- For custom domains with the Use a temporary non-HTTPS domain HTTPS option selected, if users connect using HTTP instead of HTTPS, they can experience a connection timeout.
 - If a user opens a custom domain with the Use a temporary non-HTTPS domain HTTPS option selected, we attempt to redirect the user to the site's preferred HTTPS custom domain. If the site doesn't have a preferred HTTPS custom domain, the user is redirected to the org's my.salesforce-sites.com domain. In sandboxes and Developer Edition orgs, the org's my.salesforce-sites.com domain is used. For example, you registered www.example.com as an HTTP-only custom domain. When the URL is upgraded to HTTPS and no HTTPS-capable custom domains are linked to the site, the URL changes to https://MyDomainName.my.salesforce-sites.com. For more information, see Managing Salesforce Sites Login and Registration Settings.
 - Authenticated and non-authenticated users may see different error messages for certain conditions—for example, on Apex exceptions.
 - Cache settings on static resources are set to private when accessed via a Salesforce Site whose guest user's profile has restrictions based on IP range or login hours. Sites with guest user profile restrictions cache static resources only within the browser. Also, if a previously unrestricted site becomes restricted, it can take up to 45 days for the static resources to expire from the Salesforce cache and any intermediate caches.
 - Guest users aren't owners of records they create in Salesforce Sites. Instead, when a guest user creates a record in a Salesforce Site, the record's ownership is assigned to the site's default record owner.

Salesforce Sites-Related Apex Methods and Expressions

Lightning Platform provides Apex methods and Salesforce Sites-related expressions to help execute flow and transaction control statements.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To create and edit Salesforce Sites:

Customize Application

OR

Create and Set Up Experiences

Lightning Platform Apex Methods

Apex methods for Salesforce Sites are contained in the `site` class, `cookie` class, and `urlRewriter` class. See the [Lightning Platform Apex Code Developer's Guide](#).

Salesforce Sites-Related Expressions

Salesforce Sites-related expressions can be used on Visualforce pages, email templates, and s-controls.

Merge Field	Description
<code>\$Site.Name</code>	Returns the API name of the current site.
<code>\$Site.Domain</code>	Returns your Salesforce Sites based URL.
<code>\$Site.CustomWebAddress</code>	Returns the request's custom URL if it doesn't end in <code>force.com</code> or returns the site's primary custom URL. If neither exist, <code>\$Site.CustomWebAddress</code> returns an empty string. The URL's path is always the root, even if the request's custom URL has a path prefix. If the current request isn't a site request, then this field returns an empty string. This field's value always ends with a <code>/</code> character. Use of <code>\$Site.CustomWebAddress</code> is discouraged and we recommend using <code>\$Site.BaseCustomUrl</code> instead.
<code>\$Site.OriginalUrl</code>	Returns the original URL for this page if it's a designated error page for the site; otherwise, returns <code>null</code> .
<code>\$Site.CurrentSiteUrl</code>	Returns the base URL of the current site that references and links use. This field can return the referring page's URL instead of the current request's URL. This field's value includes a path prefix and always ends with a <code>/</code> character. If the current request isn't a site request, then this field returns an empty string. Use of <code>\$Site.CurrentSiteUrl</code> is discouraged. Use <code>\$Site.BaseUrl</code> instead.
<code>\$Site.LoginEnabled</code>	Returns <code>true</code> if the current site is associated with an active login-enabled portal; otherwise returns <code>false</code> .

Merge Field	Description
\$Site.RegistrationEnabled	Returns <code>true</code> if the current site is associated with an active self-registration-enabled Customer Portal; otherwise returns <code>false</code> .
\$Site.IsPasswordExpired	For authenticated users, returns <code>true</code> if the currently logged-in user's password is expired. For non-authenticated users, returns <code>false</code> .
\$Site.AdminEmailAddress	Returns an empty string. This merge field is deprecated.
\$Site.Prefix	Returns the URL path prefix of the current site. For example, if your site URL is <code>MyDomainName.my.salesforce-sites.com/partners</code> , <code>/partners</code> is the path prefix. Returns <code>null</code> if the prefix isn't defined. If the current request isn't a site request, then this field returns an empty string.
\$Site.Template	Returns the template name associated with the current site; returns the default template if no template has been designated.
\$Site.ErrorMessage	Returns an error message for the current page if it's a designated error page for the site and an error exists; otherwise, returns an empty string.
\$Site.ErrorDescription	Returns the error description for the current page if it's a designated error page for the site and an error exists; otherwise, returns an empty string.
\$Site.AnalyticsTrackingCode	The tracking code associated with your site. Services such as Google Analytics can use this code to track page request data for your site.
\$Site.BaseCustomUrl	Returns a base URL for the current site that doesn't use a subdomain. The returned URL uses the same protocol (HTTP or HTTPS) as the current request if at least one non-force.com custom URL that supports HTTPS exists on the site. The returned value never ends with a <code>/</code> character. If all the custom URLs in this site end in <code>salesforce-sites.com</code> or if this site has no custom URLs, \$Site.BaseCustomUrl returns an empty string. If the current request isn't a site request, then this method returns an empty string. This field replaces CustomWebAddress and includes the custom URL's path prefix.
\$Site.BaseInsecureUrl	This merge field is deprecated. Returns a base URL for the

Merge Field	Description
	current site that uses HTTP instead of HTTPS. The current request's domain is used. The returned value includes the path prefix and never ends with a / character. If the current request isn't a site request, then this method returns an empty string
\$Site.BaseRequestUrl	Returns the base URL of the current site for the requested URL. The referring page's URL doesn't influence \$Site.BaseRequestUrl. The returned URL uses the same protocol (HTTP or HTTPS) as the current request. The returned value includes the path prefix and never ends with a / character. If the current request isn't a site request, then this method returns an empty string.
\$Site.BaseSecureUrl	Returns a base URL for the current site that uses HTTPS instead of HTTP. The current request's domain is preferred if it supports HTTPS. Domains that aren't force.com subdomains are preferred over force.com subdomains. A force.com subdomain, if associated with the site, is used if no other HTTPS domains exist in the current site. If there are no HTTPS custom URLs in the site, then this method returns an empty string. The returned value includes the path prefix and never ends with a / character. If the current request isn't a site request, then this method returns an empty string.
\$SiteBaseUrl	Returns the base URL of the current site that references and links use. This field can return the referring page's URL instead of the current request's URL. This field's value includes the path prefix and never ends with a / character. If the current request isn't a site request, then this field returns an empty string. This field replaces \$Site.CurrentSiteUrl.
\$Site.MasterLabel	Returns the value of the Master Label field for the current site. If the current request isn't a site request, then this field returns an empty string.
\$Site.SiteId	Returns the ID of the current site. If the current request isn't a site request, then this field returns an empty string.
\$Site.SiteType	Returns the API value of the Site Type field for the current site. If the current request isn't a site request, then this field returns an empty string.
\$Site.SiteTypeLabel	Returns the value of the Site Type field's label for the current site. If the current request isn't a site request, then this field

Merge Field	Description
	returns an empty string.

 **Note** To use these expressions, the Salesforce Sites feature must be enabled for your organization. You must also use them within the context of your public site; otherwise, an empty string is returned for all expressions except {!\$Site.Template}, which returns the default template for the site.

Salesforce Sites Considerations

Before you create and use Salesforce Sites, review considerations around App Exchange packaging, access and authentication, and potential URL conflicts between sites.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

Packaging

Sites aren't packageable. However, you can package sample code, Visualforce pages, Apex classes, or components for a site using a managed package.

To install unmanaged packages that contain Visualforce pages or Apex classes that refer to a site, Salesforce Sites must be enabled.

Access and Authentication

You can grant Read and Create permissions on all standard objects except products, price books, and ideas. Also, you can grant Read, Create, Edit, and Delete permissions on all custom objects. For additional access, you must authenticate site visitors as portal users.

Custom authentication isn't supported. Only these methods are supported for Salesforce Site authentication.

- Customer Portals—[enable public login and registration](#)
- Partner portals—create partner users

 **Tip** You can also enable single sign-on for portals, as well as Sites.

URL Conflicts Between Sites

If you host multiple sites on the same Salesforce-managed domain, be sure to review your site URLs for

conflicts, as it's possible to configure the same URL for pages on two different sites.

Let's say you use `https://SitesSubdomain.force.com` for Site A's homepage, and `https://SitesSubdomain.force.com/products` for Site B's homepage. If you create a page on Site A that uses the subpath `/products`, that page and Site B's homepage both use the URL `https://SitesSubdomain.force.com/products`.

In this scenario, a site visitor can only access the Site A page through a navigation menu on Site A. If a site visitor navigates to `https://SitesSubdomain.force.com/products` any other way, they're directed to Site B's home page.

See Also

[Salesforce Sites](#)

Salesforce Sites Usage and Billing

Salesforce Sites usage is governed by monthly and daily allocations. Understanding these allocations is important to the success of your sites.

REQUIRED EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, and Unlimited** Editions

Salesforce provides tools to help you [reduce bandwidth consumption](#) and monitor site usage.

- [Usage and Billing Terminology](#)
- [Sites Allocation by Edition](#)
- [Bandwidth and Service Request Time Limit Enforcement](#)
- [Billing and Monthly Page Views Enforcement](#)
- [What Counts as a Page View?](#)
- [Monitoring Usage](#)

Usage and Billing Terminology

This section defines the terminology used for Salesforce Sites usage and billing.

- “Page Views” are calculated as the total number of pages served from the site’s origin server.
- “Bandwidth” is calculated as the number of megabytes served and received from both the site’s origin server and the cache server.
- “Service request time” is calculated as the total server time in minutes required to generate pages for the site.
- “Rolling 24-hour period” refers to the 24 hours immediately preceding the current time.
- “Origin server” refers to the web server that hosts your site.
- “Cache server” refers to the CDN server that serves your cached site pages.

- “Current period” refers to the current calendar month for which you are entitled a certain number of page views for your organization.

Sites Allocation by Edition

The following table lists allocations for each edition.

Edition	Maximum Number of Sites	Bandwidth Allocation (per rolling 24-hour period per site)	Service Request Time (per rolling 24-hour period per site)	Maximum Page Views
Developer Edition	1	500 MB	10 minutes	N/A
Enterprise Edition	25	1 GB for sandbox 40 GB for production	30 minutes for sandbox 60 hours for production	500,000
Unlimited Edition	25	1 GB for sandbox 40 GB for production	30 minutes for sandbox 60 hours for production	1,000,000
Performance Edition				

Make sure to consider the [available caching options](#) to keep you within your allocation. Use the Site Usage analytics tools to monitor your Salesforce Sites.

Bandwidth and Service Request Time Limit Enforcement

Bandwidth and Service Request Time limits are tracked and enforced over a 24-hour period. Sites that exceed provisioned limits within the 24-hour period remain available if the host instance has resources to serve the site. However, even if a site is available once limits are exceeded, there's no guarantee in service level.

Billing and Monthly Page Views Enforcement

This section describes how Salesforce enforces limits on monthly page views for Salesforce Sites.

- Billing is based on the number of monthly page views purchased for your organization. This page view limit is cumulative for all Salesforce Sites in your organization.
- If your organization exceeds 110% of its page view limit for four consecutive calendar months, your Salesforce Sites can be disabled until the next calendar month begins or you purchase more page views. Before disabling Salesforce Sites for this reason, Salesforce sends an email notification to the site and billing administrators, and the related account executive.
- If, in a given calendar month, your organization reaches 300% of its page view limit, your Salesforce

Sites can be disabled until the next calendar month begins or you purchase more page views. Before disabling Salesforce Sites for this reason, Salesforce sends an email notification to the site and billing administrators, and the related account executive.

What Counts as a Page View?

This section describes how page views are counted for Salesforce Sites.

A page view is a request from a non-authenticated site user to load a page associated with one of the sites within your Salesforce Sites domain or custom domain. Requests from authenticated portal users are not counted as page views.

These requests count as page views.

Requests for...	Example URL
Your Salesforce Sites domain	<code>https://MyDomainName.my.salesforce-sites.com</code>
Your custom web address	<code>https://mycompany.com</code>
Any page associated with your site	<code>https://MyDomainName.my.salesforce-sites.com/mypage</code>
Authorization Required error page	<code>https://MyDomainName.my.salesforce-sites.com/Unauthorized</code>
AJAX requests such as:	n/a
<ul style="list-style-type: none"> • JavaScript remoting (for example, Apex <code>RemoteAction</code>) • Lightning • Visualforce <code><apex:actionFunction></code> • Visualforce <code><apex:actionPoller></code> 	

Requests that don't count as page views.

Requests for...	Example URL
Salesforce images	<code>https://MyDomainName.my.salesforce-sites.com/img/force_logo_w09.gif</code>
Your static resources	<code>https://MyDomainName.my.salesforce-sites.com/resource/1233771498000/background</code>

Requests for...	Example URL
Robots.txt	<code>https://MyDomainName.my.salesforce-sites.com/robots.txt</code>
Favorite icon	<code>https://MyDomainName.my.salesforce-sites.com/favicon.ico</code>
Attachments and Documents	n/a
Error pages, apart from Authorization Required, such as Limit Exceeded and Maintenance	<code>https://MyDomainName.my.salesforce-sites.com/BandwidthExceeded</code>
Images included with an HTML field	<code>https://MyDomainName.my.salesforce-sites.com/servlet/rtaImage</code>
Custom file field	<code>https://MyDomainName.my.salesforce-sites.com/servlet/fileField</code>

Monitoring Usage

Page views, bandwidth, and time consumption are tracked and made available in your org. You can view this information for a site under **Setup | Build | Develop | Sites**. Select a site, and you see related lists for page views for the current month's billing cycle, and the 24-hour bandwidth and service request time usage history.

Also, you can install the [Salesforce Sites Usage Reporting](#) app from AppExchange to monitor usage. Keep in mind that the information available in the app might not be as current as the information you find directly in your org.

For more information about bandwidth and service request time, see [View 24-Hour Salesforce Sites Usage History](#).

See Also

- [View 24-Hour Salesforce Sites Usage History](#)
- [Track Your Salesforce Sites with Google Analytics](#)

Can I use the same domain name for my Salesforce Sites and my Experience Cloud Sites?

With enhanced domains, your Salesforce org's My Domain name is the subdomain for Salesforce Sites and Experience Cloud sites. If enhanced domains aren't deployed in your org, you can't use the same domain name for Salesforce Sites and Experience Cloud sites.

Here are the URL formats for Experience Cloud sites and Salesforce Sites in orgs with and without enhanced domains.

URL Type	Format	Format Without Enhanced Domains
Experience Cloud sites	<code>MyDomainName.my.site.com</code>	<code>ExperienceCloudSitesSubdomainName.force.com</code>
Salesforce Sites	<code>MyDomainName.my.salesforce-sites.com</code>	<code>SitesSubdomainName.secure.force.com</code> or <code>SitesSubdomainName.force.com</code>

See Also

[Enhanced Domains](#)

Develop and Deploy with DX Inspector

DX Inspector is a management tool that provides quick access to metadata change tracking and visualization. It's available at the top of a page or builder in your Salesforce sandbox or scratch org.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

DX Inspector shows the name of the org (if available) that you're working in. It also provides an entry point to a panel with more metadata tracking capabilities. Before you start using these capabilities, review and accept the opt-in terms.



You can also create and view the status of work items, or switch between them to focus on high-priority work items.

Change Management

The Change Management page offers a unified view of all the modifications across your current org. From this page, you can view, manage, and commit metadata changes to your source control repository. You can also move metadata and data together for direct deployment to a target org.

Metadata (Beta) and Data (Developer Preview) Deployment

Quickly move metadata and configuration data from your current Salesforce org to a target org in a single orchestration. This type of direct deployment is ideal for moving small, specific changes without using a full DevOps pipeline.

Change Management

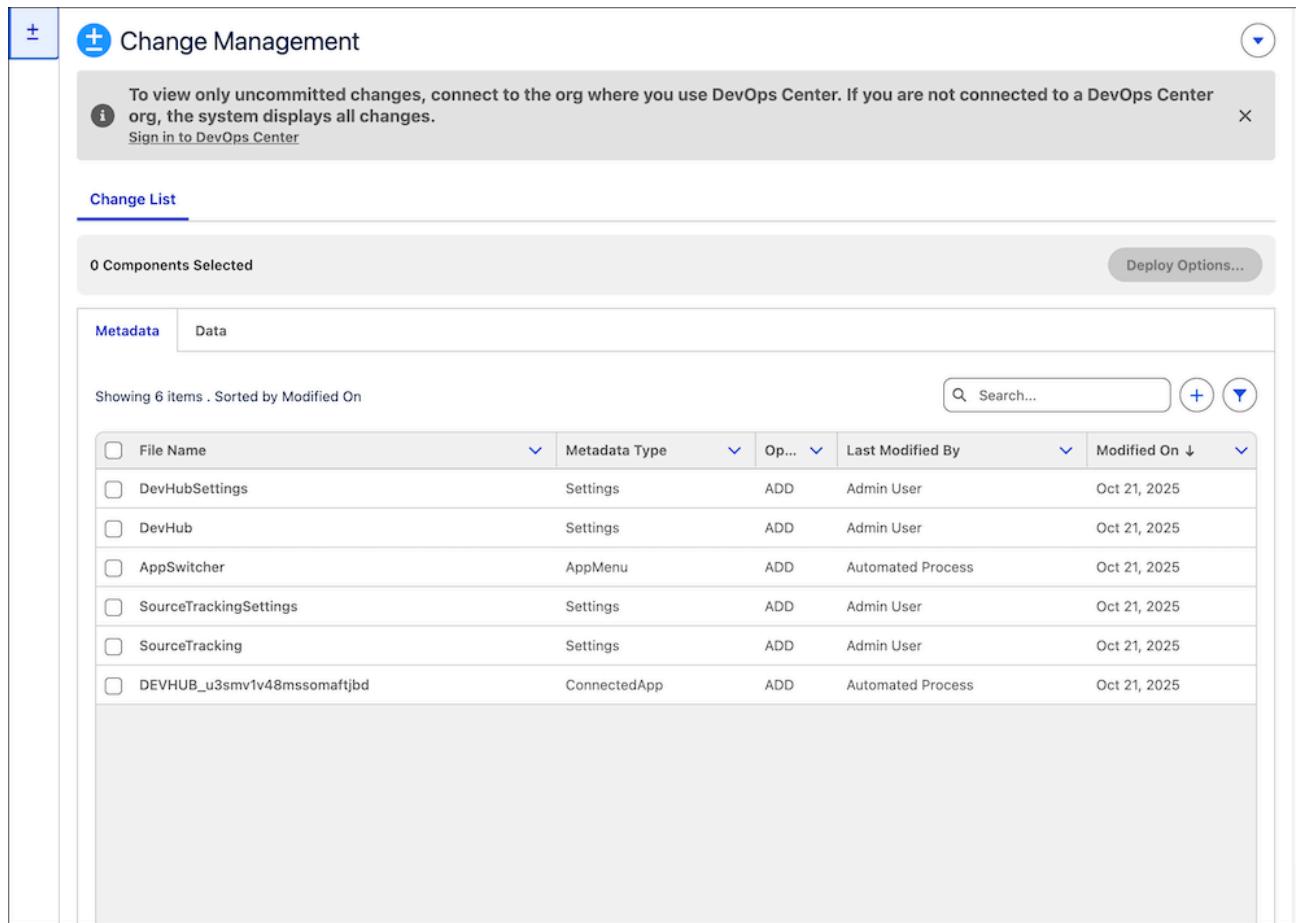
The Change Management page offers a unified view of all the modifications across your current org. From this page, you can view, manage, and commit metadata changes to your source control repository. You can also move metadata and data together for direct deployment to a target org.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Access the Change Management page by clicking the  icon in the DX Inspector panel.



File Name	Metadata Type	Op...	Last Modified By	Modified On
DevHubSettings	Settings	ADD	Admin User	Oct 21, 2025
DevHub	Settings	ADD	Admin User	Oct 21, 2025
AppSwitcher	AppMenu	ADD	Automated Process	Oct 21, 2025
SourceTrackingSettings	Settings	ADD	Admin User	Oct 21, 2025
SourceTracking	Settings	ADD	Admin User	Oct 21, 2025
DEVHUB_u3smv1v48mssomaftjbd	ConnectedApp	ADD	Automated Process	Oct 21, 2025

Source tracking automatically tracks changes to metadata files in sandbox and scratch orgs, which you can see in the Change Management page. If you haven't enabled source tracking, you can manually add metadata to the Change Management page.

-  **Note** The Change List table shows up to 4,000 changes. If source-tracked changes exceed this limit, the table appears empty. To show the changes, restart source tracking, refresh the sandbox, or delete changes from the source tracking table.

To filter and view uncommitted changes, authenticate and connect to DevOps Center. This filtered view shows the changes since your last commit to DevOps Center. You can sort, search, and filter the list by date, org, type of metadata, manual metadata, and uncommitted changes. After you review the changes, commit them directly to the source control repository.

If you're not connected to DevOps Center, you can deploy metadata and data together in a single orchestration to a target org. This type of deployment is suitable for solo Salesforce admins or smaller teams.

Commit Changes to Source Repository

Commit changes to your source repository directly from the Change Management page. After you review the changes, decide which changes to include for the work item and commit them to the DevOps Center project's source control repository. Seamlessly commit changes, and then go to DevOps Center to review, promote, and deploy your work items.

Add Metadata Manually

The Change Management page uses source tracking to automatically track metadata files in your org. However, when you require flexibility to include additional metadata files that source tracking hasn't pulled automatically, you can add metadata manually.

Create a Change Request

After you commit changes to an In Progress work item and are ready to share them with collaborators, start a review from the development environment using DX Inspector. Creating a review moves the work item to In Review status and opens a change request (also called a pull request).

Commit Changes to Source Repository

Commit changes to your source repository directly from the Change Management page. After you review the changes, decide which changes to include for the work item and commit them to the DevOps Center project's source control repository. Seamlessly commit changes, and then go to DevOps Center to review, promote, and deploy your work items.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS NEEDED

To commit changes: Customize Application

To commit changes: DevOps Center User in DevOps Center Hub org^[SEP]

1. Log into your sandbox or scratch org.
2. Click  on DX Inspector to open the DX Inspector panel.
3. From the DX Inspector panel, open the **Change Management** page.
4. Click , and then select **Connect DevOps Center**.

This action authenticates the connection to the DevOps Center GitHub source repository. On successful authentication, all uncommitted changes are listed.

5. (Optional) Add metadata manually. See [Add Metadata Manually](#).
6. Review and select the changes that you want to include for the work item. See [Work Item Management](#).
7. Click **Commit Changes....**
8. Select a DevOps Center project.
Only work items in the project with the statuses In Progress, In Review, and Ready to Promote are displayed. If there are no work items, we recommend [creating a new work item](#) directly in DevOps Center.
9. Select a work item, and then click **Next**.
10. Add a commit comment that clearly describes the changes.
This helps in identifying your commit, especially when the work item has multiple commits.
11. Review your changes, and then click **Commit**.

After a successful commit, go to DevOps Center to review, promote, and deploy your work items.

See Also

[Salesforce Help: Manage and Release Changes Easily and Collaboratively with DevOps Center](#)
[Add Metadata Manually](#)

Add Metadata Manually

The Change Management page uses source tracking to automatically track metadata files in your org. However, when you require flexibility to include additional metadata files that source tracking hasn't pulled automatically, you can add metadata manually.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, and Unlimited Editions**

USER PERMISSIONS NEEDED

To add metadata manually:	Customize Application
---------------------------	-----------------------

Consider adding metadata manually for these cases.

- Source tracking doesn't support the metadata file, so it isn't listed on the Change Management page. See [Metadata Coverage](#) for the list of metadata types supported by source tracking.
- Your development environment is shared, and another teammate has already committed a file that you also want to include in your work item commit. If the metadata hasn't changed in your environment since your teammate's commit, it doesn't automatically appear in your changes list.
- You want to commit files that were changed before source tracking was enabled in your environment.

-  **Note** You can't manually add metadata that's already on the Change Management page, and the original operation type (CHANGE or ADD) remains unchanged.

1. From the DX Inspector panel, open the **Change Management** page.
2. Click + .
3. Select a metadata type from the dropdown.
Search by file name or last modified by to narrow your search results.
4. Select the metadata files that you want to add.
5. Click **Add**.

The file appears on the Change Management page with the operation type MANUAL, ready for you to include in your next commit.

See Also

[Salesforce Help: Manage and Release Changes Easily and Collaboratively with DevOps Center](#)

Create a Change Request

After you commit changes to an In Progress work item and are ready to share them with collaborators, start a review from the development environment using DX Inspector. Creating a review moves the work item to In Review status and opens a change request (also called a pull request).

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, and Unlimited Editions**

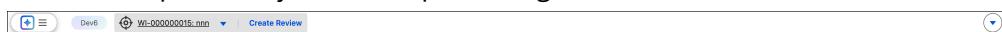
USER PERMISSIONS NEEDED

To create a change request:	Customize Application
To create a change request:	DevOps Center User in DevOps Center Hub org [SEP]

A change request is a proposal to merge your changes into the pipeline. When you initiate a review, your team can discuss and approve the changes in the source control repository.

-  **Note** Commit the changes in your development environment before the Create Review button becomes available.

1. From DX Inspector in your development org, click **Create Review**.



DX Inspector opens a change request in your source control repository and updates the work item status to In Review.

2. Click **Review Created** to view the request in your repository.

You are redirected to the source control repository where team members can review, comment on,

and approve the changes.

See Also

[Salesforce Help: Manage and Release Changes Easily and Collaboratively with DevOps Center](#)

Metadata (Beta) and Data (Developer Preview) Deployment

Quickly move metadata and configuration data from your current Salesforce org to a target org in a single orchestration. This type of direct deployment is ideal for moving small, specific changes without using a full DevOps pipeline.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

-  **Note** Metadata deployment feature is a pilot or beta service that is subject to the Beta Services Terms at [Agreements - Salesforce.com](#) or a written Unified Pilot Agreement if executed by Customer, and applicable terms in the [Product Terms Directory](#). Use of this pilot or beta service is at the Customer's sole discretion.
-  **Note** Data deployment feature is available as a developer preview. Feature isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools.

This feature deploys metadata first and then the related data. It retrieves selected records from the source org and migrates them to the target org. To maintain data integrity, it migrates records based on their relationships. For example, the feature migrates parent records first to capture their new record IDs, and then it migrates and maps child records to the correct parent records by using these IDs. If external IDs are present, they're used to upsert records, preventing duplicates and ensuring existing records in the target org are updated with the latest changes.

Use Cases

- Move configuration data, such as objects containing product catalogs and business rules, between environments.
- Deploy a new feature from a development sandbox to your production org after development and testing are complete.
- Develop and test changes in a scratch org, and then move them to a sandbox for further integration. Seed the sandbox with specific data for validation.
- Move metadata and data between developer sandboxes to integrate changes from different team members. This action is useful for moving a combined feature set to a shared sandbox for User

- Acceptance Testing (UAT).
- Deploy the final set of customizations to a staging or preproduction environment and populate it with realistic data that mirrors production for final testing.
 - Set up new environments, refresh sandboxes with production data, or move configuration records.

Plan and Prepare for Deployment

A successful deployment starts with a solid plan. To avoid conflicts or missing dependent metadata and configuration data during deployments, review these prerequisites, best practices, and limitations.

Enable Metadata (Beta) and Data (Developer Preview) Deployment

Turn on the metadata and data deployment feature so that you can start deploying metadata and migrating configuration data to your target org.

Deploy Metadata (Beta) and Data (Developer Preview) to a Target Org

Move your metadata changes and configuration data from your current org to a target org.

Monitor and Verify Deployments

After the deployment process completes, monitor its status and verify the results.

Troubleshoot Deployment Failures

Review these common issues and learn how to fix them. If you need help, contact your Salesforce admin.

See Also

[Salesforce Help: Deploy Your Changes](#)

Plan and Prepare for Deployment

A successful deployment starts with a solid plan. To avoid conflicts or missing dependent metadata and configuration data during deployments, review these prerequisites, best practices, and limitations.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

Prerequisites

- Authorize the source and target orgs to send and receive changes.
- Include all required metadata components and their dependencies to prevent deployment failure.
- Review your data structure, including all objects, fields, records, and relationships, to identify which configuration data to migrate.
- Map the relationships between your selected objects. For example, Accounts are parents to Contacts. To maintain data integrity, plan the migration order based on these dependencies.
- Understand how the deployment affects different metadata types and data objects. Some metadata components can't be overwritten. See [Special Behavior During Deployments](#).
- Verify that you have the object and field-level permissions in the target org to create and edit records

for the specific objects intended for configuration data migration.

Best Practices

- Set [external IDs](#) before you start. While optional, we highly recommend using external IDs for configuration data migration. The deployment process uses the External ID field to match records between the source and target orgs. If an external ID is present, the process upserts records, which prevents duplicates in your target org.
- Back up your target org's metadata and data before starting the deployment so that you can restore the previous state, if needed.
- Deploy in small batches to minimize risks and identify issues more quickly.
- Monitor your org limits to better coordinate deployments.

Limitation

Each deployment operation can migrate up to 2,000 records per object, with a total limit of 10,000 records.

Enable Metadata (Beta) and Data (Developer Preview) Deployment

Turn on the metadata and data deployment feature so that you can start deploying metadata and migrating configuration data to your target org.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To turn on metadata deployment: Customize Application

To turn on configuration data migration: Customize Application

To deploy metadata, turn on metadata deployment. However, for configuration data migration, first turn on metadata deployment, and then turn on data deployment. Data migration fails if the required metadata isn't present in the target org.

1. From Setup, in the Quick Find box, enter *Change Management*, and then select **Change Management**.
2. To deploy metadata, turn on **Org-to-org metadata deployment (Beta)**.
3. To migrate configuration data, turn on **Data deployment (Developer Preview)**.

Deploy Metadata (Beta) and Data (Developer Preview) to a Target Org

Move your metadata changes and configuration data from your current org to a target org.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS NEEDED

To turn on metadata deployment: Customize Application

To turn on configuration data migration: Customize Application

Before you start, plan your deployment and then, turn on the feature.

From DX Inspector, go to the Change Management page and open the Change List tab.

Select Metadata to Deploy (Beta)

On the Metadata tab, select the metadata files to deploy.

Select Data Records to Migrate (Developer Preview)

1. On the Data tab, find the data object that you want to migrate.
2. In the data object row, click the down arrow (▼) icon, and then select **Select Associated Records....**
3. Select the related objects to include in the deployment, and then click **Next**.

Configure Deployment Data (Developer Preview)

Select related objects

Object: Account Brand (Standard Object)

Related Objects
Include related objects to keep record relationships intact in the target org. You can configure fields, filters, and records in the next step.
No objects selected • Maximum 10 objects allowed
Showing 4 objects

Object	Type	Number of Records
Account Brand Share	Standard	7
Content Version	Standard	1
Content Document Link	Standard	1
Flow Record Relation	Standard	0

Cancel **Next**

4. For each selected object, complete these steps:
 - a. Select a unique External ID to map records and prevent duplicates.
 - b. (Optional) To include only specific fields, click **Edit Fields....** By default, the required fields are selected.
 - c. (Optional) To migrate a subset of records, click **Add Filters** and define your filter criteria.

Configure Deployment Data (Developer Preview)

Account
5 Records

User
7 Records

Define deployment details
Use fields, filters, and record selection to define the data you deploy. Set an external ID to maintain relationships and prevent duplicates.

External ID Configured for Insert-Only **Select...**

Fields Required Fields Preselected **Select Fields...**

Filters No Filters Applied **Add Filters**

Showing 5 of 5 records

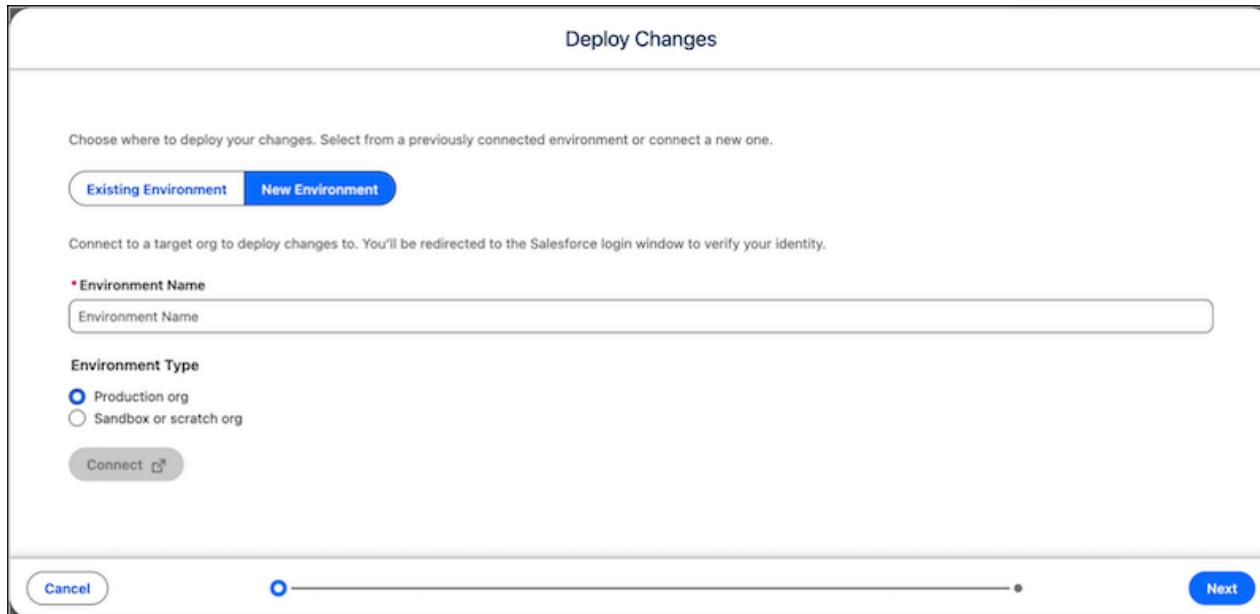
Record	Last Change	Status	Owner
ACC_OCT13_NM	10/13/2025, 09:57:58 AM	ACTIVE	System
ACC_OCT13_B	10/13/2025, 10:45:26 AM	ACTIVE	System
ACC_OCT13	10/13/2025, 10:46:05 AM	ACTIVE	System
ACC_OCT13_LA	10/13/2025, 10:46:33 AM	ACTIVE	System
ACME	10/13/2025, 12:03:47 PM	ACTIVE	System

Back **Next**

5. Click **Next**.
6. Review the list of filtered records and click **Add**.
7. (Optional) In the data object row, click the down arrow (▼) icon to download or remove the records.

Deploy Metadata and Data

1. On the Change Management page, click **Deploy Options....**
2. In the Deploy Changes window, select the target org.



3. (Optional) If your target org isn't listed, connect to a new org and authenticate.
4. Click **Next**.
5. Review the deployment summary and click **Deploy**.

Monitor and Verify Deployments

After the deployment process completes, monitor its status and verify the results.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

The deployment process runs validation checks before moving your changes. After the deployment completes, review the status and the reason for any failures.

- To monitor metadata deployment, see [Monitor Deployments](#).
- To monitor configuration data migration, see [View Bulk Data Load Job Details](#).

After a successful deployment, thoroughly test your changes in the target org to confirm that everything works as expected.

Troubleshoot Deployment Failures

Review these common issues and learn how to fix them. If you need help, contact your Salesforce admin.

REQUIRED EDITIONS

Available in: Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Issue: A metadata component, such as a custom field, exists in the source org but is missing in the target org.

Action: Identify the missing metadata component and create it in the target org before redeploying.

Issue: The target org is missing related records.

Action: Migrate the missing parent or related records first. Use external IDs to make sure that relationships are maintained correctly.

Issue: You don't have the necessary permissions in the target org.

Action: Verify that you have the required object and field-level permissions. If not, reauthenticate with the correct credentials or ask your Salesforce admin for the required access.

Issue: The connection to the target org was interrupted during the deployment.

Action: Check your network connection and firewall settings. Confirm that your credentials for the target org are still valid.

Organization Sync

This feature is now retired. In previous releases, Organization Sync let you set up a secondary, synced Salesforce org. Users could access a subset of Salesforce data in that org when the primary org was unavailable.

Managing Organization Sync

This feature is now retired. In previous releases, Organization Sync let you set up a secondary, synced Salesforce org. Users could access a subset of Salesforce data in that org when the primary org was unavailable.

Actions in the Organization Sync Record Queue

This feature is now retired. In previous releases, Organization Sync let you set up a secondary, synced Salesforce org. Users could access a subset of Salesforce data in that org when the primary org was unavailable.

Managing Organization Sync

This feature is now retired. In previous releases, Organization Sync let you set up a secondary, synced Salesforce org. Users could access a subset of Salesforce data in that org when the primary org was

unavailable.

Actions in the Organization Sync Record Queue

This feature is now retired. In previous releases, Organization Sync let you set up a secondary, synced Salesforce org. Users could access a subset of Salesforce data in that org when the primary org was unavailable.

Resources for the Point & Click Administrator

In addition to online help, Salesforce creates guides and tip sheets to help you learn about our features and successfully administer Salesforce.

Platform and Apps

Guides and Tip Sheets	For End Users	For Admins
Implementing State and Country/Territory Picklists		✓
Useful Workflow Rules		✓
Useful Approval Processes		✓

Formulas

Guides and Tip Sheets	For End Users	For Admins
Useful Formula Fields		✓
Tips for Reducing Formula Size		✓
Using Date and Date/Time in Formulas		✓
Useful Validation Rules		✓