



ORACLE
NETSUITE

SuiteBundler Guide

2022.2

March 15, 2023



Copyright © 2005, 2022, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

If this document is in public or private pre-General Availability status:

This documentation is in pre-General Availability status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

If this document is in private pre-General Availability status:

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your pre-General Availability trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described in this document may change and remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Master Agreement, Oracle License and Services Agreement, Oracle PartnerNetwork Agreement, Oracle distribution agreement, or other license agreement which has been executed by you and Oracle and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <https://www.oracle.com/corporate/accessibility>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Sample Code

Oracle may provide sample code in SuiteAnswers, the Help Center, User Guides, or elsewhere through help links. All such sample code is provided "as is" and "as available", for use only with an authorized NetSuite Service account, and is made available as a SuiteCloud Technology subject to the SuiteCloud Terms of Service at www.netsuite.com/tos.

Oracle may modify or remove sample code at any time without notice.

No Excessive Use of the Service

As the Service is a multi-tenant service offering on shared databases, Customer may not use the Service in excess of limits or thresholds that Oracle considers commercially reasonable for the Service. If Oracle reasonably concludes that a Customer's use is excessive and/or will cause immediate or ongoing performance issues for one or more of Oracle's other customers, Oracle may slow down or throttle Customer's excess use until such time that Customer's use stays within reasonable limits. If Customer's particular usage pattern requires a higher limit or threshold, then the Customer should procure a subscription to the Service that accommodates a higher limit and/or threshold that more effectively aligns with the Customer's actual usage pattern.

Beta Features

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

This documentation is in pre-General Availability status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your pre-General Availability trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Master Agreement, Oracle License and Services Agreement, Oracle PartnerNetwork Agreement, Oracle distribution agreement, or other license agreement which has been executed by you and Oracle and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

Send Us Your Feedback

We'd like to hear your feedback on this document.

Answering the following questions will help us improve our help content:

- Did you find the information you needed? If not, what was missing?
- Did you find any errors?
- Is the information clear?
- Are the examples correct?
- Do you need more examples?
- What did you like most about this document?

Click [here](#) to send us your comments. If possible, please provide a page number or section title to identify the content you're describing.

To report software issues, contact NetSuite Customer Support.

Table of Contents

SuiteBundler	1
SuiteBundler Overview	2
Moving from SuiteBundler to SuiteCloud Development Framework	3
Bundle Support Across Account Types	4
SuiteApp Creation and Distribution	8
Types of Bundles	10
Customization Bundles	11
Configuration Bundles	21
Creating a Bundle with the Bundle Builder	24
Step 1 Bundle Basics	25
Step 2 Bundle Properties	27
Step 3 Select Objects	28
Step 4 Set Preferences	29
Finish the Bundle Builder	30
Bundle Builder Reference	31
Hiding Bundle Components in Target Accounts	32
Using Bundle Installation Scripts	33
Bundle Support during NetSuite Release Phasing	34
Documenting a Bundle	34
Defining Bundle Terms of Service	36
Protecting Your Bundled Server SuiteScripts	39
File and Folder Management in Bundles	40
Bundling NetSuite Financial Statements	44
Bundle Object Preferences	45
Locking Objects in Customization Bundles	48
Managed Bundles Overview	50
Managed Bundles - Best Practices	52
Upgrading Your Managed Bundle Install Base	53
Saved Bundles	57
Reviewing the Saved Bundles List	58
Bundle Availability	59
Publishing SuiteApp Help Content to the NetSuite Help Center	66
Copying a Bundle to Other Accounts	69
Pushing a Bundle to Other Accounts	71
Viewing Your Bundle Install Base	72
Editing a Bundle	73
Deleting a Bundle	74
Bundle Details	75
Sending Bundle Messages	77
SuiteApp Installation and Update	79
Resolving Conflicting Objects	79
Bundle Searches Overview	81
Installing a Bundle	84
Choosing a Bundle to Install	84
Reviewing the Preview Bundle Install Page	85
Starting the Bundle Installation Process	87
Bundle Installation Notes	88
Reviewing the Installed Bundles List	90
Choosing to Hide or Show Bundle Components	91
Subscribing to Bundle Email Messages	92
Installed Bundle Updates	93
Choosing a Bundle to Update	94
Reviewing the Preview Bundle Update Page	95

Bundle Update Reference	99
Using Managed Bundles	110
Uninstalling a Bundle	112
Viewing the Bundle Audit Trail	112
Identifying Bundle Objects in Target Accounts	114
Filtering Bundle Objects on List Pages	115
SuiteApp Development Process with SuiteBundler	117
Single Development Account Method	121
Multiple Development Account Methods	122
Single Development Account Environment Method	122
Multiple Development Account Environments Method	124
Bug Fixing Methodology	126
SuiteApps and Sandbox Accounts	128
Single Sandbox Bundle Deployment Model	129
Two Sandbox Bundle Deployment Model	130
Sandbox Refresh Impact on Bundles	131
Selective Update of Sandbox Bundle Objects	134
Dissolving Bundles Created in Sandbox	136
Avoiding Bundle Installation Errors due to Duplicate External IDs	137
SuiteBundler Pages	140
Bundle Availability, Install Bundle Pages	140
Bundle List Pages	140
Bundle Action Pages	140
Bundle Preview Pages	140
Custom Transaction Types in Bundles	142
Pre-Bundling Checklist	142
Including a Custom Transaction Type in a Bundle	143
Other Recommended Objects	144
Post-Installation Steps	144
Removing Custom Transaction Types	145
Accounting Preferences for Custom Transactions Applied to a Bundle	145
Adding a Custom Segment to a Bundle	147
Creating and Installing Bundles with Segments	147
Updating Bundles with Segments	148
Uninstalling Bundles with Segments	149
Correcting a Damaged Custom Segment	149

SuiteBundler

- SuiteBundler Overview
- SuiteApp Creation and Distribution
- SuiteApp Installation and Update
- SuiteApp Development Process with SuiteBundler
- SuiteApps and Sandbox Accounts
- NetSuite SuiteApps

SuiteBundler Overview



Note: SuiteBundler is still supported, but it will not be updated with any new features.

To take advantage of new features for packaging and distributing customizations, you can use the Copy to Account and SuiteCloud Development (SDF) features instead of SuiteBundler.

Copy to Account is an administrator tool that you can use to copy custom objects between your accounts. The tool can copy one custom object at a time, including dependencies and data. For more information, see the help topic [Copy to Account Overview](#).

SuiteCloud Development Framework is a development framework that you can use to create SuiteApps from an integrated development environment (IDE) on your local computer. For more information, see the help topic [SuiteCloud Development Framework Overview](#).

SuiteBundler allows NetSuite users to package together groups of objects for distribution to other accounts. These packages are called bundles, or SuiteApps. Bundle authors may be internal developers creating customizations for their companies, independent software vendors (ISVs) distributing solutions to their customers, or administrators making their bundles publicly available through a deployment account.

Two types of bundles can be created and made available for installation:

- A **customization bundle** is a group of custom objects that implement customized behavior in the NetSuite application.
- A **configuration bundle** is a group of NetSuite setup entries and preference settings.

NetSuite provides an assistant called the Bundle Builder that steps you through the process of creating bundles. NetSuite account administrators and users with the SuiteApp Marketplace permission can use the Bundle Builder when the SuiteBundler feature is enabled. In the Bundle Builder, users define bundle properties and select objects to be included.

After a bundle has been created, the bundle author sets its availability, indicating whether it should be private, shared among a specified set of target NetSuite accounts, or publicly available to all NetSuite accounts.

NetSuite account administrators and other users with the SuiteApp Marketplace permission can install bundles that have been shared with their accounts or made public. Users can install a bundle directly from another NetSuite account if they have the account ID.

Users can search for a bundle across all locations, including production accounts and sandbox accounts. They also have options to narrow a bundle search to a specific location or filter it by keywords or by bundle characteristics such as Availability, Product, Vertical, and Language.



Important: Bundles are also known as SuiteApps.

For information about how bundle support varies in different NetSuite account types, including sandbox, Release Preview, and development accounts, see [Bundle Support Across Account Types](#).

- [SuiteApp Creation and Distribution](#)
 - [Types of Bundles](#)

- Creating a Bundle with the Bundle Builder
- Managed Bundles Overview
- Copying a Bundle to Other Accounts
- Deprecating a Bundle
- Pushing a Bundle to Other Accounts
- Viewing Your Bundle Install Base
- Editing a Bundle
- Bundle Details
- SuiteApp Development Process with SuiteBundler
- SuiteApps and Sandbox Accounts
- SuiteApp Installation and Update
 - Resolving Conflicting Objects
 - Bundle Searches Overview
 - Installing a Bundle
 - Reviewing the Installed Bundles List
 - Installed Bundle Updates
 - Using Managed Bundles
 - Uninstalling a Bundle
 - Viewing the Bundle Audit Trail
 - Identifying Bundle Objects in Target Accounts
 - Filtering Bundle Objects on List Pages
- SuiteApps and Sandbox Accounts
- NetSuite SuiteApps

For information about creating bundles that include custom transaction types, see [Custom Transaction Types in Bundles](#).

For information about adding a custom segment to a bundle, see [Adding a Custom Segment to a Bundle](#).

You can also use SuiteScript to get information about installed bundles and SuiteApps. For more information, see the help topic [N/suiteAppInfo Module](#).

Moving from SuiteBundler to SuiteCloud Development Framework

SuiteBundler is no longer being updated. As soon as possible, you should shift away from using SuiteBundler for development and distributions of NetSuite customizations. Instead, you can take advantage of the enhanced capabilities of SuiteCloud Development Framework (SDF), SuiteApp Control Center, and Copy to Account. To help you decide which of these features is appropriate for packaging and distributing your customizations, see the following:

- Develop New SuiteApps
- Copy a Custom Object to Another NetSuite Account
- Convert Your Existing Installed Bundles to SDF-Built SuiteApps

Develop New SuiteApps

If you are a SuiteCloud Developer Network (SDN) member, you can use SDF to develop new SuiteApps in a SuiteApp project within an IDE. You can then use SuiteApp Control Center in NetSuite to publish your SDF SuiteApp to the SuiteApp Marketplace and to distribute updates to your customers' NetSuite accounts. Customers can install or update SDF SuiteApps from the SuiteApp Marketplace.

For information about using SDF to develop custom objects, see the help topic [SuiteCloud Development Framework Overview](#).

For information about using SuiteApp Control Center to distribute SuiteApps, see the help topic [SuiteApp Control Center Overview](#).

Copy a Custom Object to Another NetSuite Account

For cases where you need to copy an individual custom object to another NetSuite account you administer, you can use Copy to Account to distribute the custom object and its dependent objects to that account.

For information about copying a custom object from one account to another account you administer, see the help topic [Copy to Account Overview](#).

Convert Your Existing Installed Bundles to SDF-Built SuiteApps

You can convert bundles that you have built with SuiteBundler and installed in accounts that you administer into SDF account customization projects (ACPs). After these customizations are converted, they are available for further development in SDF from within an IDE or CLI.

ACPs are intended for customizations in accounts that you own, including production, development, and sandbox accounts. When you convert a bundle, converted files are saved in a zip file in the File Cabinet. You can download the contents of this zip file modified as needed. This conversion allows you to use SDF for continued development of custom objects from existing SuiteApps for your own company's use.



Note: You cannot convert a bundle directly to a SuiteApp project. However, you can convert a bundle to an ACP, and you can copy custom objects from the ACP project to a SuiteApp project within an IDE.

When you are ready to deploy the customizations to a target account, you can use one of the following tools: SuiteCloud IDE for WebStorm, SuiteCloud Extension for Visual Studio, SuiteCloud CLI for Node.js, SuiteCloud CLI for Java.

To convert a bundle into an SDF account customization project, see the help topic [Conversion of a Bundle to an Account Customization Project for SDF](#).

For more information about developing custom objects in ACPs, see the help topic [Account Customization Projects](#).

Bundle Support Across Account Types

The NetSuite application is accessible in different types of accounts. These different account types serve different purposes. Support varies for bundle operations across the different account types. See the following for details:

- [Account Types that Support Bundles](#)

- Support for Bundle Install and Update Across Account Types
- Support for Bundle Copy and Push Across Account Types
- Support for Managed Bundle Upgrade Across Account Types
- Support for Bundle Search Across Account Types

Account Types that Support Bundles

- Production

The production (or live) account is where users do the daily work necessary to run their business. See the help topic [The Production Account](#).

- Sandbox Accounts

A NetSuite sandbox is an isolated account where you can develop new processes and applications. The data in the sandbox is a copy of your production data. Sandbox account data can be refreshed periodically to include updated production data. See the help topic [NetSuite Sandbox](#).

The URLs that customers use to access this type of sandbox account are the same as the URLs they use to access their production accounts, for example, <https://system.netsuite.com>. Users access sandbox accounts from the Change Roles list in the production account.

- Development Accounts

NetSuite development accounts provide an area, isolated from production, in which you can develop and test new applications and customizations without affecting your production account. Development accounts have the same features and modules as your production account, but they do not contain production data. Development accounts are never refreshed, unlike sandbox accounts. See the help topic [NetSuite Development Accounts](#) or [SuiteApp Development Process with SuiteBundler](#).

- Release Preview

Release Preview is available twice a year, immediately before each NetSuite release. In Release Preview, you can test your daily business processes with the new features available in the upcoming release. Release Preview data is a copy of your production data as of a specific point in time. See the help topics [The Release Preview Account](#) and .

The URL that customers use to access Release Preview accounts is the same URL they use to access their production accounts, for example, <https://system.netsuite.com>. Users access the Release Preview account from the Change Roles list in the production account.

Support for Bundle Install and Update Across Account Types

The following table illustrates support for bundle installation and update across account types.

		BUNDLE TARGET			
		Production	Development Accounts	Sandbox Accounts	Release Preview
BUNDLE SOURCE	Production	Yes	Yes	Yes	Yes
	Development Accounts	Yes	Yes	Yes	Yes
	Sandbox Accounts	Yes	Yes	Yes	Yes
	Release Preview	No	No	No	Yes

For information about installing a bundle, see [Installing a Bundle](#).

For information about updating a bundle, see [Installed Bundle Updates](#).

For information about sandbox refresh impact on bundles, see [Sandbox Refresh Impact on Bundles](#).

Support for Bundle Copy and Push Across Account Types

You can copy a customization bundle from one account to another. The ability to copy bundles is linked to the ability to deprecate bundles, meaning replacing a bundle with an updated version.

- Bundle copy is only supported for production accounts and development accounts. You cannot copy a bundle from or to any type of account other than production or development.
- For information about copying a bundle, see [Copying a Bundle to Other Accounts](#).

Bundle authors can push their bundles to a NetSuite account to which they have administrator access. This enables NetSuite resellers and independent software vendors to install bundles without having to log in to each of their customers' accounts.

- Bundle push is only supported for production accounts and development accounts. You cannot push a bundle from or to any type of account other than production or development.
- For information about pushing a bundle, see [Pushing a Bundle to Other Accounts](#).

Support for Managed Bundle Upgrade Across Account Types

When a managed bundle is installed, the installer agrees to subscribe to future updates at the discretion of the bundle's author. As updates are developed, they can be installed by the bundle author into the target accounts without any action required by the target account administrator. See [Managed Bundles Overview](#) and [Upgrading Your Managed Bundle Install Base](#).

The following table illustrates support for managed bundle upgrade across account types. The term "Manual" indicates that an automated upgrade initiated by the bundle author is not supported, but a target account user can initiate a manual upgrade.

		BUNDLE TARGET			
		Production	Development Accounts	Sandbox Accounts	Release Preview
BUNDLE SOURCE	Production	Yes	Yes	Manual	Manual
	Development Accounts	Yes	Yes	Manual	Manual
	Sandbox Accounts	No	No	No	No
	Release Preview	No	No	No	No

Support for Bundle Search Across Account Types

The Search & Install Bundles page, available at Customization > SuiteBundler > Search & Install Bundles, includes the following two search modes. See [Bundle Searches Overview](#).

- Basic mode. Simple bundle searches across all production locations.
- Advanced mode. Bundle searches that support additional search filters.

The following table illustrates support for the two bundle search modes across account types.

		WHERE SEARCH IS PERFORMED			
		Production	Development Accounts	Sandbox Accounts	Release Preview
WHERE BUNDLE WAS CREATED	Production	Basic Advanced	Basic Advanced	Basic Advanced	Basic Advanced
	Development Accounts	Basic Advanced	Basic Advanced	Basic Advanced	Basic Advanced
	Sandbox Accounts	Advanced	Advanced	Advanced	Advanced
	Release Preview	No	No	No	Basic Advanced

SuiteApp Creation and Distribution



Note: SuiteBundler is still supported, but it will not be updated with any new features.

To take advantage of new features for packaging and distributing customizations, you can use the Copy to Account and SuiteCloud Development (SDF) features instead of SuiteBundler.

Copy to Account is an administrator tool that you can use to copy custom objects between your accounts. The tool can copy one custom object at a time, including dependencies and data. For more information, see the help topic [Copy to Account Overview](#).

SuiteCloud Development Framework is a development framework that you can use to create SuiteApps from an integrated development environment (IDE) on your local computer. For more information, see the help topic [SuiteCloud Development Framework Overview](#).

The SuiteBundler feature enables packaging of customization objects or configuration settings into a bundle that can be installed in other NetSuite accounts. To create bundles, this feature must be enabled in your account, and you must have the SuiteApp Marketplace permission. For details about how bundles can be used and the objects they can include, see [Types of Bundles](#). Each bundle has a unique bundle ID that is used to identify it and differentiate it from other bundles.



Important: Bundles are also known as SuiteApps.

The Bundle Builder walks you through the steps of creating bundles.

- For details about this interface and the step-by-step instructions for creating a bundle, see [Creating a Bundle with the Bundle Builder](#).
- If the Managed Bundles feature is enabled in your account, you can control the update process for your bundles by defining them as managed. See [Managed Bundles Overview](#).
- SuiteCloud Development Framework (SDF) includes support for SuiteApp projects, self-contained, standalone projects that enable SuiteCloud Developer Network (SDN) members to develop and deploy SuiteApps to their NetSuite accounts. SuiteBundler is used to bundle and share SDF SuiteApps. You can create a bundle that contains all objects from an SDF SuiteApp project without manually adding all of the objects to the bundle in the Bundle Builder. For information about using SDF, see the help topic [SuiteCloud Development Framework Overview](#).

After you have created a bundle, you can set it up to be released to other accounts. See the following:

- [Bundle Availability](#)
- [Sharing a Bundle](#)

You can publish user help content for the bundle in the NetSuite Help Center. See [Publishing SuiteApp Help Content to the NetSuite Help Center](#).

You can use bundle copy and deprecate actions to implement a bundle versioning strategy. These actions provide increased flexibility and control of the development and release process. See [Copying a Bundle to Other Accounts](#) and [Deprecating a Bundle](#).

Users of other NetSuite accounts can install your bundle, if it has been shared with them or if it is public. You also can install the bundle in other accounts to which you have administrator access. This type of installation is called a push. See [Pushing a Bundle to Other Accounts](#).

NetSuite tracks the details of accounts where each bundle is installed on the Bundle Install Base page. See [Viewing Your Bundle Install Base](#).

 **Note:** Before you begin creating a bundle, you should consider your strategy for deploying it. For explanations of deployment-related terms and preferred deployment methods, see [SuiteApp Development Process with SuiteBundler](#).

Distributing Bundles Through the SuiteApp Marketplace

You can distribute your SuiteApp bundles through the SuiteApp Marketplace. The information for a SuiteApp listing on the SuiteApp Marketplace is synchronized from the Advanced Partner Center (APC) record, and includes graphics, overview, features, and publisher information. The First Released date is also listed. For more information about the SuiteApp Marketplace, see the help topic [SuiteApp Marketplace](#).

SuiteApp bundles are distributed through the SuiteApp Marketplace when they meet the following requirements:

- The Advanced Partner Center (APC) record for the SuiteApp has completed the listing review and been approved.
- The APC record provides the information displayed to your users in the SuiteApp listing. To find out how to complete a listing review, contact SDN. For more information about the Advanced Partner Center role, see the help topic [The Advanced Partner Center](#).
- The primary bundle ID in the APC record refers to an existing bundle.

 **Note:** Secondary and tertiary bundle IDs are not used for the SuiteApp Marketplace listing. If the primary bundle ID is not set in the APC record, the SuiteApp bundle is not listed in the SuiteApp Marketplace, even when the SuiteApp is listed on [SuiteApp.com](#).

- A Built for NetSuite (BFN) review has been completed.

To find out how to complete a BFN review, see [Built for NetSuite Overview](#) and [BFN Verification Process](#).



Important: Although your users can view SuiteApp bundles on the SuiteApp Marketplace, they must install them from the Bundle Details page. When viewing the SuiteApp listing for your SuiteApp bundle, users can click [View SuiteApp](#) to view and install it from the Bundle Details page. For more information, see the help topics [Installing from the SuiteApp Marketplace](#) and [Installing a Bundle](#).

SuiteApp bundles that are distributed through the SuiteApp Marketplace also appear in bundle searches. For information about searching for bundles, see [Bundle Searches Overview](#) and [Choosing a Bundle to Install](#).

Giving Users Dashboard Access to a SuiteApp

You can provide target account users with visibility and access to your SuiteApps by including a Dashboard SuiteApp portlet script in each of your SuiteApp bundles. This type of script provides a dashboard component for your SuiteApp, a portlet that is available to users from the Personalize Dashboard menu when the bundle is installed in an account. This menu includes a SuiteApps option. A user can click this option to get a quick view of the SuiteApps with a dashboard component that are installed in their account. These SuiteApps are represented by portlet icons in the Personalize Dashboard window. The icon for each SuiteApp is displayed with graphics that support its branding. A user can

double-click or drag a SuiteApp portlet's icon to add a portlet to their dashboard that gives direct access to the SuiteApp.

To add this capability to a SuiteApp, you can create a SuiteScript 2.0 portlet script that provides custom content. You can upload a graphics file to be shown as the icon in the Personalize Dashboard window. On the script deployment record for the new portlet script, you should indicate that the script is designed for a dashboard SuiteApp portlet. And you can indicate the graphics file to be shown for the SuiteApp. For more information, see the help topics [SuiteScript 2.x Portlet Script Type](#) and [Guidelines for Creating a Dashboard SuiteApp Icon](#).

After you have created the portlet script and edited its script deployment record, you should include the script in your SuiteApp. On the Select Objects page of the Bundle Builder, you can indicate the script should be included in the bundle. The graphics file selected for the icon is automatically included in the bundle with the script. In accounts where the bundle containing the new script is installed, users can add the SuiteApp portlet to their dashboards. For more information, see the help topic [SuiteApp Portlets](#).



Important: Be aware that changes to an icon associated with a dashboard SuiteApp portlet are always copied during a bundle update. This overwrite of the icon in the target account occurs even if the bundle preference is set to prevent the update of script deployments.

Types of Bundles

Two types of bundles can be made available for installation in NetSuite accounts:

- A **customization bundle** is a group of custom objects that implement customized behavior in the NetSuite application.
 - Customization bundles allow installation of application packages.
 - Customization bundles can include objects such as custom forms, SuiteScripts, and website layouts.
 - Preferences can be set for some types of objects in customization bundles, to ensure desired setup in target accounts.
 - Some types of objects in customization bundles can be locked to prevent changes from being made to them in target accounts after bundles have been installed, to eliminate unintended changes.
 - As part of the bundle installation and update process, conflicts between customization bundle objects and existing target account objects are detected and resolved.
 - SuiteScripts that implement required bundle setup, called bundle installation scripts, can be associated with customization bundles to be run as part of bundle installation, update, or uninstall.
 - For more information, see [Customization Bundles](#).
- A **configuration bundle** is a group of NetSuite setup entries and preference settings.
 - Configuration bundles can simplify and standardize NetSuite account configuration.
 - Configuration bundles can include settings such as auto-generated numbering settings, company preferences, and enabled features.
 - You cannot do the following with a configuration bundle:
 - Include locked objects in the bundle
 - Share the bundle as public
 - Uninstall the bundle from an account
 - Include a bundle installation script to be run as part of bundle installation, update, or uninstall
 - For more information, [Configuration Bundles](#).



Important: You cannot mix object types between customization and configuration bundles. For example, you cannot include custom fields in a configuration bundle. And, you cannot include a company preference in a customization bundle. If your solution requires both customization and configuration objects, then you must create two separate bundles.

Customization Bundles

Customization bundles are used to group custom objects into packages that can be copied to target accounts to implement customized behavior.

Objects Available in Customization Bundles

The following table lists the object types that you can include in a customization bundle.

If an object is lockable, you can lock it in the Bundle Builder, to indicate that the object should be viewable but not editable in target accounts. See [Locking Objects in Customization Bundles](#).

Object Type	Lockable?	Comments
Advanced PDF/HTML Templates	No	<p>Advanced PDF/HTML templates are listed under Custom Forms on the Select Objects page of the Bundle Builder.</p> <p>Advanced templates for use with custom forms, used to format printed and emailed versions of forms transactions.</p> <p>Advanced PDF/HTML templates cannot be hidden, meaning that end users can modify the template.</p> <p>If you associate an advanced PDF/HTML template with a custom form and then include the custom form in the bundle, the Bundle Builder automatically includes the template.</p> <p>Note: If you select an advanced PDF/HTML template as a preferred template for a form, the template is not installed as preferred when you install the bundle.</p> <p>For more information, see the help topic Advanced PDF/HTML Templates.</p>
CSV Import Maps	No	<p>These objects are saved field mappings for record types. All mappings saved in the Import Assistant are available to be included in bundles. The access and audience set up for each mapping determine when it is listed in the Bundle Builder.</p> <p>If the Audience subtab refers to a custom role, the role is automatically included in the bundle as well.</p> <p>Note that you can also bundle CSV files that are used as templates or to store data to be imported, as File objects.</p> <p>For more information, see the help topic Including Import Maps and CSV Files in SuiteBundler.</p>
Custom Fields	Yes	<p>For each bundled custom field, a preference is available to indicate whether the field should be shown or hidden on existing custom forms when it is installed in target accounts. (Bundled custom fields are always shown on bundled custom forms.) See Bundle Object Preferences.</p> <p>Depending on the custom field, objects referenced by the custom field may or may not be automatically included in the bundle. For more information, see Referenced Objects in Customization Bundles.</p>

Object Type	Lockable?	Comments
Custom Lists	Yes	For each bundled custom list, a preference is available to indicate, on bundle updates in target accounts, whether the bundled object's data should replace the target account object's existing data. This preference is also available when a custom list that conflicts with an existing target account custom list is installed and the installer chooses to replace the existing object. See Bundle Object Preferences .
Custom Records	Yes	<p>When including custom record types in your bundle, you can also include the custom records already created in your account by checking the Include Data box. If the bundled custom record types have List or Record fields that reference a standard list, such as items or customers, then those fields are installed with no data. The locking behavior is the same whether the custom record includes data.</p> <p>For each bundled custom record that includes data, a preference is available to indicate, on bundle updates in target accounts, whether the bundled object's data should replace the target account object's existing data. This preference is also available when an Include Data custom record that conflicts with an existing target account custom record is installed and the installer chooses to replace the existing object. See Bundle Object Preferences.</p> <p>The links on a custom record are preserved only if they refer to standard centers and sections or if the custom centers and sections are also included in the bundle.</p> <p>Online custom record entry forms are included in bundles by default when a custom record type is included.</p>
Custom Segments	Yes	<p>Custom segments are listed under Custom Lists/Records on the Select Objects page of the Bundle Builder.</p> <p>Note: Bundle installation does not copy the Filtered By value for a custom segment to the target account if this value is set to a standard record, such as class, department, location, or subsidiary. If in the target account you manually set the custom segment's Filtered By value to a standard record, a bundle update causes the deletion of this value. Bundle installations and updates delete the Filtered By values in bundled custom segments because these values can be enabled or disabled by a feature setting in the target account.</p> <p>For information about custom segments, see the help topic Custom Segments. For details about current limitations on bundling custom segments, see Adding a Custom Segment to a Bundle.</p>
Custom Transactions	Yes	Before bundling a custom transaction, complete the steps on the pre-bundling checklist. For more information about bundling a custom transaction, see Custom Transaction Types in Bundles .
Dashboards & Centers	No	<p>Dashboard elements that can be bundled include center categories, center tabs, custom centers, custom KPI scorecards, and dashboards.</p> <p>For each bundled custom dashboard, a bundle update preference is available to indicate whether dashboard updates should be applied to target account users to whom the dashboard has been published. See Bundle Object Preferences.</p> <p>When you add a dashboard to a bundle, objects referenced in the dashboard, such as center tabs and saved searches displayed in dashboard portlets, are also included in the bundle.</p>

Object Type	Lockable?	Comments
Extensible Lists	No	<p>Includes the following types of lists:</p> <ul style="list-style-type: none"> ■ Charge Type. Accounting List used by the Recurring Billing SuiteApp. If you uninstall a bundle with a Charge Type object, and another object in the target account references the Charge Type object, the uninstallation continues but NetSuite does not remove the Charge Type object.
File Cabinet Files	Yes	<p>To include a file in a bundle, check the Available for SuiteBundles box on the file record in the File Cabinet and add the file to a bundle in the Bundle Builder.</p> <p>For more information about bundling files and accessing files after bundle installation, see File and Folder Management in Bundles.</p>
File Cabinet Folders	Yes	<p>To include a folder in a bundle, check the Available for SuiteBundles box on the folder record in the File Cabinet and add the folder to a bundle in the Bundle Builder.</p> <p>For more information about bundling files and accessing files after bundle installation, see File and Folder Management in Bundles.</p> <div style="background-color: #ffffcc; padding: 10px;"> <p> Important: Including large folders in a bundle can extend installation time, in some cases causing network connection problems that result in a failed installation. To ensure installation success, you can perform one of the following actions:</p> <ul style="list-style-type: none"> ■ You can divide folder contents into multiple smaller bundles. ■ Instead of transferring folder contents to the target account through a bundle installation, you can export all folder contents from the File Cabinet of the source account to a ZIP file and then upload that ZIP file to the File Cabinet of the target account, where you have installed the bundle. For more information, see the help topics Downloading Files from the File Cabinet and Uploading Files to the File Cabinet. <p>However, even if you exclude some files from bundle installation and import them to the target account through a ZIP file afterward, some objects in the bundle can still reference such previously excluded files and pull them into the target account during bundle installation, thus resulting into duplicate files being installed in the target account.</p> <p>There is also a difference in where the files are installed in the File Cabinet, depending on whether you choose to install the files through a bundle, or import the files through a ZIP file.</p> <p>For example, if you include a file named foo.js in a bundle with an ID of 100, NetSuite installs the file into the following File Cabinet directory in the target account:</p> <pre>/SuiteBundles/Bundle 100/foo.js</pre> <p>However, if you decide to exclude files from bundle installation and import them to the target account through a ZIP file instead, the files imported to the File Cabinet in the target account retain the same folder structure as in the source account.</p> </div>
Forms	Yes	<p>For each bundled custom form, a preference is available to indicate, when it is pushed to target accounts, whether it should be set as the globally preferred form for the record type. See Bundle Object Preferences.</p>

Object Type	Lockable?	Comments
		<p>Important: This preference is applied during push installations only. Unlike other preferences, bundle installers cannot override this preference's Bundle Builder setting.</p> <p>If a translated form in a bundle matches the base language in the target account, that form is used.</p> <p>If you associate an advanced PDF/HTML template with a custom form and then include the custom form in the bundle, the Bundle Builder automatically includes the template.</p> <p>Note: Unlike other objects, adding a custom form to a bundle does not automatically include all of the custom elements included in the form like custom tabs or custom fields. These elements must be added individually.</p> <p>After bundle installation, a custom field in the target account will automatically appear in the custom form if the field applies to the form's record type. For example, when a custom form for the Customer record type is deployed to a target account where a custom field exists that applies to the Customer record type, the custom field automatically appears on that form even though the field was not included in the bundle.</p> <p>Note: When you update a bundle that includes a form that you have updated to be the preferred form for one or more roles, the preferred form setting transfers to a target account only for the roles that are included in that bundle.</p> <p>Note: When a bundled standard customer form is installed in a target account, the Use for Pop-ups field remains unchecked in the target account even if this field is checked on the source account form.</p>
Online Forms	No	<p>The publishable form URL for an installed bundle is updated to reference the target account where it is installed.</p> <p>Any files referenced by an online form (form logo, auto-reply email template, and HTML online form template) are included in the bundled form. If the bundle's installer does not want to use these files, the online form should be updated after installation.</p> <p>The redirect to URL is not carried over from the source account. You must set the redirect to URL after installing a bundle with an online customer form.</p> <p>The lead status and lead source settings for online customer forms are not maintained when the form is bundled. These must be configured manually after the bundled is installed.</p>
Plug-ins	Yes	<p>Include the following plug-in types or plug-in implementations:</p> <ul style="list-style-type: none"> ■ Custom plug-in types and custom plug-in implementations. See the help topic Custom Plug-in Overview. ■ implementations. ■ Email Capture Plug-in implementations.

Object Type	Lockable?	Comments
		<ul style="list-style-type: none"> ■ Bank Statement Parser Plug-in implementations. ■ Financial Institution Connectivity Plug-in implementations.
Reports	No	<p>If a custom report's audience includes custom roles, the custom roles are not included in the bundle by default. You must add the roles to the bundle as well. For details about how bundle updates affect report audience, see Bundle Update Reference.</p> <p>If you include a custom financial statement in a bundle, the custom layout assigned to the report is automatically bundled as well. Whenever possible, the assignment of a layout to a bundled financial statement from the source account is preserved in target accounts. If there is ambiguity about this assignment, it is not preserved in target accounts, and users in these accounts can set up the assignments themselves. See Bundling NetSuite Financial Statements.</p>
Roles	Yes	<p>If a role has custom record permissions, then NetSuite does not automatically include corresponding custom records in the bundle. If you explicitly include custom records in a bundle, any permission levels granted to roles in that bundle are maintained when installed in the target account.</p> <p>If role-based preferences have been defined, these are included with the bundled role.</p> <p>Subsidiaries selected for a role in a OneWorld source account are not included with the bundled role, because subsidiaries are specific to accounts.</p>
Saved Searches	Yes	<p>Only public saved searches can be bundled.</p> <p>If a saved search includes a formula that references a custom field, then NetSuite does not automatically pull the custom field into the bundle. You must explicitly add the custom field to the bundle.</p> <p>If a saved search's audience includes custom roles, the custom roles are included in the bundle automatically. For details about how bundle updates affect search audience, see Bundle Update Reference.</p> <p>If a saved search is filtered by a custom transaction type value, the custom transaction type is included in the bundle automatically.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> Note: If a saved search with an email alert defined is included in a bundle, note that in target accounts where the bundle is installed, the email alert is not sent unless the Enable Email Alerts for WS and CSV Imports preference is enabled, at Setup > Company > Email Preferences. </div>
SSP Applications	Yes	<p>An SSP application implements Web store customizations .</p> <p>All files in an SSP application are grouped together in a folder in the source account's NetSuite File Cabinet. An SSP Application record indicates the touch points where pages in the SSP application should be substituted for default Web store pages.</p> <p>When a bundled SSP application is installed in target accounts, it is hosted from the same location in each target account File Cabinet. The installed bundle includes touch points, but SSP application pages are not used in the Web store until touch points are deployed to the site. See the help topic Bundle an SSP Application.</p>

Object Type	Lockable?	Comments
		<p>Note: As of 2019.1, when you update an installed bundle that includes an SSP application, the script deployments are not updated in the target account. Because script deployments are not updated, the existing touch points are not overwritten and both the Log Level and Status fields are also not updated in the target account. The Log Level and Status fields can be viewed on the Script Deployment page.</p>
Sublists	No	
Subtabs	No	
SuiteScripts	Yes	<p>You can check the Hide in SuiteBundle box on the file record for a server script file to prevent those who download your bundle from viewing your server SuiteScript files. You cannot hide client SuiteScripts in a bundle. You can only bundle a client SuiteScript if you also include the associated form in the bundle. See Protecting Your Bundled Server SuiteScripts.</p> <p>You should make script parameters settable as preferences. This insures that updates to your bundle do not change the way scripts are set up in the target account. Also, allowing users to set parameters as preferences can help users in the target account to set up the bundle. For more information about scripting parameters, see the help topic Creating Script Parameters.</p> <p>Locking the script record does not lock its deployments. The Bundle Builder pages do not display script deployments. Therefore, the bundle author cannot specify which deployments are locked.</p> <p>Be aware that the order in which bundled scripts run in the source account is maintained when the bundle is installed in target accounts. You can set up script execution order for client and user event scripts on the Scripted Records page, at Customization > Scripting > Scripted Records. See the help topic The Scripted Records Page.</p> <p>For each bundled SuiteScript, a preference is available to indicate whether the script's deployments should be updated in target accounts when bundle updates occur. See Bundle Object Preferences.</p> <p>If a script deployment's audience includes custom roles, the custom roles are included in the bundle automatically. For details about how bundle updates affect script deployment audience, see Bundle Update Reference.</p> <p>You cannot attach a SuiteScript 2.0 script file to more than one bundle. If you create multiple bundles and the bundles include the same SuiteScript 2.0 script file, your script may throw a Module does not exist exception. NetSuite is instituting this restriction to avoid the following scenario: You have the same script file bundled in both SuiteApp A and SuiteApp B. If SuiteApp B is upgraded, then the script file moves to SuiteApp B. The old location (module path) to SuiteApp A no longer works. This triggers an exception when the script in SuiteApp A attempts to execute.</p>
		<p>Important: Bundle installation scripts have different requirements and characteristics than other scripts included in bundles. For details, see Using Bundle Installation Scripts.</p>
SuiteSignOn	No	<p>Each Outbound Connection object includes the data entered on the SuiteSignOn page used to implement single sign-on integration between NetSuite and an external application. This data includes the name of the NetSuite user interface element where the connection is implemented, the</p>

Object Type	Lockable?	Comments
		<p>external application landing page, optional data that sets context for the landing page, and optional user identification data.</p> <p>Any custom subtabs and portlet scripts defined as connection points, and any custom fields defined as user identification, are automatically included with the SuiteSignOn Outbound Connection object.</p> <p>If any custom fields are defined as integration variables, you must explicitly add these fields to the bundle.</p> <p>For more information, see the help topic Creating a SuiteSignOn Bundle.</p>
Translation Collections	Yes	<p>Translation Collections are custom objects comprised of translated terms. The Translations Collection object includes terms and their translations in the specified locale. For more information about Translation Collections, see the help topics Translation Collections Overview and Creating a Translation Collection.</p> <p>After installation, the bundle ID displays in the From Bundle field when the Translation Collection is viewed or edited in the target account.</p> <p>The Multi-Language feature must be enabled in the source account, to support the bundling of Translation Collection objects. If that feature is not enabled in the target account, the objects are installed and scripts can access them, although the Translation Collections are not viewable in the NetSuite UI.</p> <p>During bundle installation, the script ID for a Translation Collection object is automatically renamed when there is a script ID conflict but the default locale is different. Unlike other script ID conflicts, there is no option to replace the existing object during bundle installation.</p> <p>As of 2020.2, SuiteBundler supports enhancements to custom object translatable fields and Translation Collections. For more information, see the help topic Translatable Fields Supported in SDF Custom Objects.</p> <p>This support imposes the following limitations on bundles that include a Translation Collection and custom objects that contain translatable fields that reference translation strings from that Translation Collection:</p> <ul style="list-style-type: none"> ■ You cannot install these bundles from a source account running 2020.2 to a target account running 2020.1, or from a source account running 2020.1 to a target account running 2020.2. ■ You can only install such bundles to target accounts running the same NetSuite version as the source account.



Note: These limitations do not apply to bundles that include only Translation Collections.

Consider the following when modifying Translation Collections in a bundle:

- If you delete a translation string from a bundled Translation Collection in a source account, and later you update that bundle in a target account:
 - All references in custom object translatable fields to that deleted translation string from the Translation Collection are unlinked from that deleted translation string in the target account. The unlinked translation string becomes a standalone translation string.
 - Custom object translatable fields retain the original translation string, including the translations, in the target account.

Object Type	Lockable?	Comments
		<ul style="list-style-type: none"> ■ If you update a translation string in a bundled Translation Collection in a source account, and a custom object translatable field in a target account that references that translation string fails validation: <ul style="list-style-type: none"> □ The custom object translatable field is not updated in the target account and is unlinked from the translation string in the Translation Collection. The unlinked translation string becomes a standalone translation string. □ The custom object translatable field retains the original translation string, including the translations, in the target account. <p>For more information about translation strings, see the help topic String Hierarchy and Translation String Linking.</p>
Web Site Customizations	No	Website customizations that can be bundled include color themes, item/category templates, layouts, tags, and themes.
Workflows	Yes	<p>If a bundled workflow is locked, you may not edit the definition, instance fields, states, transitions, or actions in a target account. Also, you may not add transitions or states to a locked workflow.</p> <p>Workflow instances and history logs are not copied to target accounts when a bundle is installed or updated.</p> <p>For more information about bundling a workflow, see the help topic Bundling a Workflow.</p>

Referenced Objects in Customization Bundles

For most objects, the Bundle Builder automatically adds all referenced objects when you add the object to a bundle. However, the Bundle Builder does not automatically include referenced objects for all object types. If a referenced object can be included in a bundle and the Bundle Builder does not automatically add the referenced object, you must manually add it.

Refer to the following sections for more information about the referenced objects that are automatically added to a bundle and the referenced objects that are not automatically included.

- [Objects Automatically Included In Customization Bundles](#)
- [Objects Not Automatically Included In Customization Bundles](#)



Note: The effect of not adding the referenced objects depends on the type of object. Some referenced objects are not included because they cannot be included in bundles. Other objects can have the reference to the object nulled out in the target account or use objects present in the target account. For objects like custom forms, reports, and saved searches, you must include the referenced objects for the object to function correctly in the target account. For more information, see [Objects Not Automatically Included In Customization Bundles](#).

Objects Automatically Included In Customization Bundles

The following table lists objects that you can select in the Bundle Builder and the referenced objects that are automatically included.



Note: If SuiteBundler adds a referenced object to a bundle, you can view the referenced object on the Set Preferences page in the Bundle Builder. For more information, see [Step 4 Set Preferences](#).

Object Selected in Bundle Builder	Referenced Objects Included	Referenced Objects Description
Custom Field	Customized Object	<p>Any object in an account that is referenced by a custom field and has an object type that can be bundled.</p> <p>For example, you create a custom field that references a saved search and include the custom field in a bundle. The saved search is also included in the bundle.</p>
Custom Financial Statement	Layout	<p>Custom layout associated with a custom financial statement.</p> <p>For more information, see Bundling NetSuite Financial Statements.</p>
Custom Record Type	Custom Online Form	Online custom record entry forms for the custom record type.
Dashboard	Center Tab Custom Record Type Saved Search Script File	<p>Any bundleable objects referenced by the dashboard are included. These may include scripts, saved searches, or custom records used for dashboard portlet display.</p> <p>Note: Other referenced objects may also be included.</p>
Online Form	Auto-reply Email Template Form Logo HTML Online Form Template	Files referenced by the online form. If the installer of the bundle does not want to use the files, the online form must be updated after installation.
Saved Custom Record Search	Custom Field Custom Record Type	Custom record type on which the search is based and all custom fields created for that record type.
Saved Search	Custom Transaction	If a saved search is filtered by a custom transaction type value, the custom transaction type is included in the bundle automatically.
Script Deployment	Custom Role	Custom roles included in the audience for the script deployment.
SuiteScript Record	Script File	<p>Script file for the SuiteScript record.</p> <p>Note: If you lock the SuiteScript record in the bundle, NetSuite does not automatically lock the script file in the target account. The script file must be manually locked in the target account.</p>
SuiteSignOn Outbound Connection Object	Custom Field Custom Subtab	Custom subtabs and portlet scripts defined as connection points and custom fields defined as user identification.

Object Selected in Bundle Builder	Referenced Objects Included	Referenced Objects Description
	Portlet Script	
Workflow	Custom Field Custom Role	Custom roles referenced by the workflow and custom fields referenced by the workflow.

Objects Not Automatically Included In Customization Bundles

The following table lists objects that you can select in the Bundle Builder, the referenced objects that must be manually added, and the effect of not including the referenced object in the bundle. In some cases, the referenced object cannot be included and the object that you include must be edited in the target account after you install the bundle.

Object Selected in Bundle Builder	Referenced Objects Not Included	Referenced Objects Description
CSV Import Map	Custom Role	Custom roles included in the audience for a CSV import map.
Custom Field	Customized Object	<p>Any object in an account that is referenced by a custom field and has an object type that cannot be bundled. For example, entities, items, transactions, or objects in a list, like Employee Type, cannot be bundled.</p> <p>If a custom field references this type of object, the reference is nulled out in the target account.</p> <p>Edit the custom field in the target account and recreate the reference after you install the bundle.</p>
Custom Field	Standard List	<p>Any standard list object referenced by a custom field, for example, Approval Status.</p> <p>Because the data in standard lists are the same in every account, installing the bundle only copies the list data from the bundle to the target account. The list object itself is not included in the bundle.</p> <p>You do not need to take any action after you install the bundle.</p>
Custom Form	Custom Field Custom Tab	<p>Custom objects on a custom form.</p> <p>You must manually add any custom object referenced in a custom form to the bundle. Otherwise, the custom form may appear incorrectly in the target account.</p>
Custom Report	Custom Role	<p>Custom role included in the audience for a custom report.</p> <p>You must manually add any custom role included in the audience of a report to the bundle. Otherwise, the report may not be available to the intended audience.</p>
Role	Custom Record	<p>If a role contains permissions for a custom record type, the corresponding custom record type is not included in the bundle.</p> <p>You must manually add the custom record to the bundle. Otherwise, the custom role will contain permission for a record type that does not exist in the target account.</p> <p>Subsidiaries selected for a role in a OneWorld source account are not included with the bundled role, because subsidiaries are specific to accounts.</p>

Object Selected in Bundle Builder	Referenced Objects Not Included	Referenced Objects Description
Saved Search	Custom Field	If a formula in a saved search references a custom field, the custom field is not included in the bundle. You must manually add any custom field referenced by a formula in a saved search. Otherwise, the saved search may not function correctly.
SuiteSignOn Outbound Connection Object	Custom Field	Custom field referenced as an integration variable in a SuiteSignOn outbound connection object. You must manually add any custom field referenced by the SuiteSignOn outbound connection object. Otherwise, the custom field will not exist in the target account and the single sign-on integration between NetSuite and the external application fails.

Configuration Bundles

Configuration bundles allow administrators to copy setup and configuration settings from one account to another. NetSuite vertical solution providers and independent software vendors (ISVs) can use configuration bundles to automate the setup and configuration of a customer account. The supported settings include company preferences, accounting lists, customer relationship management lists, record names, and transaction names. A configuration bundle can also enable or disable features in target accounts.

Note that you cannot include data, such as a chart of accounts, in a configuration bundle. If an object that you install as part of a bundle requires data, then you must import the required data separately from the bundle.



Important: You should install configuration bundles immediately after provisioning an account. This ensures that you do not overwrite existing production account settings. If your solution requires specific setup and configuration parameters that you plan to roll out to existing customers, be sure to work with each account administrator on an impact analysis prior to installing the bundle.

Objects Available in Configuration Bundles

The following table lists the types of objects that you can include in a configuration bundle. These objects are preferences and configuration settings. For a description of an object type's function, go to the path shown and see the help for that page.

Object Type	Object	Navigation Path
Auto-Generated Numbers	<p> Note: Updating a configuration bundle does not change existing auto-generated numbers.</p>	
	CRM	Setup > Company > Setup Tasks > Auto-Generated Numbers
	Entities	Setup > Company > Setup Tasks > Auto-Generated Numbers
	Schedules	Setup > Company > Setup Tasks > Auto-Generated Numbers

Object Type	Object	Navigation Path
	Transactions	Setup > Company > Setup Tasks > Auto-Generated Numbers
Company Preferences		
	Approval Routing	Setup > Accounting > Preferences > Accounting Preferences
	CSV Import	Setup > Integration > Manage Integration > SOAP Web Services Preferences
	Email	Setup > Company > Preferences > Printing, Fax & Email Preferences
	External	Setup > Support > Preferences > Support Preferences
	Fax	Setup > Company > Preferences > Printing, Fax & Email Preferences
	Forecasts	Setup > Sales > Preferences > Sales Preferences
	General Accounting	Setup > Accounting > Preferences > Accounting Preferences
	General Company	Setup > Company > Preferences > General Preferences
	Note: Override settings for the preferences on the Overriding Preferences subtab are preserved if the matching preferences are a part of the bundle.	
	General Issues	Setup > Issues > Preferences > Issue Preferences
	General SFA	Setup > Sales > Preferences > Sales Preferences
	General Support	Setup > Support > Preferences > Support Preferences
	Inbound Email	Setup > Marketing > Preferences > Marketing Preferences
	Inventory Management	Setup > Accounting > Preferences > Inventory Management Preferences
	Invoicing	Setup > Accounting > Preferences > Invoicing Preferences
	Items / Transactions	Setup > Accounting > Preferences > Accounting Preferences
	Marketing	Setup > Marketing > Preferences > Marketing Preferences

Object Type	Object	Navigation Path
	Notification	Setup > Accounting > Preferences > Accounting Preferences
	Notifications	Setup > Issues > Issue Preferences
	Order Management	Setup > Accounting > Preferences > Accounting Preferences (Administrator)
	Printing	Setup > Company > Preferences > Printing, Fax & Email Preferences (Administrator)
	Shipping	Setup > Accounting > Setup Tasks > Shipping (Administrator)
	Time & Expenses	Setup > Accounting > Preferences > Accounting Preferences (Administrator)
	Web Services	Setup > Integration > Manage Integrations > SOAP Web Services Preferences(Administrator)
Features		
	Company	Setup > Company > Setup Tasks > Company Information (Administrator)
	Accounting	Setup > Accounting > Preferences > Accounting Preferences (Administrator)
	Tax	Setup > Company > Setup Tasks > Enable Features
	Transactions	Setup > Sales > Preferences > Sales Preferences (Administrator)
	Items & Inventory	Setup > Accounting > Preferences > Accounting Preferences (Administrator)
	Employees	Setup > Users/Roles > User Management > Manage Users (Administrator)
	Customer	Setup > Marketing > Preferences > Marketing Preferences (Administrator)
	Relationship	
	Management	
	Web Presence	Setup > Company > Setup Tasks > Enable Features > Web Presence
	SuiteCloud	Customization > Forms > Transaction Forms (Administrator)

Object Type	Object	Navigation Path
Other Lists		
	Accounting Lists	Setup > Accounting > Setup Tasks > Accounting Lists (Administrator)
	Customer Relationship Management (CRM) Lists	Setup > Sales > Preferences > Sales Preferences (Administrator)
Renaming		
	Account Type Names	Setup > Company > Setup Tasks > Rename Records/ Transactions (Administrator)
	Record Names	Setup > Company > Setup Tasks > Rename Records/ Transactions (Administrator)
	Transaction Names	Setup > Company > Setup Tasks > Rename Records/ Transactions (Administrator)
i Note: If a configuration bundle includes any settings that are inactive in the source account for the bundle, these settings are not installed in bundle target accounts. Configuration bundles cannot be managed bundles.		

Creating a Bundle with the Bundle Builder

You can use the Bundle Builder to create bundles. Bundles are packages of customization or configuration objects that can be installed in other NetSuite accounts. The Bundle Builder is available at Setup > Customization > SuiteBundler > Create Bundle if the SuiteBundler feature is enabled in your account and you have the SuiteApp Marketplace permission.

i Note: Before you can create a bundle, an administrator must enable the Create Bundles With SuiteBundler feature at Setup > Company > Enable Features.

Use the following information to create a bundle in the Bundle Builder:

Step	Description
Step 1 Bundle Basics	Enter basic definitions for the bundle.
Step 2 Bundle Properties	Enter optional descriptive information that helps users to understand the purpose of the bundle and helps them to find the bundle when they search for it.
Step 3 Select Objects	Select the objects to be included in the bundle.
Step 4 Set Preferences	Review the bundle objects, including objects you select and any objects they reference, and save the bundle. For customization bundles, set preferences for objects to be applied during bundle installation and update. You can also lock some types of objects so that target account users cannot make changes to them.
Finish the Bundle Builder	Set up sharing or review a list of all saved bundles.

If the Managed Bundles feature is enabled in your account, you can create managed bundles. With this type of bundle, you determine when to update target accounts with new versions of the bundle. See [Managed Bundles Overview](#).

Step 1 Bundle Basics

On the **Bundle Basics** page of the Bundle Builder, enter basic definitions for a bundle.

The following screenshot shows the **Bundle Basics** page:

The screenshot shows the 'Bundle Basics' step of the SuiteBundler 'Create Bundle' wizard. The progress bar at the top indicates Step 1 is active. The main form contains the following fields:

- Name ***: A required text input field.
- Version**: An optional text input field.
- Abstract**: A large text area for abstract text.
- Installation Script**: A dropdown menu.
- Owner**: Pre-filled with 'A Wolfe-admin'.
- Created on**: Pre-filled with '7/21/2014'.
- Last Updated**: A timestamp.
- Checkboxes** (under 'Last Updated'):
 - Configuration Bundle
 - Managed Bundle
 - Bundle All
 - Hide Components

At the bottom are 'Cancel', '< Back', and 'Next >' buttons.

To complete Step 1 Bundle Basics in the Bundle Builder:

1. Log in to the account where you want to create a bundle.

You should consider the bundle deployment strategy before you begin creating a bundle, so that you create the bundle in the appropriate account. For information about bundle deployment, see [For explanations of deployment-related terms and preferred deployment methods, see SuiteApp Development Process with SuiteBundler](#).

2. Go to Customization > SuiteBundler > Create Bundle.
3. Enter the following properties:

Property	Description
Name	Name of the bundle. This is the only required field.
Version	Bundle version used to track future updates. You can use any numbering scheme in this field. This is an optional field, but you should give each bundle a version number to be displayed on bundle list pages. This number is vital to help you manage updates. SuiteBundler alerts your install base to version updates with an icon on bundle list pages and a SuiteBundles to Update reminder in the Reminders portlet; these alerts do not occur if you do not use version numbers.

Property	Description
Abstract	<p>Brief description of the bundle. This description is displayed in the Abstract column of the Installed Bundles page and Saved Bundles pages, and on the Bundle Details page. This is an optional field.</p> <p>You can enter a longer description on the next page in the Bundle Builder, Bundle Properties.</p> <p>For more information, see Bundle Details.</p>
Installation Script	<p>Specialized SuiteScript to be associated with the bundle. A bundle installation script fires triggers that execute as part of bundle installation, update, or uninstall. These triggers automatically complete required setup, configuration, or data management tasks, depending on the bundle. Use this option if you require additional operations in addition to installing the bundle.</p> <p>This script is optional and is not required to install the bundle.</p> <p>See Using Bundle Installation Scripts.</p>

4. If necessary, check the following boxes:

Box	Description
Configuration Bundle	<p>Indicates the bundle is a configuration bundle that contains setup and configuration objects. When you install a configuration bundle in a target account, it sets configuration parameters, enables or disables features, and sets company preferences.</p> <p>If you check this box, the Select Objects page displays only setup and configuration object types. See Configuration Bundles.</p> <p>When this box is checked, the Managed Bundle box is disabled because configuration bundles cannot be managed.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i Note: You cannot create a bundle that contains both customization and configuration objects. If your solution requires both types of objects, you must create and distribute two distinct bundles. </div>
Managed Bundle	<p>Indicates that the upgrade process for the bundle should be controlled by you. When this option is enabled, you can choose when to push updates to accounts that have previously installed the bundle. This control over bundle updates enhances your ability to manage your install base.</p> <p>You must have the Managed Bundles feature is enabled in your account to use this option. See Managed Bundles Overview.</p>
Bundle All	<p>Auto-selects all available objects for the bundle. All objects are listed under Bundle Contents on the Select Objects page, and you can remove objects from the bundle as necessary, rather than adding them.</p> <p>This option may save time, if it is quicker to de-select the objects that you do not want to include in the bundle rather than select the objects that you want to include.</p>
Hide Components	<p>Indicates that the bundle objects do not display in target account lists. By default, bundled objects installed in target accounts display in object list pages.</p> <p>See Hiding Bundle Components in Target Accounts.</p>

5. If you are creating a bundle for an SDF SuiteApp project, select the correct value for the **SuiteApp ID** property.

Property	Description
SuiteApp ID	<p>Lists SDF SuiteApp projects in your account. You can select a SuiteApp project to be bundled. The objects included in the SuiteApp project are added to the bundle and are shown on the Select Objects page of the Bundle Builder.</p>

The selected SuiteApp ID associates a specific publisher, or owner, with the bundle. To obtain a unique publisher ID, you must register with SDN. For information about SDN, go to <http://www.netsuite.com/portal/developers/sdn.shtml>.

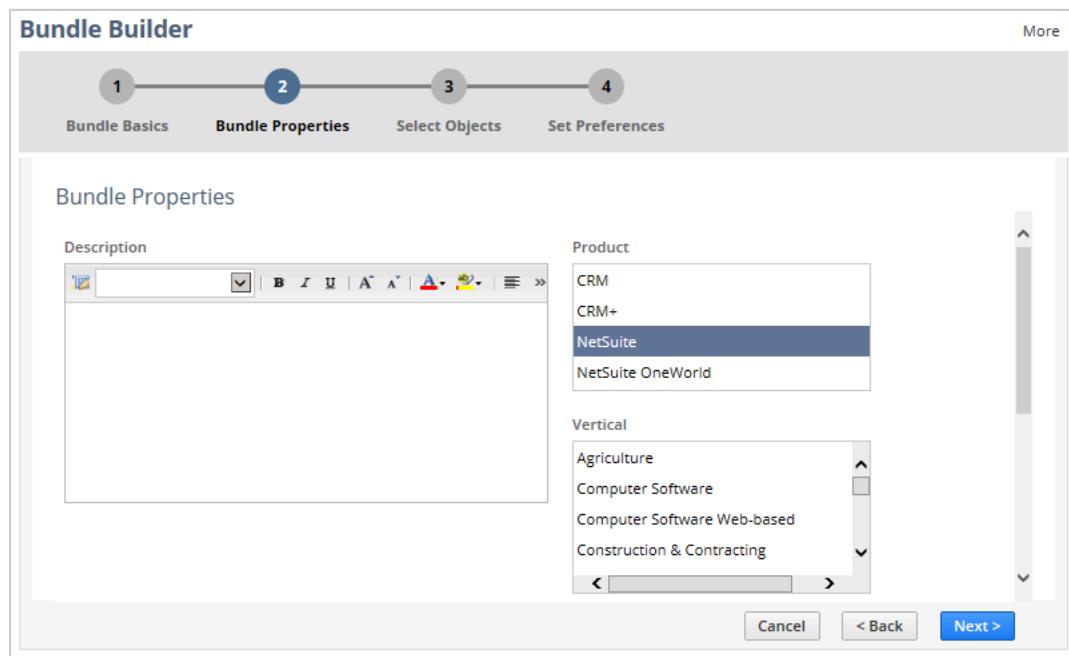
For information about using SDF, see the help topic [SuiteCloud Development Framework Overview](#).

6. Click **Next**.

Step 2 Bundle Properties

On the Bundle Properties page of the Bundle Builder, enter optional descriptive information that helps users to understand the purpose of the bundle and help users find the bundle in a search.

The following screenshot shows the **Bundle Properties** page:



To complete Step 2 Bundle Properties in the Bundle Builder:

1. If you have not already done so, complete [Step 1 Bundle Basics](#).
2. On the **Bundle Properties** page, enter the following optional bundle metadata information:

Property	Description
Description	Information about the bundle, in addition to the Abstract . This information is displayed on the Bundle Details page.
Product	NetSuite product for which this bundle is created. The default is the product used in the source development account where this bundle is created.
Vertical	Target vertical market for this bundle, for example, IT Services . Customers can use this searchable field on the Search & Install Bundles page to identify solutions appropriate for their business needs.
Language	Languages in which the bundle is available. Customers can use this searchable field on the Search & Install Bundles page to identify solutions appropriate for their locale and business needs. The default is the base language of the source account.

3. Optionally, in the **Admin Documentation** field, select the file with details about your bundle, or enter a URL to link to an externally hosted site that provides this information.

After administrators install this bundle, they can open the documentation file or access the URL from the **Installed Bundles** or **Bundle Details** pages in the target account. This documentation should be aimed at administrators. Another mechanism is available to publish user documentation in the NetSuite Help Center. See [Publishing SuiteApp Help Content to the NetSuite Help Center](#).

For more information about the steps to set this property, see [Documenting a Bundle](#).

4. Optionally, click the arrow next to **Terms of Service** and define bundle-specific terms of service.

If you check the **Requires Acceptance of Terms** box, users must accept these terms before they install or update the bundle. The terms of service are in addition to the default NetSuite terms of service for bundles. If you do not check this box, users will not be required to accept terms and the link to these terms is not displayed.

For more information about this property, see [Defining Bundle Terms of Service](#).

5. Click **Next**.

Step 3 Select Objects

On the **Select Objects** page of the Bundle Builder, select the objects to be included in the bundle. The available objects vary, depending on whether you are creating a customization bundle or a configuration bundle. See [Objects Available in Customization Bundles](#) and [Objects Available in Configuration Bundles](#).

To complete Step 3 Select Objects in the Bundle Builder:

1. If you have not already done so, complete [Step 2 Bundle Properties](#).
2. In the **Object Types** list, expand the folder for the type of object that you want to include in the bundle.

The **Choose Objects** list displays available objects of the selected type, if any.

3. In the **Choose Objects** list, check the box next to each object you want in the bundle, or check the **Select All** box to check all of the objects listed for the object type.

Each time you check a box, NetSuite updates the **Bundle Contents** list.



Warning: Enabling the **Include Data** option for a bundled custom record object with an extremely large number of custom records may result in bundle installation failure due to network connection problems. To ensure installation success, do not enable the **Include Data** option for the custom record object. Then run a bundle installation script that executes after installation to import custom record object data into the target account through CSV import. For information, see the help topics [SuiteScript 2.x Bundle Installation Script Type](#) and [task.CsvImportTask](#).



Note: If there are existing custom records for a bundled custom record object, check the **Include Data** box under the object to include this data in the bundle. Even if you do not check the **Include Data** box, NetSuite may still pull data for a custom record, if another object in the bundle depends on the data in the custom record.

4. To de-select an object, select its object type in the **Bundle Contents** list and clear its box in the **Choose Objects** list.
5. Click **Next**.

General Guidelines for Selecting Objects

- When you add most types of custom objects to a bundle, all of the necessary elements referenced by that object are also automatically added to the bundle, so you do not need to explicitly add them. However, some referenced objects are not added automatically. For more information, see [Referenced Objects in Customization Bundles](#).
- When you create a customization bundle, you should enter a unique script ID for each object in the bundle, to avoid conflicts during installation. If the ID for a bundle object matches an object in the target account, the bundled object's ID is appended with a number. For more information, see [Resolving Conflicting Objects](#).
- If your bundle includes server SuiteScripts, you can check the Hide in SuiteBundle box on their File Cabinet file records to prevent users who install the bundle from viewing the script files. See [Protecting Your Bundled Server SuiteScripts](#). Client SuiteScripts cannot be hidden.
- If you see configuration objects (Company Preferences, Features, Other Lists, and Renaming) on the Select Objects page, you have set the bundle to be a configuration bundle. To define the bundle as a customization bundle, go back to the Bundle Basics page and clear the **Configuration Bundle** box.
- Including a configuration object in a configuration bundle does not create a new instance of that object in the target account when the bundle is installed. Configuration bundle installation simply copies the settings for included configuration objects from the source account to the target account.

Step 4 Set Preferences

On the **Set Preferences** page of the Bundle Builder, you can review the bundle objects, including objects you select and any objects they reference, and save the bundle. For customization bundles, set preferences for objects to be applied during bundle installation and update. You can also lock some types of objects so that target account users cannot make changes to them.

The following screenshot shows the **Set Preferences** page:

NAME	ID	REFERENCED BY	PREFERENCE	LOCK ON INSTALL
Custom Fields				
CRM Fields				
• Frequent Support User?	custevent1_1		Hide on Existing Custom Forms ▾	<input type="checkbox"/>
Custom Lists/Records				
Lists				
• Help Use Occurrence (4 records)	customlist7		Preserve Data ▾	<input type="checkbox"/>
Records				
• Training Level (include 3 records)	customrecord_level		Preserve Data ▾	<input type="checkbox"/>
File Cabinet				
Files				
• field_changed.js	70	Phone Call Form - Automation		<input type="checkbox"/>

To complete Step 4 Set Preferences in the Bundle Builder:

1. If you have not already done so, complete [Step 3 Select Objects](#).
2. Review the contents of your bundle. You can take the following actions for the objects included in the bundle:
 3. To add or remove objects, click **Back** to return to the **Select Objects** page and make changes.
 4. For dashboards, custom fields, custom lists, custom records that include data, custom forms, and SuiteScript files, review the settings in the **Preference** column, and change as desired. These settings are applied for managed bundles and other push installations and updates, and are defaults for other bundle installations and updates. See [Bundle Object Preferences](#).
 5. If you are creating a customization bundle, check boxes in the **Lock on Install** column to lock selected bundle objects, or click **Lock All** to lock all lockable objects in the bundle. See [Locking Objects in Customization Bundles](#).

For example, an independent software vendor (ISV) may distribute a customization bundle to its customers. To minimize the potential for support and upgrade issues, the ISV may choose to lock objects in the bundle. Locking also protects the ISV's customers from changing objects in a bundle and then discovering they cannot upgrade the bundle without breaking the changed objects.

You cannot lock some objects in customization bundles. Objects that are not lockable do not have a box in the **Lock on Install** column.

Locking an object does not prevent target account users from seeing it, only from editing it. To hide bundle objects in target accounts, see [Hiding Bundle Components in Target Accounts](#). To prevent target account users from seeing code in bundled server SuiteScripts, see [Protecting Your Bundled Server SuiteScripts](#).

6. After you are done making changes, click **Save**.

Finish the Bundle Builder

After you complete [Step 4 Set Preferences](#) in the Bundle Builder and Bundle Builder creates the bundle, the **Finished** page displays. On this page, click the link to set up sharing, click the link to review a list of all saved bundles, or click the link to publish user help content for the bundle in the NetSuite Help Center.

The following screenshot shows the **Finished** page.

Bundle Builder

STEPS

- ✓ Bundle Basics
- ✓ Bundle Properties
- ✓ Select Objects
- ✓ Set Preferences

Finished

Your bundle has been created!

What's Next?

Frequently, users do one of the following after completing this task:

Modify availability of your new bundle By default, new bundles are created as private - only people with administrator roles in your account may access them. But you can share bundles with other accounts and even make them publicly available to everyone.	Review the list of saved bundles View all bundles created from this account.	Publish Help Content Add user documentation to the Help Center
---	--	--

On this page, you can complete the following tasks:

- Click **Modify availability of your new bundle** to go to the **Bundle Availability** page, where you can set up bundle sharing, deprecate the bundle, and set the bundle availability.

You can also access the **Bundle Availability** page after you complete the Bundle Builder from the **Saved Bundles** page by choosing the **Set Availability** option from the **Action** menu.

See [Bundle Availability](#).

- Click **Review the list of saved bundles** to display the **Saved Bundles** page.

You can also access the **Saved Bundles** page at Customization > SuiteBundler > Create Bundle > List.

See [Reviewing the Saved Bundles List](#).

- Click the **Publish Help Content** link to publish user documentation for your bundle in the NetSuite Help Center. See [Publishing SuiteApp Help Content to the NetSuite Help Center](#).

Bundle Builder Reference

Use the following information to get more information about creating bundles in the Bundle Builder:

Topic	Applicable Bundle Builder Step
Hiding Bundle Components in Target Accounts	Step 1 Bundle Basics
Using Bundle Installation Scripts	
Bundle Support during NetSuite Release Phasing	
Documenting a Bundle	Step 2 Bundle Properties
Defining Bundle Terms of Service	

Topic	Applicable Bundle Builder Step
Protecting Your Bundled Server SuiteScripts	Step 3 Select Objects
File and Folder Management in Bundles	
Bundling NetSuite Financial Statements	
Bundle Object Preferences	Step 4 Set Preferences
Locking Objects in Customization Bundles	

Hiding Bundle Components in Target Accounts

By default, list pages for bundleable objects include objects that have been installed by bundles. You can identify an object that is from a bundle, because the ID of the bundle is displayed in the From Bundle column of the list page. See [Identifying Bundle Objects in Target Accounts](#).

In some cases, you may prefer that objects installed by your bundle are not visible in target account list pages.

To enable an option not to display bundle objects in target account list pages, check the Hide Components box on the Bundle Basics page of the Bundle Builder:

The screenshot shows the 'Bundle Basics' step of the Bundle Builder. The top navigation bar has four steps: 1. Bundle Basics (highlighted), 2. Bundle Properties, 3. Select Objects, and 4. Set Preferences. The main area is titled 'Bundle Basics' and contains fields for 'Name *' (with a required asterisk), 'Version', 'Abstract', and 'Installation Script'. To the right, there are details like 'Owner' (A Wolfe-admin), 'Created on' (7/21/2014), and 'Last Updated'. Below these are three checkboxes: 'Configuration Bundle', 'Bundle All', and 'Hide Components', which is the one highlighted with a red box. At the bottom are 'Cancel', '< Back', and 'Next >' buttons.

When this box is checked for a bundle, by default its components are not displayed in target account list pages. Also, the bundle is not listed in the From Bundle filter on list pages. See [Filtering Bundle Objects on List Pages](#). Note that this option cannot be enabled for a configuration bundle.

After bundle installation in a target account, an administrator can change this hide or show setting. This change can be made by selecting the Hide Components or Show Components option button on the Components subtab of the Bundle Details page.

NAME	ID	REFERENCED BY	LOCK ON INSTALL
field_changed_a.js	2865		
ClientAssistant.js	2866		
field_changed_b.js	2864		



Important: The target account selection of the Hide Components or Show Components display option for a bundle is not overwritten by the Bundle Builder setting during bundle updates. The Bundle Builder Hide Components option is applied only during bundle installation.



Warning: If Hide Components is enabled for a bundle created in a production account, refreshes to Release Preview and sandbox accounts cause that bundle's components to be hidden in the Release Preview and sandbox accounts.

Using Bundle Installation Scripts

Each bundle can include a specialized server SuiteScript that is automatically run when the bundle is installed, updated, or uninstalled. A bundle installation script can contain triggers to be executed before install, after install, before update, after update, or before uninstall. This trigger code can ensure that bundles are implemented and removed correctly and efficiently, and can prevent bundle installation, update, or uninstall if proper setup has not occurred. Note that this type of script can be run in any target account where the associated bundle is installed, even if scripting features are not enabled in that account.

To associate a bundle installation script with your bundle, you need to create the script .js file, upload the script file to the NetSuite File Cabinet, create a script record, and define one or more deployments for the script.

After a bundle installation script has been created and at least one deployment has been defined for it, you can associate the script with a bundle by selecting from the Installation Script list on the [Step 1 Bundle Basics](#) page. A single bundle installation script can be associated with multiple bundles. When you associate a script with a bundle, you select a specific script deployment.

Detailed documentation about bundle installation scripts is available in the SuiteScript section of the Help Center:

- For step-by-step instructions for developing bundle installation scripts and associating them with bundles, see the SuiteScript help topic [Setting Up a Bundle Installation Script](#).
- For more details about bundle installation scripts, see the SuiteScript help topic [What are Bundle Installation Scripts?](#).
- To review a sample script, see the help topic [Sample Bundle Installation Script](#).



Note: On the [Step 3 Select Objects](#) page of the Bundle Builder, you do not have to explicitly add the bundle installation script. This script record and the related .js file are included automatically in the bundle, as are any other .js files that are listed as library script files on the script record.

Bundle Support during NetSuite Release Phasing

Upgrades of NetSuite accounts to each new release occur as a phased process. During phasing of a new release, some accounts are still using the lagging version of NetSuite during the time that other accounts have been upgraded to the leading version. During this time period, you are likely to be using a different NetSuite version than some of the accounts where your bundles are installed, so you need to be aware of the following:

- Bundles developed with the lagging version can be installed into accounts that are already using the leading version.
- Bundles developed with the leading version can be installed into accounts that are still using the lagging version, but note the following limitations:
 - When a user in a lagging version account installs a bundle from a leading version account, any object types in the bundle that are not supported in the lagging version are not installed, resulting in potentially serious impact to bundle function.
 - A bundle with a SuiteScript that uses a new API available only in the leading version can be installed into a lagging version account, but the script may not function correctly because the lagging version does not support the API.

You should avoid distributing bundles containing any new leading version API to accounts still using the lagging version. To accomplish this purpose, you can create a bundle installation script that uses the **getVersion()** method to check the target account version, and prevents installation of the associated bundle if the lagging version is in use. See [Using Bundle Installation Scripts](#) and the SuiteScript help topic [Setting Up a Bundle Installation Script](#).

Documenting a Bundle

For every bundle you create, you should attach a document that describes the purpose of the bundle, how to install and use it, and where bundle users can get support when needed. This document should be aimed at administrators. There is a separate mechanism you can use to publish user documentation to the NetSuite Help Center. See [Publishing SuiteApp Help Content to the NetSuite Help Center](#).

You specify a bundle documentation file in the **Admin Documentation** field, on the [Step 2 Bundle Properties](#) page of the Bundle Builder. NetSuite provides a template for this purpose and you should use this template as the basis for your file. For details, see:

- [Using the Template to Create a Bundle Documentation File](#)
- [Selecting a Bundle Documentation File from the File Cabinet](#)
- [Selecting a Bundle Documentation File from Your Computer](#)

You also can enter a URL in the **Admin Documentation** field, to link to an externally hosted site that provides documentation. For more information, see [Using a Website to Provide Bundle Documentation](#).

Using the Template to Create a Bundle Documentation File

To facilitate the creation of your bundle document, NetSuite provides a Microsoft Word template (.dot file) that you can upload and use to structure your document. The template has been designed specifically

for bundle documents and can be filled in. By using it, you ensure that the key documentation your customers require is presented in an efficient and consistent manner.

The template includes the following elements or sections:

- Cover page
- Copyright page
- Table of Contents
- Overview
- Installing the Bundle
- Using the Bundle
- Updating the Bundle
- Uninstalling the Bundle
- Support

Although you can adapt this template, you should use this structure to ensure correct installation by administrators and effective adoption by users of the SuiteBundler.

To create a document using the bundle document template:

1. Start the Bundle Builder, at Setup > Customization > SuiteBundler > Create Bundle.
2. On the Bundle Properties page, click the inline text Template link, or click the label for the **Admin Documentation** field, and then click **Template** in the field level help.
3. Choose whether to open the template using Microsoft Word or to save the template file.
4. Before you begin entering your text, save the template as a Word .doc type file.
5. In the Word .doc file, replace the text in the grey fields with your own text.



Important: You must include the entire NetSuite Disclaimer provided in the Support section.

6. When you have completed entering the text in your Word .doc file, generate a PDF file.
7. Return to the Bundle Properties page of the Bundle Builder.
8. In the **Admin Documentation** field, click **New** from the dropdown list for the field.
9. In the **File** popup window, select **Computer** in the **Attach From** field, check the **Available for SuiteBundles** box, click the **Browse** button, browse to the file, and click **Save**.

The selected file is listed in the **Documentation** field.

Selecting a Bundle Documentation File from the File Cabinet

To use a previously created document that has already been uploaded to the NetSuite File Cabinet as your bundle documentation file:

1. Go to the Bundler Properties page of the Bundle Builder .
2. In the **Admin Documentation** field, click **List** from the dropdown list for the field.
3. In the **Choose Documentation** popup, select the file.

The selected file is listed in the **Documentation** field.



Note: If you do not see the file in the list and you are sure it was uploaded to the File Cabinet, it is possible it was not marked Available for SuiteBundles. You can edit the File Cabinet record to check this box.

Selecting a Bundle Documentation File from Your Computer

To use a previously created document that has not been uploaded to the NetSuite File Cabinet as your bundle documentation file:

1. Go to the Bundle Properties page of the Bundle Builder .
2. In the **Admin Documentation** field, click **New** from the dropdown list for the field.
3. In the **File** popup, select **Computer** in the **Attach From** field, check **Available for SuiteBundles**, click the **Browse** button, browse to the file, and click **Save**.

The selected file is listed in the **Documentation** field.

Using a Website to Provide Bundle Documentation

To use an externally hosted website to provide bundle documentation:

1. Go to the Bundle Properties page of the Bundle Builder .
2. In the **Admin Documentation** field, click **New** from the dropdown list for the field.
3. In the **File** popup, select **Web** in the **Attach From** field and enter the URL in the **URL** field, and click **Save**.

The entered URL is listed in the **Documentation** field.

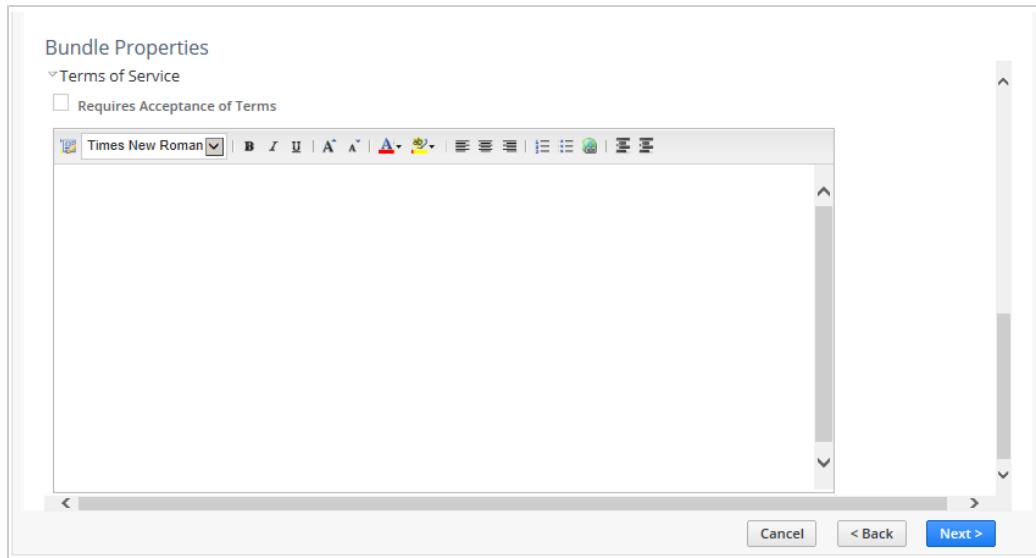
Defining Bundle Terms of Service

You can associate terms of service with a bundle and require target account users to accept these terms before they install or update the bundle.

Note that terms of service defined for an individual bundle do not replace general SuiteBundler terms of service. Bundle-specific terms of service are in addition to general terms of service.

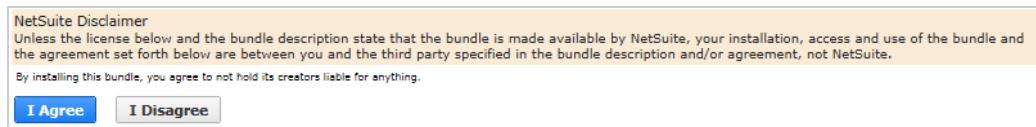
To define bundle-specific terms of service:

1. On the Bundle Properties page of the Bundle Builder, click the arrow next to Terms of Service and enter terms in the box provided.



- Formatting tools are available for terms of service text (when the **Enable Rich Text Editing** option is enabled at Home > Set Preferences).
 - Lengthy text can be edited externally and pasted into the box on this page.
2. If you want to require users to agree terms of service before proceeding with installations and updates, check the **Requires Acceptance of Terms** box.

When this box is checked, users must agree to terms of service during installation and update. Note that a standard NetSuite disclaimer is always included along with your text:



Important: If you do not check the **Requires Acceptance of Terms** box, users will not be required to accept terms and the link to these terms is not displayed.

3. After installation, the terms of service defined for a bundle are available from a link on the Bundle Details page. For details, see [Bundle Terms of Service](#).

Note the following:

- To stop requiring users to agree to terms of service, edit the bundle in the Bundle Builder and clear the **Requires Acceptance of Terms** box.
- Managed bundles only present the terms of service the first time a customer installs the bundle. Users are not required to agree to terms of service for managed bundles during updates, as these updates are pushed by bundle authors. If the terms of service have changed, this change could be communicated to customers through bundle email messages. Customers may respond, indicating whether they accept or reject the new terms of service. If a customer rejects the new terms of service, the author can exclude the customer from future bundle upgrades.
- For pushed bundles, users are not required to agree to terms during installations or updates, as these are pushed by bundle authors with administrative access to target accounts.

Verifying Acceptance of Bundle Terms of Service

You can review the most recent date that each target account has accepted a bundle's terms of service on the Bundle Install Base page.

Bundle Install Base											
<input type="button" value="OK"/> <input type="button" value="Refresh"/>											
<input type="button" value="FILTERS"/>											
1	OneWorld All Access	US	2014.2	57436			✓	Production	22-Jul-2014 3:38 pm	22-Jul-2014 3:38 pm	



Note: The date value is hardcoded to Pacific Time Zone and does not reflect the time zone setting on Home > Set Preferences.

For details on accessing this page, see [Viewing Your Bundle Install Base](#).

Revising Bundle Terms of Service

You can revise a bundle's terms of service at any time, by editing the text in the Bundle Builder.

To force users to accept the updated terms of service when they install or update the bundle, check the **Requires Reacceptance** box. If this box is not checked, the updated terms of service are written to the Bundle Details page, but they are not displayed during bundle installation or update.

Bundle Properties

▼ Terms of Service

Requires Acceptance of Terms

Requires Reacceptance of Terms

By installing this bundle, you agree to not hold its creators, or any of their associated entities, liable for anything.

The **Requires Reacceptance** box is displayed after the **Requires Acceptance** box has been checked at least one time. This box is cleared each time you return to the Bundle Builder to edit terms of service text. The most recent date when you previously required reacceptance of revised terms of service is displayed, if applicable.



Warning: This feature is not intended to provide source control for changes to bundle terms of service, but only to maintain the most current terms of service text. Each bundle author is responsible for maintaining records of changes to the bundle's terms of service.

Protecting Your Bundled Server SuiteScripts

NetSuite provides a mechanism for you to include server SuiteScripts in bundles, and prevent users who install the bundles from seeing the contents of these scripts. This mechanism is the Hide in SuiteBundle preference that you can enable on the script file record in the NetSuite File Cabinet.

For this preference to be available, you must upload your SuiteScript .js file to the File Cabinet and enable the **Hide in SuiteBundle** preference before you create the bundle that includes the server SuiteScript. (Note that this preference is not available for client SuiteScripts.)

The screenshot shows the 'File' edit screen for a file named 'ClientAssistant.js'. The 'Actions' dropdown is open, displaying several checkboxes. The 'Available for SuiteBundles' and 'Hide in SuiteBundle' checkboxes are checked. The 'Hide in SuiteBundle' checkbox is specifically highlighted with a red rectangular border around its label and the checked state.



Important: Setting the Hide In SuiteBundle preference for a script is the only way to prevent target account users from seeing its code. Locking a bundled file, as described in [Locking Objects in Customization Bundles](#), only prevents users from editing the script, not from viewing it or downloading it. For a comparison of setting the Hide in SuiteBundle preference and locking a bundled file, see [Locked vs. Hidden Bundle Scripts Overview](#).

To prevent bundle installers from seeing server SuiteScript contents:

1. Make sure your SuiteScript .js file is uploaded to the File Cabinet.
2. Before you create your bundle, go to Documents > Files > File Cabinet.
3. Click the link for the File Cabinet folder where your script is located, to display its contents. Scripts generally are in the SuiteScripts folder, but may be located in another folder.
4. Click **Edit** next to the script you want to protect.
5. Check the **Hide In SuiteBundle** box.
6. Click **Save**.



Warning: You should not check the **Hide In SuiteBundle** box for a script to be included in a bundle developed in sandbox. Checking this box for a script in the source sandbox account for the bundle can eventually lead to the script being hidden in that account. After the bundle is installed in production and the sandbox account is later refreshed, the production account setting for this option is copied to the sandbox account. This copy causes the **Hide in SuiteBundle** box to be checked and disabled in the source sandbox account, preventing bundle developers from accessing the script.

Locked vs. Hidden Bundle Scripts Overview

You can do the following to limit user access to objects installed by bundles in target accounts:

- In the Bundle Builder, you can lock objects so that users in target accounts cannot edit them. However, users can still view locked objects, including scripts. See [Locking Objects in Customization Bundles](#).
- In the Bundle Builder, you can enable the Hide Components option so that by default, objects installed by the bundle are not displayed on list pages in target accounts. See [Hiding Bundle Components in Target Accounts](#). This setting is a default, however, that can be overridden by any target account administrator. When an administrator disables the Hide Components option, users can see all bundled objects in list pages.
- In the File Cabinet, you can enable the Hide in SuiteBundle option for a server SuiteScript included in a bundle. See [Protecting Your Bundled Server SuiteScripts](#). Enabling this option is the only way to ensure that target account users cannot view the code in a script.

You can use any or all of these options. To understand the combined effects of locking and setting the Hide in SuiteBundle option for a script, see the following table.

Locked?	Hide in SuiteBundle option enabled?	Effect in Target Accounts
No	No	User can view and edit both the script and its NetSuite file record, and can download the script file.
No	Yes	User cannot view, edit, or download the script file, but can view and edit the file record.
Yes	No	User can view both the script and the file record, and download the script file, but cannot edit the script or file record.
Yes	Yes	User cannot view, edit, or download the script file, and cannot view or edit the file record.

File and Folder Management in Bundles

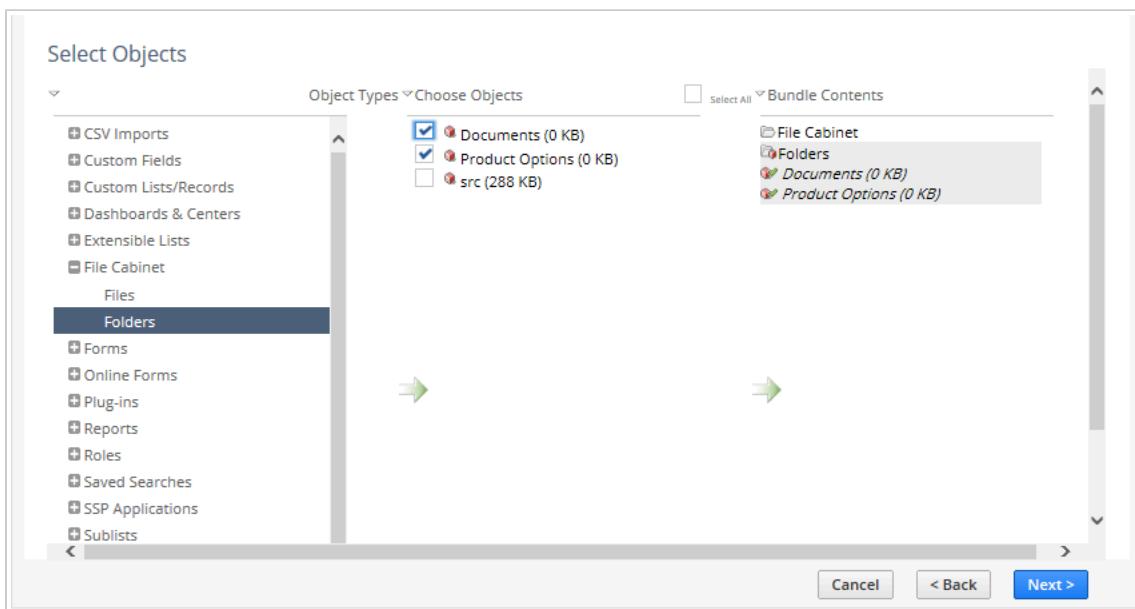
You can use the Bundle Builder to include files and folders from the File Cabinet in a bundle. When a user installs the bundle, NetSuite installs the files and folders included in the bundle into the target account. You can use this feature to include any File Cabinet file in a bundle.

To include a file or folder in a bundle, you must first check the **Available in SuiteBundles** box on the file or folder record. The file or folder record then appears in the Bundle Builder.

The location where NetSuite installs files and folders in the target account depends on the type of source account:

Source Account Type	Installation Location
Sandbox	<p>NetSuite installs the files and folders with the same path information as the source account. The installation location is the same for bundles installed from a sandbox account to another sandbox account, or for bundles installed from a sandbox account into a production account.</p> <p>The Preview Install page for SuiteBundler displays the folder and file paths for the files included in the bundle. Use this information to determine the changes that the bundle makes to the files and folders in the target account.</p> <p>Note: This feature does not apply for bundles created in Version 2014 Release 2 or earlier and installed in Version 2015 Release 1 or later, or for bundles installed from a production account into another production account.</p>
Production	<p>If it does not already exist, NetSuite creates a folder called SuiteBundles in the target account and installs all files and folders into a subfolder that contains the bundle ID in the title.</p> <p>For example, if you include a file named <code>foo.js</code> in a bundle with an ID of 100, NetSuite installs the file into the following File Cabinet directory in the target account:</p> <pre>/SuiteBundles/Bundle 100/foo.js</pre> <p>A user can access this file or folder with a descriptive URL that references the file and folder location. See Accessing Installed Files and Folders with Descriptive URLs.</p>

The following screenshot shows an example of using the Bundle Builder to include folders in a bundle:



Rules and Guidelines for Bundling Files and Folders



Important: Use the following rules and guidelines when working with files and folders in SuiteBundler.

- If the file or folder included in a bundle already exists in the target account, NetSuite updates the file or folder and the associated properties from the source account.

- To update a file or folder in a bundle after you create the bundle, edit the bundle and update the file or folder. When a user installs or updates the bundle, the file or folder is installed or updated with the changes. See [Editing a Bundle](#).

Note: When a remote file that was added to the File Cabinet using a URL is modified, the File Content Hash field changes when you download this file or its parent folder from the File Cabinet. The value of the File Content Hash field also changes when NetSuite accesses the modified file through an internal process, such as a SuiteScript method call, or a SuiteFlow workflow step. This change is reflected in the system notes for that file. For more information, see the help topic [File Cabinet Overview](#).

- If you are creating a bundle on a sandbox account for installation in a production account, and you include a file from a subfolder, but do not explicitly include the folder in the bundle, NetSuite automatically includes the complete folder path for the file. The included subfolders do not appear on the Set Preferences page in the Bundle Builder. However, you can use the Preview Bundle Install page to view the included files and folders when you install the bundle.
- If you include a file or folder in a bundle and then clear the **Available for Suitebundles** box, NetSuite still includes the file or folder in the bundle. To remove the object, edit the bundle and remove the file or folder.
- Use the **Hide in SuiteBundles** box on the file or folder record to prevent anyone who installs the bundle from accessing the file or folder.
- NetSuite may automatically include files in a bundle due to dependency on other objects. For example, when you include a script deployment record in a bundle, NetSuite automatically includes the script file. However, if you lock the script deployment record, NetSuite does not automatically lock the script file. You must manually lock the script file for the script deployment.
- Including large folders in a bundle can extend installation time, in some cases causing network connection problems that result in a failed installation. To ensure installation success, you can perform one of the following actions:
 - You can divide folder contents into multiple smaller bundles.
 - Instead of transferring folder contents to the target account through a bundle installation, you can export all folder contents from the File Cabinet of the source account to a ZIP file and then upload that ZIP file to the File Cabinet of the target account, where you have installed the bundle. For more information, see the help topics [Downloading Files from the File Cabinet](#) and [Uploading Files to the File Cabinet](#).

However, even if you exclude some files from bundle installation and import them to the target account through a ZIP file afterward, some objects in the bundle can still reference such previously excluded files and pull them into the target account during bundle installation, thus resulting into duplicate files being installed in the target account.

There is also a difference in where the files are installed in the File Cabinet, depending on whether you choose to install the files through a bundle, or import the files through a ZIP file.

For example, if you include a file named `foo.js` in a bundle with an ID of 100, NetSuite installs the file into the following File Cabinet directory in the target account:

```
/SuiteBundles/Bundle 100/foo.js
```

However, if you decide to exclude files from bundle installation and import them to the target account through a ZIP file instead, the files imported to the File Cabinet in the target account retain the same folder structure as in the source account.

Adding Files and Folders to a Bundle

To add a file or folder to a bundle, check the **Available for SuiteBundles** box for the file or folder record and then add the file or folder to a bundle in the Bundle Builder.

To add a file or folder to a bundle

1. Make the file or folder available for inclusion in a bundle:
 - a. Go to Documents > Files > File Cabinet, and select the folder that contains the file or folder.
 - b. Click **Edit** for the file or folder, and check the **Available for SuiteBundles** box.

The screenshot shows the 'Document Folder' edit screen. It includes fields for 'Folder Name' (set to 'Documents'), 'Sub-Folder of', 'Type' (set to 'Documents and Files'), 'Derived From', 'Description', and several dropdowns for 'Restrict by Class', 'Restrict by Subsidiary', and 'Restrict by Group'. At the bottom right, there are checkboxes for 'Private' and 'Inactive', and a checked checkbox for 'Available for SuiteBundles', which is highlighted with a red border.

Folders and files with the **Available for SuiteBundles** option enabled are available on the Select Objects page of the Bundle Builder, when **Files** or **Folders** is selected under the File Cabinet object type.

2. Go to .
3. On the [Step 3 Select Objects](#) page of the Bundle Builder, in the **Object Types** list, expand File Cabinet.
4. To add files, select **Files** in the **Object Types** list and in the **Choose Objects** list, check boxes as desired.
5. To add folders, select **Folders** in the **Object Types** list and in the **Choose Objects** list, check boxes as desired.
6. Finish selecting objects to be included in the bundle.

Installing a Bundle That Contains Files and Folders

You can review the files and folders included in a bundle before you install it. If you are installing a bundle from another production account, NetSuite installs the files and folders in a folder called SuiteBundles. If you are installing a bundle from a sandbox account associated with the target account, NetSuite installs the files and folders with the same path information as the source account.

The following screenshot shows the properties for the files and folders that are being installed from a sandbox source account:

The screenshot shows the 'Preview Bundle Install' screen. It has a header with 'Preview Bundle Install' and 'Cancel' / 'Install Bundle' buttons. Below is a table titled 'Name: FilesAndFolders Version:' with columns: BUNDLE CONTENTS, SCRIPT ID, ACTION, PREFERENCE, INTERNAL ID, and LOCK ON INSTALL. The table lists two 'File Cabinet' entries: 'Files' containing '/TempFolder/FolderB/FileB.txt' (Action: Add, Internal ID: 1174) and '/TempFolder/FolderA/FileA.txt' (Action: Update, Internal ID: 2064); and 'Folders' containing '/TempFolder/FolderC (0 KB)' (Action: Add, Internal ID: 180) and '/TempFolder/FolderA (1 KB)' (Action: Update, Internal ID: 282).

Preview Bundle Install						
		Cancel	Install Bundle			
Name: FilesAndFolders Version:						
BUNDLE CONTENTS	SCRIPT ID	ACTION	PREFERENCE	INTERNAL ID	LOCK ON INSTALL	
File Cabinet						
Files						
/TempFolder/FolderB/FileB.txt		Add		1174		
/TempFolder/FolderA/FileA.txt		Update		2064		
Folders						
/TempFolder/FolderC (0 KB)		Add		180		
/TempFolder/FolderA (1 KB)		Update		282		

For more information, see [Starting the Bundle Installation Process](#).

Accessing Installed Files and Folders with Descriptive URLs

The inclusion of folder hierarchies in bundles, and resulting support of descriptive URLs, makes it possible for SuiteScripts to reference files uniformly across different accounts, instead of relying upon file internal IDs that can change from account to account. To support the use of descriptive, or true path, URLs to access bundled files in target accounts, you need to enable the **Available for SuiteBundles** option on the File Cabinet records for files, and their folders and subfolders.

After the bundle has been installed in a target account, the following format can be used for descriptive URLs to access bundled files:

```
1 | https://system.netsuite.com/c.<account ID>/suitebundle<bundle ID>/<folder>/<subfolder>/<file>. <ext>
```

For example, if the target account ID is 160532, bundle ID is 11006, folder names are Documents/ProductOptions, and file name is SearchStrings.txt, the descriptive URL would be:

```
1 | https://system.netsuite.com/c.160532/suitebundle11006/Documents/ProductOptions/SearchStrings.txt
```

Guidelines for Descriptive URLs

- You can only access files and folders with a descriptive URL if the file or folder was installed from another production account, and not from a sandbox account associated with the target account. This restriction only applies to bundles created in NetSuite Version 2015 Release 1 and later. See [File and Folder Management in Bundles](#)
- Files are available only to users logged in to NetSuite, except for files that have the **Available without Login** option enabled on the File Cabinet record.
- To support the use of descriptive URLs to access folder contents in the source account, the Host HTML Files feature must be enabled. This feature provides access to the Web Site Hosting Files/Live Hosting Files File Cabinet folder, where these folders are stored in the source account. This feature is not required for target accounts, where these folders are stored under the SuiteBundles folder.
- If a bundle includes scripts that rely on descriptive URL access, to construct the appropriate URL in the target account, the bundle author must do the following:
 - Parameterize the target account ID using the runtime.accountId API call.
 - Parameterize the bundle ID using the Script.bundleIds API call.



Note: The Script.bundleIds API call returns an array, and the first returned bundle ID should be used to construct the URL.

Bundling NetSuite Financial Statements

Financial statements, which include income statements, balance sheets, and cash flow statements, are specialized reports that use layouts to determine their look and feel. A financial statement layout is a set of definitions for report rows, including the rows and financial sections to be displayed, row order and hierarchy, section filter criteria, methods for calculating summary amounts, row labels, and text, line, and background formatting options.

When you include a custom financial statement in a bundle, the custom layout assigned to the report is automatically bundled as well. The layout, as well as the report, is added to each target account when the bundle is installed.

You can assign layouts to financial statements, and view layout assignments, at Reports > Financial > Row Layout Assignment. Be sure to review these before bundling financial statements. Whenever possible, the assignment of a layout to a bundled financial statement from the source account is preserved in target accounts. If there is ambiguity about this assignment, it is not preserved in target accounts, and users in these accounts can set up the assignments themselves.

For example, in OneWorld accounts, NetSuite provides specialized country-specific layouts for each subsidiary's financial statements, so that each custom layout is associated with a subsidiary as well as with a financial statement. If the source account for a bundled financial statement uses OneWorld, the layout assignment in the source may not be preserved in the target, because the associated subsidiary in the source may not exist in the target.

In cases where the assignment is not preserved, the layout is still added to the target account. After bundle installation, a target account user should review the layout assigned to the bundled financial statement and change this assignment as necessary, at Reports > Financial > Row Layout Assignment. For details, see the help topic [Financial Reports: Row Layout Assignment Page](#).



Important: An update to a bundle that includes a custom financial statement updates the report and the layout objects, but does not update the layout assignment in target accounts. If the assignment has been changed in a target account since the bundle was originally installed, the update does not revert the change.



Note: SuiteBundler cannot properly match up data for financial statements installed from two different bundles sourced from a sandbox account. To avoid duplicate sections and rows in financial statements, use only one bundle to install them from a sandbox account to a production account.

Bundle Object Preferences

For customization bundles, preferences are available for some object types. These preferences, which are applied during bundle installations and updates in target accounts, help to properly set up bundle objects. You set these preferences on the [Step 4 Set Preferences](#) page of the Bundle Builder when you create or edit a bundle, by choosing from options displayed in Preference column dropdown lists.

You can control preferences for pushed and managed bundle accounts, and you can supply default preferences for other install base accounts. The bundle preference options you choose are applied directly to target accounts during managed bundle and other pushed bundle installations and updates. For other bundles, target account administrators can override these preferences before bundle installation or update.

The following types of bundle objects have preferences available in the Bundle Builder.

■ Dashboards

For any dashboard, you can set a preference to indicate whether updates to the bundled dashboard should be applied to the dashboards of target account users to whom the dashboard has been published. (Updates are always applied to the target account dashboard definition.)

This setting is applied during bundle updates, when the dashboard is updated in the target account.

The default option is not to apply updates to users, allowing target account administrators to publish updates to users' dashboards as desired after bundle updates. See the help topic [Applying Changes to Published Dashboards](#).

■ Custom Fields

For any custom field, you can set a preference to choose whether the field should be shown or hidden on existing custom forms in target accounts. The default option is to hide the field.

This setting is applied during bundle installations and updates, whenever the field is first added to a target account. This setting does not affect bundled forms, because all custom fields in a bundle are always shown on all custom forms included in that bundle.

■ Custom Lists

For any custom list, you can set a preference, applied during bundle updates, to choose whether: the bundled custom list data should replace data in the target account custom list, the target account custom list data should be preserved, or data from the bundled custom list and target account custom list should be merged. The default option is to preserve data.

This preference is also available, during both installations and updates, for conflicting custom lists that have the **Replace Existing Object** option chosen. See [Resolving Conflicting Objects](#).

■ Custom Records

For any custom record that includes data, you can set a preference, applied during bundle updates, to choose whether: the bundled custom record data should replace data in the target account custom record, the target account custom record data should be preserved, or data from the bundled custom record and target account custom record should be merged. The default option is to preserve data.

This preference is also available, during both installations and updates, for conflicting custom records that include data and have the **Replace Existing Object** option chosen. See [Resolving Conflicting Objects](#).

■ Custom Segments

For any custom segment that includes data, you can set a preference, applied during bundle updates, to choose whether: the target account custom segment data should be preserved, or data from the bundled custom segment and target account custom segment should be merged. The default option is to preserve data.

When the **Preserve Data** option is selected, the bundle update does not make any changes to target account custom segment values. When the **Merge Data** option is selected, the bundle update merges custom segment values, without deleting any values from the target account custom segment. The avoidance of target account custom segment value deletion is due to the possibility that a custom segment may have GL impact. In addition, no option is provided to replace data for a custom segment, also to avoid unintentional changes to GL impact.

For more information about bundling custom segments, see [Adding a Custom Segment to a Bundle](#).



Note: If a custom record's data is referenced by another custom record for which the Include Data preference is enabled, data for the referenced custom record is included as well, even if the Include Data preference is not enabled for that referenced custom record.

■ Custom Forms

For any custom form, you can set a preference to choose whether the form should be set as the preferred form for the record type in target accounts. The default option is not to set the form as preferred.



Warning: This setting is applied during pushed installations only. Unlike other preferences, bundle installers cannot override this preference's Bundle Builder setting.

■ SuiteScripts

For any SuiteScript, you can set a preference, applied during bundle updates, to choose whether target account script deployments should be updated with deployments from the bundled scripts. The default option is to not update target account script deployments, leaving them unchanged after the bundle update.

If the update deployments option is applied, bundled script deployments completely overwrite target account script deployments, except for script deployment audience definitions. Bundle updates implement an intelligent merge of audience definitions in source and target accounts when the update deployments option is applied. For details, see [Bundle Update Reference](#).

Note that changes to an icon associated with a dashboard SuiteApp portlet are always copied during a bundle update. This overwrite of the icon in the target account occurs even if the bundle preference is set to prevent the update of script deployments. See [Giving Users Dashboard Access to a SuiteApp](#).

i Note: Bundle updates do not change the values of script deployment parameters that have a user or company preference set, even if the bundle preference for the script is set to Update Deployments. For more information, see the help topic [Script Parameter Preferences and Bundles](#) in the SuiteScript Help.

Bundle Object Preferences Table

The following table provides a quick reference to customization bundle object preferences:

Object	Options (** = Default)	Applied During
Dashboards	Do Not Update Users** Update Users	updates
Custom Fields	Hide on Existing Custom Forms** Show on Existing Custom Forms	installations, updates (if new fields)
		i Note: This setting is applied whenever the field is first added to a target account. This setting does not affect bundled forms, because all custom fields in a bundle are always shown on all custom forms included in that bundle.
Custom Lists Custom Records (Include Data only)	Preserve Data** Replace Data Merge Data	installations, updates (for conflicting objects with Replace Existing Objects chosen and for objects to be updated)
Custom Segments	Preserve Data** Merge Data	updates
Custom Forms	Not Preferred** Preferred Form for Record Type	installations (push only)
SuiteScripts	Do Not Update Deployments** Update Deployments	updates

Preferences in Bundles Created Prior to Version 2011 Release 1

Most bundle object preferences became available to bundle authors as of Version 2011 Release 1. After your account was upgraded to this release, any bundles that you created prior to this release have the default option set for each preference. If you want one of these bundles to use any non-default options, you need to edit the bundle in the Bundle Builder and explicitly set the preference(s). See [Editing a Bundle](#).



Note: The preference for dashboards became available as of Version 2012 Release 1.

Preferences in Copied Bundles

Preferences set for a bundle are not applied in an account where the bundle is copied. Note that data is always replaced and script deployments are always updated during an update of a bundle copy. However the preferences set for a bundle are maintained as the defaults on the [Step 4 Set Preferences](#) page of the copied bundle, and are applied in target accounts where the copied bundle is installed, unless the preferences are edited in the copied bundle. See [Copying a Bundle to Other Accounts](#).

Locking Objects in Customization Bundles

A bundle author can lock some of the objects in a customization bundle on the [Step 4 Set Preferences](#) page of the Bundle Builder. When an administrator installs a bundle containing locked objects, these locked objects cannot be edited. Sometimes editing objects in a bundle can break a solution or make future upgrades impossible without data loss. Locking prevents this and allows customization in a controlled and safe manner.

Object locking provides bundle authors and independent software vendors (ISVs) the ability to control which aspects of their solution can be extended and customized to avoid upgrade issues. In addition, locking can ensure compatibility when bundle updates are rolled out.

Before a NetSuite account administrator installs or updates a bundle, the administrator can view which components in the bundle are locked, on the Preview Bundle Install Page. When an installed bundle includes locked objects, these objects' NetSuite pages do not have buttons that allow changes to the object, such as Edit, Delete, and Change ID. Also, all list and view pages display a lock icon to provide complete visibility of locked objects in installed bundles.

You can lock bundle objects as part of bundle creation or editing. To lock a bundle object, check the Lock on Install box on the [Step 4 Set Preferences](#) page of the Bundle Builder.



Warning: You should not lock bundle objects to be included in a bundle developed in sandbox. Locking objects in the source sandbox account for the bundle can eventually lead to the objects being locked in that account. After the bundle is installed in production and the sandbox account is later refreshed, the production account settings for locked objects are copied to the sandbox account. This copy causes the objects to be locked in the source sandbox account, with their lock options disabled, preventing bundle developers from editing these objects.

The installation of a bundle that contains locked objects requires the same steps as installing a bundle with no locked objects. For a detailed description of the procedure used to install a bundle, see [Installing a Bundle](#). Also, see [Review Locked Bundle Objects](#).

The [Objects Available in Customization Bundles](#) table includes a column indicating which object types are lockable.

Effects of locking vary according to object type. See [Effects of Locking Different Object Types](#).



Note: If you bundle a component that is already locked in another installed bundle, the component stays locked in the new bundle as well.

Precautions about Locking Bundled Objects

Locking an object does not prevent target account users from seeing it, only from editing it. For details about how to hide bundle objects in target accounts, see [Hiding Bundle Components in Target Accounts](#).

For details about how to prevent target account users from seeing code in bundled server SuiteScripts, see [Protecting Your Bundled Server SuiteScripts](#).



Warning: Bundle locking functionality is not a tool for Intellectual Property management and should not be treated as such. The intention of this functionality is to discourage end users from making changes to locked bundle objects. Locking an object does not prevent a user from installing another third party bundle with an object of the same name that is unlocked, and that can be used to replace the locked object. However, the resulting replacement object is locked.

When you copy (rather than install) a bundle that includes locked objects, the objects are not locked in the account where the bundle is copied. However, their locked settings are maintained on the [Step 4 Set Preferences](#) page of the copied bundle, and are enforced in target accounts where the copied bundle is installed. See [Copying a Bundle to Other Accounts](#).

Effects of Locking Different Object Types

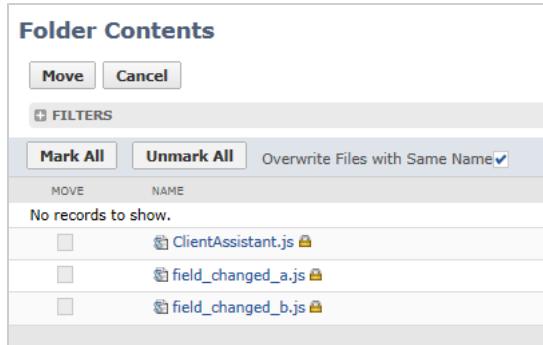
The following list summarizes the effects of locking bundle objects for target accounts:

- **Custom Fields:** Target account administrators can control access to locked custom fields through form customization as before.
- **Custom Lists:** Values cannot be added, edited, or deleted in locked custom lists in target accounts.
- **Custom Records:**
 - Data can be modified in locked custom records in target accounts. In other words, records can be added, edited, and deleted for these custom records.
 - Edits to metadata are restricted for locked custom records in target accounts.
 - For locked custom records that included data when installed in target accounts, no edits to metadata are allowed. Custom record fields cannot be added, edited, or deleted.
 - For locked custom records that did not include data when installed in target accounts, some edits to metadata are allowed. Fields can be added to these locked custom records, and these fields can later be edited and deleted. However, custom record fields from the source account cannot be edited or deleted.
- **File Cabinet**
 - **Files:** In target accounts, the File Cabinet displays lock icons for locked files from installed bundles, as these files cannot be edited.

SuiteBundles > Bundle 57374					
EDIT	NAME &	SIZE	LAST MODIFIED	TYPE	DOWNLOAD
View	ClientAssistant.js	1 KB	07/21/2014 3:09 pm	JavaScript File	Download
View	field_changed_a.js	1 KB	07/21/2014 3:09 pm	JavaScript File	Download
View	field_changed_b.js	1 KB	07/21/2014 3:09 pm	JavaScript File	Download

- Locked files cannot be overwritten in target accounts through Add File, Advanced Add, Copy Files, or Move Files actions in the File Cabinet.

For example, if a user attempts to add a file with the same name as the locked file in the same location in the target account's File Cabinet, the action is not completed and the user receives a warning that a file with the same name already exists.
- The File Cabinet Move Files and Delete Files actions are not available for these locked files. If you choose Move Files or Delete Files for a File Cabinet folder that includes any locked files, the Folder Contents page disables the box for these files and displays a lock icon for each one.



- **Folders:** In target accounts, the File Cabinet displays lock icons for files contained in locked folders from installed bundles. Files in locked bundle folders are treated in the same manner as locked bundle files, as described above.
- **Custom Forms:** Target account administrators who install a bundle containing locked objects can create a new instance of the locked form and customize it to fit their needs. This behavior is similar to how standard NetSuite forms behave.
- **Custom Roles:** Target account administrators who install a bundle containing locked objects, can create a copy of the locked role and further customize it to fit their needs. This behavior is similar to how standard NetSuite roles are handled.
- **Saved Searches:** Users who have access to the saved search can create a new copy of the saved search and add or remove filters and results columns.
- **SuiteScripts:** Locking script records does not lock the script deployments. Customers can change properties of the deployment to fit their business needs. Script parameters lock automatically. However, users can define values for them, if the parameter is defined as company or user preference.

Managed Bundles Overview



Important: This topic is aimed at solution providers who create managed bundles. For information about installing and using managed bundles, see [Using Managed Bundles](#).

If the Managed Bundles feature is enabled in your account, you can create managed bundles to give you control over when bundles are updated in target accounts. When a managed bundle is installed, the installer agrees to subscribe to future updates at the discretion of the bundle's author. As updates are developed, they can be installed by the bundle author into the target accounts without any action required by the target account administrator. Only customization bundles can be managed bundles.



Note: If you are a Premier or Select member of the SuiteCloud Developer Network, you can request access to the Managed Bundles feature by filing a support case through the NetSuite Partner Center. Please be sure to provide your deployment account ID.

NetSuite only enables the Managed Bundles feature on SDN trailing accounts. An SDN trailing account is one that is upgraded in the last phase of a phased release. The Managed Bundles feature is not available for SDN leading accounts.

To indicate that a bundle is a managed bundle, check the Managed Bundle box on the [Step 1 Bundle Basics](#) page of the Bundle Builder. This box is not displayed if the Managed Bundles feature is not enabled in your account. Also, this option is disabled if you enable the Configuration Bundle option, because only customization bundles can be managed.

Unlike pushing bundles, updating a managed bundle from the source account does not require administrator access to the target account. If, however, a bundle author has administrator access to another NetSuite account, they can install a managed bundle in that account. A managed bundle can be saved and upgraded through a deployment account.



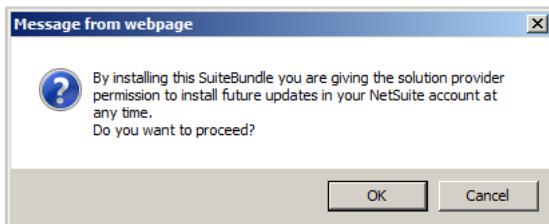
Warning: You should not remove a bundle from the copy chain while the older version of the bundle is installed by some users and that version is deprecated by a newer version of the bundle. Removing a bundle from the copy chain results into users not being able to install the latest version of the bundle. To understand the chaining of bundle copies, see [Ancestry of Copied Bundles](#).

For more information about managed bundles, see:

- [Managed Bundles in Target Accounts](#)
- [SuiteBundler Upgrade Install Base Permission](#)
- [Managed Bundles - Best Practices](#)
- [Upgrading Your Managed Bundle Install Base](#)
- [Phased Upgrade of Managed Bundles](#)
- [Subscribing to Bundle Email Messages](#)
- [Using Managed Bundles](#)
- [NetSuite SuiteApps](#)
- [SuiteApps and Sandbox Accounts](#)
- [Defining Bundle Terms of Service](#)
- [Deprecating a Bundle](#)

Managed Bundles in Target Accounts

When users install a managed bundle in their account, they are presented with a popup message informing them that by installing the bundle, they are giving the bundle author permission to install future updates at any time.



If a bundle is installed and later changed to a managed bundle, the next time the installer clicks **Update**, they are informed of the change and must agree to have the bundle managed by the developer before it can be updated. If the installer no longer wants the managed bundle, they can uninstall it and receive no further upgrades.

In addition, for managed bundles, target account users cannot override bundle object preferences set by bundle authors. If a user-installed bundle later becomes managed, all preferences previously selected in target accounts are ignored during bundle updates. See [Bundle Object Preferences](#).

SuiteBundler sends email notifications of managed bundle upgrades to target account users. Email recipients can be designated on the Bundle Details page. When a managed bundle is successfully

upgraded in a target account, an email message containing the bundle name, bundle ID, bundle version, and account ID is sent. No message is sent when an upgrade fails. The recipients for this email message are based on the **Email Recipients** setting on the Messages subtab of the Bundle Details page in each target account. The default choice is **All Bundle Admins**, indicating that all administrators and users with the SuiteApp Marketplace permission in the account are recipients. A choice of **None** opts out of receiving this email. A choice of **Custom** allows the entry of specific users to receive the email.

The list of installed bundles, at Customization > SuiteBundler > Install Bundle > List, has a Managed column that indicates which bundles are currently managed, for reference purposes.

SuiteBundler Upgrade Install Base Permission

By default, only account administrators can upgrade bundles managed by your company, but you can assign the SuiteBundler Upgrade Install Base permission to custom roles to allow developers and release engineers to perform upgrades.

To grant the SuiteBundler Upgrade Install Base permission:

1. Edit or customize the role to which you want to give this permission.
2. On the **Permissions** subtab, click the **Setup** subtab.
3. In the **Permission** column, select **SuiteBundler Upgrade Install Base**.
The access level is set to **Full**.
4. Click **Add**.
5. Click **Save**.

This permission gives access to the Managed Bundles page.

Managed Bundles - Best Practices

Follow these practices when you use the Managed Bundles feature:

- Develop a communication plan for your updates. You can push bundle updates that are bug fixes quickly and seamlessly without disturbing your install base. For bundle updates that are new releases, be sure to communicate with your install base ahead of time.
- You can communicate with your bundle install base by sending bundle messages, for example, to welcome new users or to provide upgrade notifications. These messages appear on the Bundle Details page and can also be sent as email to bundle administrators in target accounts. See [Sending Bundle Messages](#).
- Avoid doing updates during install base peak hours. You can review the regions in which your install base are located on the Bundle Install Base page to determine the time zones that you need to take into account.
- Be sure that bundle updates do not overlap with the NetSuite phased release cycle if the upgrades rely on new features that are only available in the leading version. You can review the current NetSuite versions of your install base on the Bundle Install Base page. See [Bundle Support during NetSuite Release Phasing](#).
- Managed bundles support a phased upgrade process similar to the release process for NetSuite upgrades. You can release a new managed bundle version to subsets of the install base, in phases. See [Phased Upgrade of Managed Bundles](#).

- Be aware that the order in which bundled scripts run in the source account is maintained when the bundle is installed in target accounts. You can set up script execution order for client and user event scripts on the Scripted Records page, at Customization> Scripting > Scripted Records. See the help topic [The Scripted Records Page](#).
- Lock objects in managed bundles to avoid clobbering customizations in target accounts. You can mark bundle objects as locked on the [Step 4 Set Preferences](#) page of the Bundle Builder. See [Locking Objects in Customization Bundles](#).
- Set bundle object preferences on the [Step 4 Set Preferences](#) page of the Bundle Builder, to ensure that objects are set up properly in target accounts. For information about available preferences, see [Bundle Object Preferences](#).
- You can write SuiteScripts to accomplish required setup tasks that are not covered by bundle preferences, and execute these scripts as part of bundle updates in target accounts. You can associate a SuiteScript with a bundle on the [Step 1 Bundle Basics](#) page of the Bundle Builder. See [Using Bundle Installation Scripts](#).
- If you would prefer that bundle objects not be displayed in target account list pages, you can enable the Hide Components option. See [Hiding Bundle Components in Target Accounts](#).

Upgrading Your Managed Bundle Install Base

One of the issues that solution providers face is ensuring that users of their bundles apply updates as they become available. Managed bundles allow you to upgrade bundles in the accounts where they are installed whenever a new version is available.

Users who have installed a managed bundle do not have responsibility for updating it; the bundle author completes this task at their discretion. For managed bundles, only an administrator or a user with the SuiteBundler Upgrade Install Base permission can apply updates to the install base.

For information about managed bundle upgrades, see:

- [Timing Managed Bundle Upgrades](#)
- [Canceling Pending Managed Bundle Upgrades](#)
- [Notifications for Managed Bundle Upgrades](#)
- [Steps for Managed Bundle Upgrades](#)



Note: Support for managed bundle upgrades varies across the different types of NetSuite accounts. See [Bundle Support Across Account Types](#).

Timing Managed Bundle Upgrades

With managed bundles, you can more carefully manage updates during the phased NetSuite release cycle when some of the accounts in your install base might be running a different version of NetSuite than your account. Managed bundles support a phased upgrade process that supports maintenance of multiple versions of a managed bundle. With phased upgrades, you can continue to push maintenance updates to accounts still using the older version of the bundle, during the time that other accounts are being upgraded to the new version. See [Phased Upgrade of Managed Bundles](#).

You can use the Release Preview account to test updates to a managed bundle before you push updates to the production accounts for your install base. For more information about the Release Preview account, see the help topics [Overview of Release Preview](#) and [Test Installed SuiteApps \(Bundles\)](#).



Note: You should not attempt to maintain more than two released versions of a managed bundle at any one time.

Canceling Pending Managed Bundle Upgrades

Typically a managed bundle upgrade is pushed to many target accounts at one time. If you discover an error before the upgrade is completed in all target accounts, you can stop the upgrade immediately. You do not have to wait for the upgrades to complete in all accounts and then start over. The **Cancel Pending Upgrades** button is available on the Initiate Upgrade page for managed bundles. You can click this button to cancel the upgrades in any target accounts that have a status of **Pending**.

Notifications for Managed Bundle Upgrades

SuiteBundler sends email notifications of successful managed bundle upgrades to target account users. For details, see [Managed Bundle Upgrade Notifications to Target Account Users](#).

SuiteBundler also sends email notifications of managed bundle upgrades to bundle owners. After a managed bundle upgrade has ended, a summary email message containing details about the upgrade is sent to the user who initiated the upgrade. This message can also be sent to other users in the managed bundle's source account. These additional recipients can be set in a new **Email Recipients** field that is available on the Bundle Details page in the source account.

This summary email has a subject of **Summary of Managed Upgrade for Bundle <BundleID>**. It includes the following details:

- bundle ID
- bundle name
- bundle version
- source account ID
- overall upgrade start date and time
- overall upgrade end date and time
- overall status
 - Completed – Upgrade was initiated on all target accounts.
 - Failed – Process failed unexpectedly before upgrade was initiated on all accounts.
 - Canceled – Upgrade initiator clicked Cancel before upgrade was initiated on all accounts.
- details for each target account
 - account ID
 - company name
 - status
 - upgrade start date and time
 - upgrade end date and time

Steps for Managed Bundle Upgrades

Complete the following procedure to upgrade a managed bundle.

To upgrade a managed bundle:

1. Go to Customization > SuiteBundler > Managed Bundles.
2. Click **Upgrade** next to the bundle you want to install in other accounts.
3. The Initiate Upgrade page shows the following:
 - the account number and name of each company that has installed your managed bundle
 - the version of NetSuite they are running
 - the version of the bundle installed in the account
 - the date of the installation



Note: The date value is hardcoded to Pacific Time Zone and does not reflect the time zone setting on Home > Set Preferences.

- the date of the last upgrade



Note: The date value is hardcoded to Pacific Time Zone and does not reflect the time zone setting on Home > Set Preferences.

You can filter the target accounts displayed on this page by region, bundle version, and NetSuite version. See [Filtering on Bundle Install Base Lists](#) and [Filtering Managed Bundle Install Base by NetSuite Version](#).

4. If you have deprecated the managed bundle to be upgraded, be aware of the following:
 - The Initiate Upgrade page for the deprecated version of a managed bundle lists only the accounts that have not yet been upgraded to the replacement version. This page can be used to push maintenance fixes to accounts with the lagging bundle version.
 - The Initiate Upgrade page for the replacement version of the bundle lists accounts with the deprecated version as well as accounts with the replacement version installed. This page can be used to upgrade accounts with the lagging bundle version to the newer version as well as push maintenance fixes to accounts that already have the newer version installed.
5. Check the box in the **Include** column next to each account where you want to install the new version, or use **Mark All** and **Unmark All** to select accounts to upgrade.
6. Click Upgrade.
 - After you have initiated the upgrade, the Bundle Install Base shows the status of the upgrade in each account in the **Install Status** column
 - At any time, you can click the **Cancel Pending Upgrades** button to prevent the upgrade of accounts with a status of **Pending**. When you click this button, a popup asks whether you are sure. When you confirm, a banner appears on the Initiate Upgrade page stating that cancellation of pending upgrades is in progress, until the cancellation is complete.



Important: During an upgrade to a managed bundle, handling of data updates for custom lists, and for custom records that include data, depends on the option set for each of these objects on the [Step 4 Set Preferences](#) page of the Bundle Builder. The default is to preserve target account object data, but the bundle author can also choose an option to replace target account object data with bundled object data. See [Bundle Object Preferences](#).



Note: You can update a managed bundle in accounts that have installed it, but you cannot install a bundle in other accounts unless you have administrator access. For more information, see [Pushing a Bundle to Other Accounts](#).

The user shown in the bundle audit trail as executing a managed bundle update in a target account is an arbitrary designation. The managed bundle update process selects a user in the target account employee

list to be the entry in the **Installed By** column of the audit trail. This selection is the administrator that has the lowest internal ID.

After an upgrade of a managed bundle, the owner of the bundle is set to a random account administrator. This setting may not correspond to the administrator who originally installed the bundle. This behavior is intentional and should not cause issues.

Phased Upgrade of Managed Bundles

SuiteBundler supports a phased release process for managed bundle upgrades, similar to the release process for NetSuite upgrades. You can release a new managed bundle version to subsets of the install base, in phases, instead of having to upgrade the entire install base at the same time. You can maintain multiple versions of a managed bundle, because you can continue to push maintenance updates to accounts still using the older version of the bundle, during the time that other accounts are being upgraded to the new version.



Note: You should not attempt to maintain more than two released versions of a managed bundle at any one time.

You can copy both a managed or a non-managed bundle to begin work on a new version, then deprecate the original bundle. However, the consequences of deprecating a managed bundle are different from deprecating a non-managed bundle.

- When you deprecate a non-managed bundle, you can no longer edit it or install it in target accounts. Also, any accounts that upgrade a non-managed deprecated bundle automatically received the new version.
- When you deprecate a managed bundle, you can continue to edit it to fix issues, and you can continue to install it and push upgrades to it in target accounts.



Warning: You should not remove a bundle from the copy chain while the older version of the bundle is installed by some users and that version is deprecated by a newer version of the bundle. Removing a bundle from the copy chain results into users not being able to install the latest version of the bundle. To understand the chaining of bundle copies, see [Ancestry of Copied Bundles](#).

The following screenshots of the Bundle Availability page illustrate the differences in deprecation behavior for managed and non-managed bundles.

- For a non-managed bundle:

Deprecate Bundle

Choose the copy to replace bundle 100. Previous installers will automatically be upgraded to this replacement bundle the next time they update.

Deprecate With Bundle

- For a managed bundle:

Deprecate Bundle

Choose the copy to replace bundle 100. You can select which previous installations to upgrade and you can maintain the deprecated bundle for others as needed.

Deprecate With Bundle

XXX Account: XXXXXXXX

Deprecated managed bundles have the Edit, Push, and Upgrade Install Base options available in the Action menu on the Saved Bundles page and Managed Bundles page. Non-managed bundles do not have these options available on the Saved Bundles page.

Managed Bundles					
New					
Action	Name	Bundle ID	Copied From	Deprecated By	Version
	MBA Courses - Managed	XXXXXX	XXXXXX Account:XXXXXXXX		0.7
 Test		XXXXXX			1.1
 Set Availability					
 Push					
 Upgrade Install Base					
 Show Install Base					
 Copy					
 Show Copies					

For steps for upgrading a managed bundle, see [Upgrading Your Managed Bundle Install Base](#).

Filtering Managed Bundle Install Base by NetSuite Version

The ability to filter managed upgrades by NetSuite version is supported. If you own a managed bundle, you can use this filter to see the NetSuite version being run in your managed bundle install base accounts. You can choose to implement managed bundle upgrades only for target accounts running either the lagging or leading version of NetSuite. Also, you can choose to send bundle notification messages only to target accounts running a specified NetSuite version.

A **NetSuite Version** dropdown list is available in the Filters section on the Initiate Upgrade page for managed bundles. To get to this page, go to Customization > SuiteBundler > Create Bundle > List, and select Upgrade Install Base for a managed bundle.

The **NetSuite Version** dropdown list is also available on the Recipients subtab on each bundle Message page for a managed bundle. To get to this page, go to Customization > SuiteBundler > Create Bundle > List, and click the bundle name to go to the Bundle Details page. On the Bundle Details page, click the Messages subtab and the New Message button.

Saved Bundles

See the following for details about tasks you can complete with saved bundles:

- Reviewing the Saved Bundles List
 - Bundle Availability
 - Sharing a Bundle
 - Deprecating a Bundle
 - Publishing SuiteApp Help Content to the NetSuite Help Center
 - Copying a Bundle to Other Accounts

- Pushing a Bundle to Other Accounts
- Viewing Your Bundle Install Base
- Editing a Bundle
- Bundle Details
 - Bundle Terms of Service
- Sending Bundle Messages

Reviewing the Saved Bundles List

After you have created a bundle, it is listed on the Saved Bundles page, available at Customization > SuiteBundler > Create Bundle > List.

Saved Bundles										
	New									
ACTION	NAME	BUNDLE ID	VERSION	APP ID	ABSTRACT	ADMIN DOCUMENTATION	HELP CONTENT	CREATED BY	CREATED ON	LAST UPDATE
	Enhanced Order Form	419	1.1			Admin Documentation	Update Help Content		07/06/2017 11:20 am	
	second test	385					Create Help Content		06/30/2017 9:38 am	
	test	351					Update Help Content		06/30/2017 6:18 am	

This page includes the following information for each bundle created in the current account:

- Bundle name
- Bundle ID
- Bundle version
- Abstract
- Link to the administrator documentation file for the bundle
- Link to a page where you can set up user help content for the bundle to be published in the NetSuite Help Center
- User name of the bundle author
- Date of bundle creation

Note: The date and time value is hardcoded to Pacific Time Zone and does not reflect the time zone setting on Home > Set Preferences.

- Date of last update.

Note: The date and time value is hardcoded to Pacific Time Zone and does not reflect the time zone setting on Home > Set Preferences.

- (Copied bundles only) bundle from which it was copied
- (Deprecated bundles only) replacement bundle

You can click the bundle name to go to the Bundle Details page, which provides more extensive information about the bundle, and the ability to send and review bundle messages. See [Bundle Details](#).

You can use the dropdown menu in the Action column to link to pages for other bundle tasks, including:

- Editing the bundle in the Bundle Builder (See [Editing a Bundle](#).)
- Editing bundle availability settings (See [Bundle Availability](#).)
- Publishing help content for the bundle in the NetSuite Help Center (See [Publishing SuiteApp Help Content to the NetSuite Help Center](#).)
- Pushing the bundle to other accounts (See [Pushing a Bundle to Other Accounts](#).)
- Reviewing a list of accounts where the bundle is installed (See [Viewing Your Bundle Install Base](#).)
- Creating a copy of the bundle in another account (See [Copying a Bundle to Other Accounts](#).)
- Reviewing a list of accounts where the bundle has been copied (See [Reviewing Copy Bundle Status](#).)
- (Managed bundles only) Upgrading bundles installed in other accounts (See [Upgrading Your Managed Bundle Install Base](#).)

Bundle Availability

You can access the Bundle Availability page from the Saved Bundles page, by clicking **Set Availability** in the Action list.

On the Saved Bundles page, you can do the following:

- Set a bundle's availability level by indicating where it can be installed and whether it is visible in all users' basic search results. See [Sharing a Bundle](#).
- Indicate whether the bundle is a deprecated version that has been replaced by another bundle. See [Deprecating a Bundle](#).

You can use this page to implement phased release of updates for non-managed bundles. See [Implementing Phased Updates by Setting Bundle Availability](#).



Important: The bundle availability level is ignored for users who are account administrators for the source account of a bundle. Any user who is an account administrator for the account where a bundle was created can install the bundle to any other accounts where this same user is an account administrator, without limitations.

Sharing a Bundle

By default, new bundles are created as private, meaning the bundle can only be installed in other accounts to which you have administrator access. You can choose to share a bundle with a selected list of accounts or make it publicly available to all NetSuite accounts.

For shared and publicly available bundles, you can also indicate whether the bundle is visible in basic search results for all users, including those who do not have access to install the bundle.



Note: Only administrators and users with roles that have the SuiteApp Marketplace permission can share bundles.

To share a bundle:

1. Go to Customization > SuiteBundler > Create Bundle > List.
2. In the **Action** list for the bundle you want to share, click **Set Availability**.
3. On the Bundle Availability page, choose a value for the **Level** field:

- **Private** – A private bundle is only accessible to other accounts to which you have administrator access. This is useful during development and testing, before you are ready to share the bundle with others.

It is a best practice that bundles in sandbox or other development accounts have a **Level** of **Private**.

The **Visible By All** option is disabled and cannot be edited for a private bundle.

- **Shared** – Administrators of accounts that you list in the **Shared Account IDs** field can view and install a shared bundle from a deployment account if the bundle is copied there.

If the **Visible By All** box is checked, all users can view the bundle in basic search results. This visibility advertises the bundle to users who do not have install access and enables users with install access to find the bundle, as they do not need to know an account ID.

- **Public** – Any NetSuite administrator can view and install a public bundle from a deployment account if the bundle is copied there.

If the **Visible By All** box is checked, all users can view the bundle in basic search results, so they do not need to know an account ID to install the bundle.



Important: You cannot make a configuration bundle public.

4. If you have set the **Level** to **Shared**, enter the account IDs of the NetSuite accounts with which you want to share your bundle.

The account ID is the number that is shown when you click **SuiteAnswers** on the **Support** tab and then click **Contact Support by Phone** on the SuiteAnswers site. This is usually a 6 or 7-digit number. You can separate account IDs with commas.

5. If you have set the **Level** to **Shared** or **Public**, you can clear the **Visible By All** box if you do not want the bundle to be returned in basic search results.
 - If this option is enabled, the bundle is visible in all users' basic search results, which for shared bundles, includes users who do not have access to install the bundle.
 - If this option is disabled, the bundle is not returned in basic search results. Users with access to install the bundle can find it through an advanced search.
6. Click **Save**.



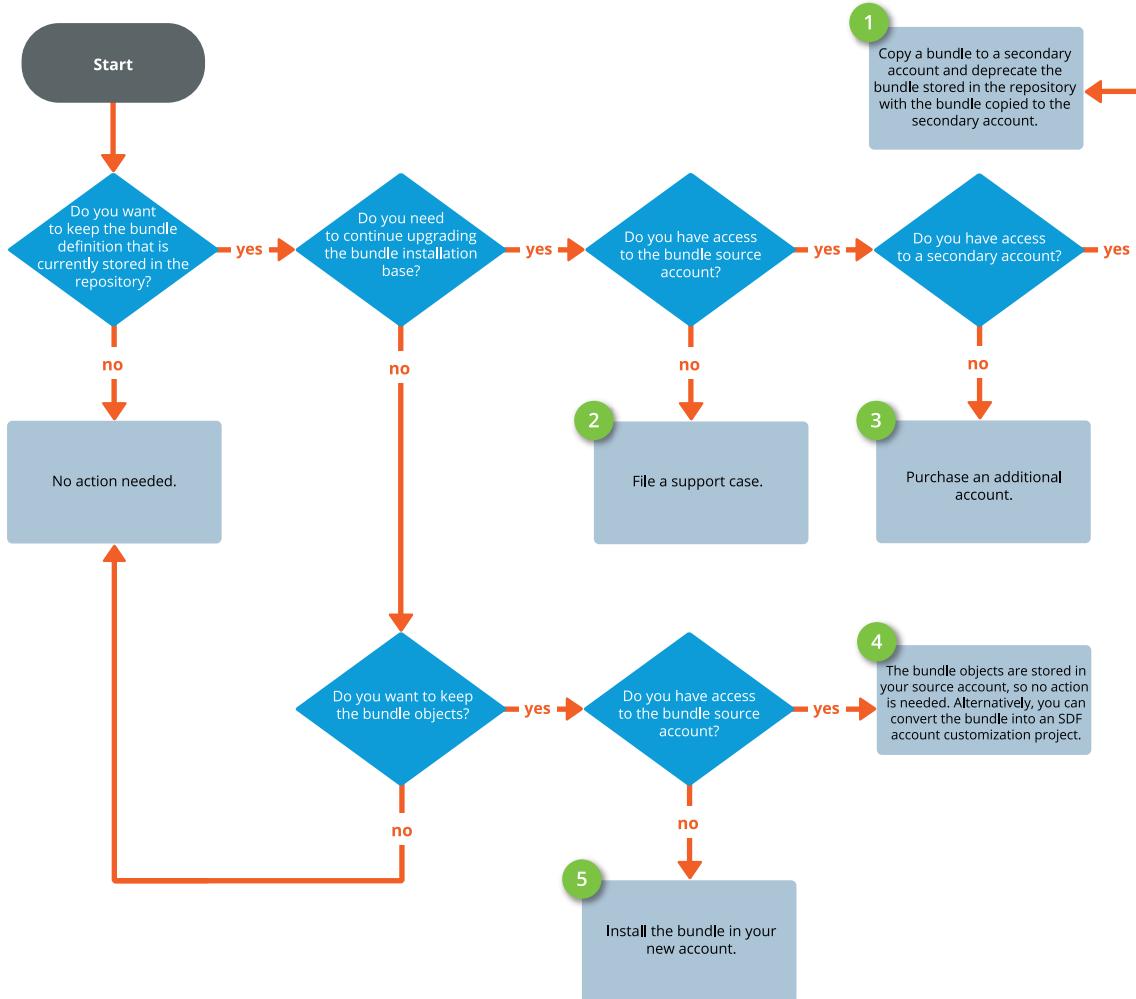
Note: A bundle that is shared with a production account is automatically shared with the first sandbox account associated with that production account because the two accounts share an ID. However, you must explicitly share the bundle with any additional sandbox accounts associated with that production account because they have different account IDs.

Moving a Bundle Out of the Bundle Repository



Important: The NetSuite bundle repository was deprecated as of January 18, 2021.

If you have not already done so, take the required actions shown in the following diagram to remove your existing bundles from the bundle repository. There are five possible scenarios for bundles in the repository that may require you to take action. These different scenarios are indicated by the numbers in the diagram.



Based on your chosen use case in the preceding diagram, see the following options for moving your bundles out of the repository:

- **Scenario 1:** You need to continue upgrades to the bundle install base, and you have access to the bundle source account and to a secondary account.
 1. To copy a bundle to a secondary account and deprecate the bundle stored in the repository with the bundle copied to the secondary account, see [Remove Install Base of Custom Bundle Out of NetSuite Repository](#).
 2. After deprecating the bundle stored in the repository with the bundle copied to the secondary account, the bundle from the secondary account will be the one installed into target accounts, instead of the bundle from the repository.
 3. After this deprecation, the bundle from the secondary account will be the one installed into target accounts, instead of the bundle from the repository.
- **Scenario 2:** You need to continue upgrades to the bundle install base and you do not have access to the bundle source account.
 1. File a case with NetSuite Support to gain access to the source account for the bundle.
 2. Start the case title with the following prefix: Bundle repository deprecation support
 3. Include the bundle ID and account ID of the bundle source account that you cannot access.

4. After you gain access to this source account, check whether you have access to a secondary account. If yes, see Scenario 1. If no, see Scenario 3.
- **Scenario 3:** You need to continue upgrades to the bundle install base, you have access to the bundle source account, but not to a secondary account.
 1. To purchase an additional account, please contact your Account Manager or Sales Rep.
 2. If you do not know your Account Manager, please contact NetSuite Customer Support and ask for this information.
 3. After you have purchased an additional account, see Scenario 1.
 - **Scenario 4:** You do not need to continue upgrades to the bundle install base, but you want to keep the bundle objects and you have access to the bundle source account.
 1. You have the option of making the bundle available for future tasks in SDF instead of SuiteBundler. To convert a bundle into an SDF account customization project, see the help topic [Converting a Bundle into an Account Customization Project from a NetSuite Account](#).

Note: Converting a bundle into an account customization project saves the bundle definition as a zip file in the file cabinet.
 - **Scenario 5:** You do not need to continue upgrades to the bundle install base, but you want to keep the bundle objects and you do not have access to the bundle source account.
 1. You can install the bundle in a different account so that you will have access to the bundle objects in the future. To install a bundle in your new account, see the help topic [Installing a Bundle](#).

Note: If a bundle installed in a new account contains locked objects, you can file a support issue to get these locked objects unlocked.

 2. After the bundle is in a new account that you can access, you have the option to convert the bundle into an SDF account customization project. See Scenario 4.

Deprecating a Bundle

You can deprecate a bundle to indicate that it is no longer the most up to date version available. When you deprecate a bundle, you specify another bundle as a replacement, newer version. You specify this replacement bundle by its name and NetSuite account ID. For instructions, see [Steps to Deprecate a Bundle](#).

You can deprecate a bundle only after you have copied this bundle from one account to another. The copy of the bundle serves as the basis for development of a new bundle version. For details, see [Copying a Bundle to Other Accounts](#).

A bundle may have multiple copies, and each copy may in turn have multiple copies of its own. When you deprecate a bundle, the replacement bundle must be a direct descendant of the deprecated bundle. To understand the chaining of bundle copies, see [Ancestry of Copied Bundles](#).

The effects of depreciation are different for managed bundles and other bundles. For details, see [Results of Deprecating Non-Managed Bundles](#) and [Results of Deprecating Managed Bundles](#).

On the Saved Bundles page, a deprecated bundle includes a Deprecated By column with the bundle ID and account ID of the replacement bundle, in the format <Bundle ID> Account:<Account ID>. For other details about deprecated bundles, see [Notes about Deprecated Bundles](#).



Important: Deprecation is available only for customization bundles. You cannot copy or deprecate configuration bundles. Also, you cannot deprecate a bundle that has been created in a sandbox account.

Steps to Deprecate a Bundle

Complete the following procedure to deprecate a bundle.

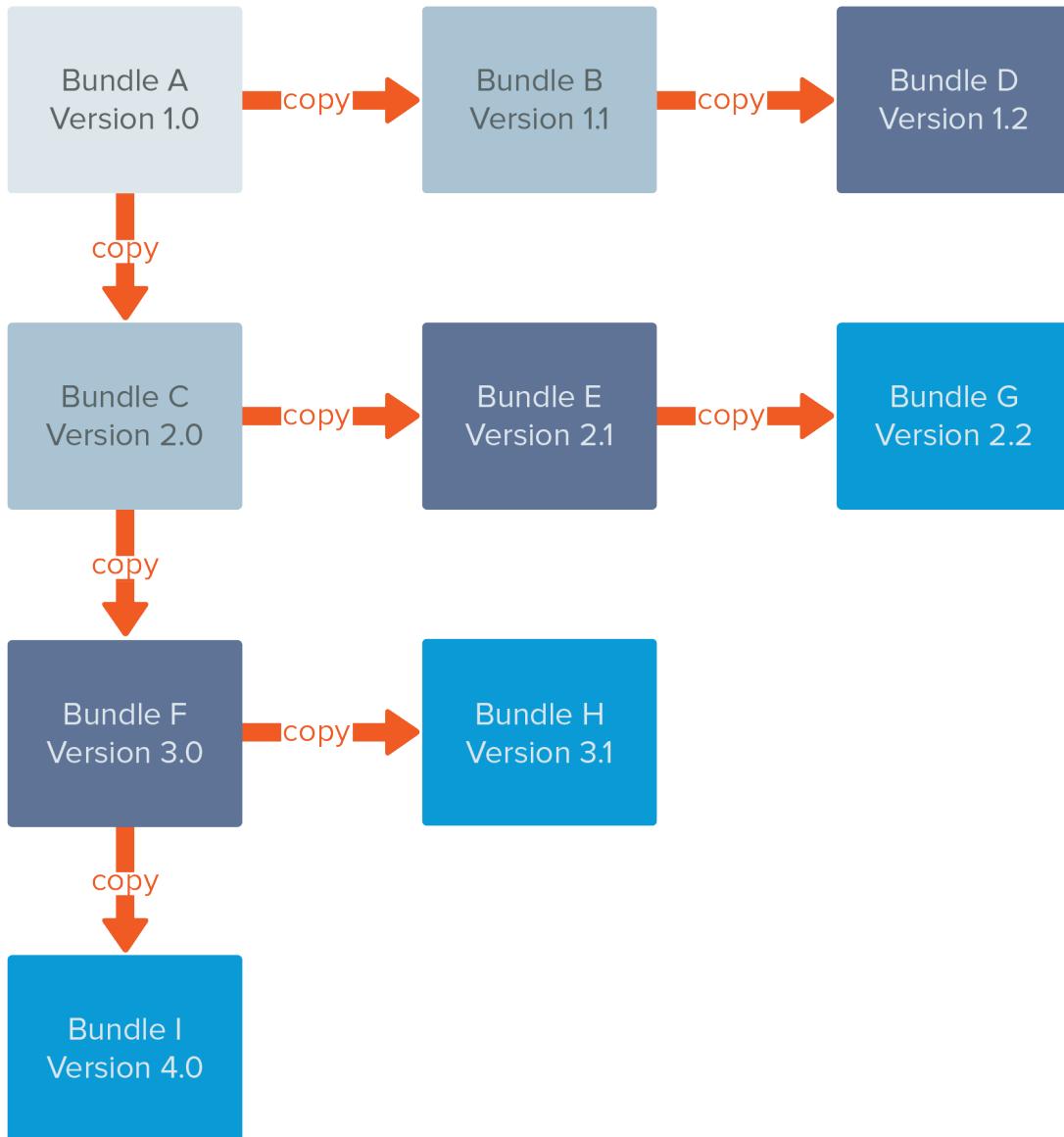
To deprecate a bundle:

1. Go to Customization > SuiteBundler > Create Bundle > List.
2. In the **Action** list for the bundle you want to deprecate, click **Set Availability**.
3. On the Bundle Availability page, in the Deprecate Bundle section, select a replacement bundle for the bundle to be deprecated.
 - **Deprecate With Bundle** lists bundles that are descendants of the deprecated bundle, in accounts where you have a role with SuiteApp Marketplace permission.
 - The format of list options is **<Bundle ID> Account:<Account ID>**.
4. Review other settings on the page. You may want to change the level of availability for the deprecated bundle or **Visible By All** setting.
5. Save the changes to the Bundle Availability page.

Ancestry of Copied Bundles

Each copied bundle can have multiple ancestors, depending on the versioning strategy used for the bundled solution. An ancestor can be the bundle from which a bundle was directly copied, or a bundle from which a chained set of copies was made. The following illustration shows a chained set of copies. In this illustration, Bundle A is the common ancestor of all of the other bundles.

Note that bundles must be deprecated by their descendants. For example, in the scenario illustrated here, you could deprecate Bundle B and replace it with Bundle D, but you could not replace it with Bundle C.



- The deprecation feature supports chained updates across multiple new bundle versions. For example, if you deprecate Bundle A with Bundle B, and later deprecate Bundle B with Bundle D, accounts that still have Bundle A installed will be upgraded to Bundle D when they choose to update.
- A bundle must be deprecated by a direct descendant. For example, in the scenario where: A is copied to B, A is copied to C, and B is copied to D:
 - A can be deprecated by B, C, or D.
 - B can be deprecated by D.
 - D cannot be deprecated because it does not have any descendants.

Results of Deprecating Non-Managed Bundles

- After a non-managed bundle has been deprecated, it can no longer be edited in the Bundle Builder.
- The following occurs when users attempt to update or install a non-managed bundle that has been deprecated:

- When users attempt to install a bundle that has been deprecated, they are notified that the bundle can no longer be installed, and directed to install the replacement bundle instead.
- When users attempt to update a bundle that has been deprecated, the update uses the replacement bundle. Because a replacement bundle may have different availability settings than its deprecated bundle, an access check for the replacement bundle is performed. If access has not been provided for the user's account, the following message is returned: "You cannot update this bundle as it has been deprecated, and you have not been granted access to the replacement bundle. Please contact the solution provider."
- The Bundle Install Base page for a deprecated bundle lists only accounts that have not yet updated to the replacement bundle. The Bundle Install Base page for a replacement bundle lists accounts that have updated to the replacement bundle, AND accounts that still have the deprecated bundle installed.
- Non-managed deprecated bundles do not have the Edit and Upgrade Install Base options available in the Action menu on the Saved Bundles page, because these actions are not supported.

Results of Deprecating Managed Bundles

Deprecation of managed bundles is handled differently than deprecation of non-managed bundles. Managed bundle deprecation allows continuing support of an existing bundle version that has been installed in target accounts, including the ability to release bug fixes and minor changes, at the same time that development of a new version is occurring. See [Phased Upgrade of Managed Bundles](#).

- A deprecated version of a managed bundle can be edited in the Bundle Builder.
- You can continue to push upgrades of the deprecated version of a managed bundle to target accounts that have not yet been upgraded to the replacement version. These actions can occur during the same time period that you are installing and upgrading the replacement version of the bundle in other accounts.
- Deprecated managed bundles have the Edit, Push, and Upgrade Install Base options available in the Action menu on the Saved Bundles page and Managed Bundles page.
 - Edits to a deprecated managed bundle should be for bug fixes only. Edits for new functionality should be made to the replacement bundle.
- The Initiate Upgrade page for deprecated managed bundle versions are organized as follows:
 - The Initiate Upgrade page for the deprecated version of a managed bundle lists only the accounts that have not yet been upgraded to the replacement bundle. This page can be used to push maintenance fixes to accounts with the lagging bundle version.
 - The Initiate Upgrade page for the replacement version of the bundle lists accounts with the deprecated version as well as accounts with the replacement version installed. This page can be used to upgrade accounts with the lagging bundle version to the newer version as well as push maintenance fixes to accounts that already have the newer version installed.

Notes about Deprecated Bundles

- Deprecating a bundle affects the display of bundle messages on the Bundle Details page in install base accounts. After an account installs the replacement bundle, only messages sent for that replacement bundle are displayed. Any messages previously sent for the deprecated bundle are not available in these updated accounts. The bundle author can continue to send messages from a deprecated bundle, but these are only sent to accounts that have not yet updated to the replacement bundle. See [Sending Bundle Messages](#).
- If a replacement bundle includes a bundle installation script, any later replacement bundle deprecating the same bundle must also include a bundle installation script with the same logic. For

example, in the case above, if Bundle B's bundle installation script executes data changes before update, Bundle C's bundle installation script must include this same logic in its Before Update function.

Implementing Phased Updates by Setting Bundle Availability

For managed bundles, NetSuite supports phased release of upgrades to subsets of the install base. For details, see [Phased Upgrade of Managed Bundles](#).

NetSuite does not support this same process for bundles that are not managed. However, you can complete the following steps to release an upgrade for a non-managed bundle to subsets of your install base.

To release a bundle update to subsets of customers in phases:

1. Create a new bundle. (See [Creating a Bundle with the Bundle Builder](#).)
 1. Set the bundle version number and include it in the bundle name for maximum clarity, for example, My SuiteApp v1.0.
 2. Set the bundle's availability to **Public** or **Shared**, so that it is available for installation by customers. If you set it to **Shared**, enter account IDs for all customers on the Bundle Availability page. (See [Bundle Availability](#).)
2. Copy the original bundle. (See [Copying a Bundle to Other Accounts](#).)
 1. Up the version number of this copy and include the version number in the bundle name, for example, My SuiteApp v2.0.
 2. Set the bundle availability to **Private** during the time that it is in development.
 3. Make the desired updates to the bundle.
3. At this point, you can deprecate the original bundle with the copied bundle, to prevent new customers from installing the older bundle version. (See [Deprecating a Bundle](#).)
Be aware that after the original bundle is deprecated, it can no longer be edited.
4. Set the availability of the copied bundle to Shared, and enter account IDs on the Bundle Availability page for the subset of customers who should have access to the updated bundle in the first phase.
 - All of the customers who have installed My SuiteApp v1.0 will see an "Update Available" indicator.
 - However, only the customers whose account IDs are listed on the Bundle Availability page for My SuiteApp v2.0 will be able to update the bundle.
 - Other customers will see an error when they attempt to run the bundle update. To reduce confusion, you need to proactively communicate with customers about when they can expect the update to be available for their accounts.
5. Repeat step 4 multiple times, as desired, to enter account IDs for additional customers on the Bundle Availability page, until you have made the updated bundle available to all of your customers.

Publishing SuiteApp Help Content to the NetSuite Help Center

The NetSuite Help Center supports publishing documentation for SuiteApps developed by partners and customers. You can use this feature to share SuiteApp documentation with users. This feature is

different from the SuiteApp documentation that you can make available through a link on the Bundle Details page, as described in [Documenting a Bundle](#). The Bundle Details page is usually accessible only to administrators, so documentation available from this page is not readily available to users. This documentation should be aimed at administrators. The SuiteApp documentation published in the Help Center should be aimed at users.

To publish SuiteApp help content to the NetSuite Help Center:

1. Go to the Publish Help Content page for a bundle. After you have saved the bundle:
 - a. Click the **Publish Help Content** link on the Bundle Builder Finished page, or
 - b. Click the **Create Help Content** link in the Help Content column for the bundle on the Saved Bundles list page.
2. On the Publish Help Content page, enter the URL for your website, to be shown in the Help Center topic about the SuiteApp.

TITLE *	LANGUAGE *	FILE *	SIZE (KB)	LAST MODIFIED	FILE TYPE
netsuitedevelopmentandcu	English (U.S.)	netsuitedevelopmentandcustomization.pdf	4757	07/06/17 00:00:00 AM	pdf

3. Enter a description of the SuiteApp, to be shown in the Help Center topic.
You can enter a description in languages other than English that are supported by the Help Center.
4. On the Files subtab, click the **Add File** button to upload one or more files containing the SuiteApp documentation content.
Currently, only PDF files are supported.
This subtab shows the title and language to be shown in the Help Center for each file. You can edit the title and select a language other than English as appropriate.

After you have saved your entries on the Publish Help Content page, your SuiteApp documentation is published to the Help Center, in accounts where the SuiteApp is installed. This content is available from a page titled Third Party SuiteApps. Users can access this documentation by going directly to this page. The Third Party SuiteApps page includes a table of third party SuiteApps. Each SuiteApp listed in the table has a link to a child help page titled with the name of the SuiteApp. This help page links to the PDF file or files containing the SuiteApp documentation, and to the website entered on the Publish Help Content page.

The screenshot shows a web browser window with the URL [SuiteApps > SuiteApps > Third Party SuiteApps \(not provided by Oracle\)](#). At the top right is a print icon. The main title is "Third Party SuiteApps (not provided by Oracle)". Below it is a paragraph about third-party SuiteApps. A section titled "Installed SuiteApps" contains a table with one row. The table has columns: Name, ID, Version, Website, Description, and Help. The single row shows "TestHelpBundle" with ID 139022, Version 000565555, Website <https://> (redacted), Description "testing netsuite bundle", and Help "lorem (en_US)" and "small - Copy (en_US)".

Name	ID	Version	Website	Description	Help
TestHelpBundle	139022	000565555	https:// (redacted)	testing netsuite bundle	lorem (en_US) small - Copy (en_US)

Users can click the name of the bundle to display a page with full details.

The screenshot shows a detailed view of the "TestHelpBundle (not provided by Oracle)". It includes a warning box about third-party content, the bundle's name, version, website, and help content sections. Under "Help Content", there is an "English" section with "lorem" and "small - Copy" options.

TestHelpBundle (not provided by Oracle)

Version 000565555
testing netsuite bundle

[Website](#)

[Help Content](#)

English

lorem
[small - Copy](#)

Users also can access the documentation through a Help Center search.

You can make changes to published SuiteApp help content as needed. An **Update Help Content** link is available in the Help Content column of the Saved Bundles list page. Click this link to make edits to the Publish Help Content page.

Copying a Bundle to Other Accounts

You can use the Copy option available in the Action list on the Saved Bundles page to copy a customization bundle from one account into another. You can copy one bundle to multiple other accounts. If you make changes to the original bundle, you can repeat the copy action to update copied bundles in other accounts.

- Copied bundles expand the options for bundle deployment. For example, you can copy a bundle to a deployment account to support your release process.
- Copied bundles can help you to implement a bundle versioning strategy. You can copy a bundle to a development account, allowing the separation of changes to the released bundle version from the development of a new version.
- The ability to copy bundles is linked to the ability to deprecate bundles, meaning replacing a bundle with an updated version. For details about deprecation, see [Deprecating a Bundle](#).

For details about copying bundles to other accounts, see the following:

- [Limitations for Copying Bundles](#)
- [Steps to Copy a Bundle](#)
- [Reviewing Copy Bundle Status](#)
- [Notes about Copied Bundles](#)

Limitations for Copying Bundles

Note the following limitations on copying bundles:

- The copy action is available only for customization bundles. You cannot copy or deprecate configuration bundles.
- You cannot create a copy of the bundle in the same account where it is currently located (the source account). Each copy must be made to a different account.
- You cannot create more than one copy in any account. When you copy a bundle to an account where it has been previously copied, the existing copy is updated.
- Bundle copy is only supported for production accounts and development accounts. You cannot copy a bundle from or to any type of account other than production or development.
- You cannot copy a bundle to an account where it has been previously installed, and you cannot install a bundle in an account where it has been previously copied. For bundles that are copies of copies, this limitation extends to any bundles that have a common ancestor, based on tracking of the full ancestral path of each bundle.
- You cannot copy a bundle created from an SDF SuiteApp to an account where the same SDF SuiteApp has been previously installed.

Steps to Copy a Bundle

Complete the following procedure to copy a bundle.

To copy a bundle:

1. Go to Customization > SuiteBundler > Create Bundle > List.
2. In the **Action** list for the bundle you want to deprecate, click **Copy**.

The Copy Bundle to Accounts page displays, with a list of the accounts to which you have administrator access. (Account IDs and names in the following screenshot have been obfuscated for privacy reasons.)

Copy Bundle To Accounts												
<input type="button" value="Cancel"/> <input type="button" value="Mark All"/> <input type="button" value="Unmark All"/> <input type="button" value="Copy"/>												
Name: Industry Specific Bundle ID: 6062 New Version: 1.3.5												
#	A	COPY/UPDATE	ACCOUNT ID	COMPANY NAME	REGION	NETSUITE VERSION	BUNDLE ID	BUNDLE VERSION	STATUS	INSTALLED FROM	INSTALLED ON	LAST UPDATE
1		<input type="checkbox"/>	XXXXXXX	XXXXXXX	US	2014.2	6062	1.3.5		Production	08/31/2009 11:40 am	
2		<input type="checkbox"/>	XXXXXXX	XXXXXXX	US	2014.2			Not Installed	Production		
3		<input type="checkbox"/>	XXXXXXX	XXXXXXX	US	2014.2			Not Installed	Production		
4		<input type="checkbox"/>	XXXXXXX	XXXXXXX	US	2014.2			Not Installed	Production		

- Check one or more boxes in the **Copy/Update** column, and click the **Copy** button.

The **Status** column indicates whether the bundle has been previously copied in the account.

- If you check the box for an account where the bundle has not been copied, a new copy is created.
- If you check the box for an account where the bundle has been previously copied, this copy is updated with any changes.

Note: During a bundle copy, if conflicts are detected between a copied bundle's objects and existing target account objects, copied bundle objects are added and renamed. For more information about how SuiteBundler resolves conflicts, see [Resolving Conflicting Objects](#).

Reviewing Copy Bundle Status

After you click the Copy button on the Copy Bundle to Accounts page, the Copy Bundle Status page appears. This page is also available from the Saved Bundles page, by clicking Show Copies in the Action list.

Copy Bundle Status											
<input type="button" value="OK"/> <input type="button" value="Refresh"/>											
Name: Test Bundle ID: 57380 New Version:											
#	A	ACCOUNT ID	COMPANY NAME	REGION	NETSUITE VERSION	BUNDLE ID	BUNDLE VERSION	STATUS	INSTALLED FROM	INSTALLED ON	LAST UPDATE
1		XXXXXXX	XXXXXXX	US	2014.2	6062			Production	07/23/2014 11:35 am	
2		XXXXXXX	XXXXXXX	US	2014.2	6061		Pending...	Production		

- The copied bundle's status displays as Pending, whereas previously copied bundle(s) display a green check mark to indicate that the copy action is complete.
- Each copied bundle has a new bundle ID, different from the ID of the source bundle.

Notes about Copied Bundles

- A copied bundle is available from the Saved Bundles list in the new account and has the same actions available as bundles created in that account. It includes a Copied From column with the bundle ID and account ID of the bundle from which it was copied, in the format <Bundle ID> Account:<Account ID>.
- When you copy a bundle that includes locked objects, the objects are not locked in the account where the bundle is copied. However, their locked settings are maintained on the [Step 4 Set Preferences](#)

page of the copied bundle, and are enforced in target accounts where the copied bundle is installed, unless the settings are changed in the copied bundle. See [Locking Objects in Customization Bundles](#).

- When you copy a bundle that includes hidden scripts, the scripts are not hidden in the account where the bundle is copied. However, their hidden property is enforced in target accounts where the copied bundle is installed. See [Protecting Your Bundled Server SuiteScripts](#).
- Bundle object preferences set in the source account are not applied to the account where the bundle is copied. Note that data is always replaced and script deployments are always updated during an update of a bundle copy. However, source account preferences are maintained on the [Step 4 Set Preferences](#) page of the copied bundle, and are applied in target accounts where the copied bundle is installed, unless preferences are edited in the copied bundle. See [Bundle Object Preferences](#).
- The initial availability of a copied bundle is always set to Private, no matter what the availability of the source bundle was, and any shared account IDs listed for the source bundle are cleared from the copied bundle. Copied bundle availability can be reset to Shared or Public as needed, and the account ID list can be cut and pasted from the source bundle, if desired.
 - To change the availability level and add account IDs for a copied bundle, click Set Availability in the Action list. See [Sharing a Bundle](#).
 - To make other changes to a copied bundle, click Edit in the Action list. See [Editing a Bundle](#).
- You can differentiate objects added to your account through bundle copy from other account objects, because they have a bundle ID listed in the From Bundle column of list pages. See [Identifying Bundle Objects in Target Accounts](#).
- When you remove any object from a bundle and then copy the bundle to the target account, the object is not removed from the target account if the object is also included in any other bundle copied to the account.

Pushing a Bundle to Other Accounts

Bundle authors can push their bundles to any NetSuite account to which they have administrator access. This enables NetSuite resellers and independent software vendors to install bundles without having to log in to each of their customers' accounts.



Important: The Push action is not available in sandbox accounts. If you want to install a bundle from a sandbox account to another account where you have administrator access, you will need to go to that target account to search for and install the bundle. The Push action is only available in production accounts and development accounts and bundles can only be pushed to production accounts and development accounts.

When you push a bundle installation or update, the preferences you set in the Bundle Builder for custom fields, custom lists, custom records that include data, and SuiteScript deployments are applied directly to target accounts. Users in these accounts cannot override your preferences. See [Bundle Object Preferences](#).

To push a bundle to another account:

1. Go to Customization > SuiteBundler > Create Bundle > List.
2. Select **Push** from the **Action** list for the bundle you want to push.

On the Push Bundles to Accounts page, NetSuite displays all of the accounts to which you have administrator access, with details about each account, including the status of the selected bundle's installation.

You can filter the target accounts displayed on this page by region and bundle version. See [Filtering on Bundle Install Base Lists](#).

#	ACCOUNT ID	COMPANY NAME	REGION	NETSUITE VERSION	BUNDLE ID	BUNDLE VERSION	STATUS	INSTALLED FROM	INSTALLED ON	LAST UPDATE	TERMS ACCEPTED ON
1	XXXXXX	XXXXXX	US	2014.2	6061	1.2.3	✓	Repository	07/22/2014 10:44 pm		
2	-----	XXXXXX	US	2014.1	6061	1.2.3	✓	Production	01/20/2010 3:59 pm		

3. Check the **Include** box for each account where you want to install the bundle.
4. Click the Push button.

After you have pushed the bundle, the Bundle Install Base page appears. This page includes the status of the bundle installation on each account.

Viewing Your Bundle Install Base

You can view details about the accounts that have installed your bundle on the Bundle Install Base page. This page includes the following information for each account:

- NetSuite account ID
- Company name
- Geographic location
- Current NetSuite version
- Bundle ID of the installed bundle
- Version of the installed bundle
- Bundle installation status
- Account type from which bundle was installed
- Bundle installation date

Note: The date value is hardcoded to Pacific Time Zone and does not reflect the time zone setting on Home > Set Preferences.

- Most recent bundle update
- Most recent date when terms of service were accepted

Note: The date value is hardcoded to Pacific Time Zone and does not reflect the time zone setting on Home > Set Preferences.

To view the Bundle Install Base page:

1. Go to the Saved Bundles page at Customization > SuiteBundler > Create Bundle > List.
2. In the **Action** list for a bundle, click **Show Install Base**.

#	ACCOUNT ID	COMPANY NAME	REGION	NETSUITE VERSION	BUNDLE ID	BUNDLE VERSION	STATUS	INSTALLED FROM	INSTALLED ON	LAST UPDATE	TERMS ACCEPTED ON
1	XXXXXX	XXXXXX	US	2014.2	6061	1.2.3	✓	Repository	07/22/2014 10:44 pm		
2	-----	XXXXXX	US	2014.1	6061	1.2.3	✓	Production	01/20/2010 3:59 pm		

You can filter the target accounts displayed on this page by region and bundle version. See [Filtering on Bundle Install Base Lists](#).

Filtering on Bundle Install Base Lists

You can filter the target accounts displayed on bundle install base lists. These filtering capabilities help you to review target accounts for bundles you have created, and help you to manage push installs and managed bundle upgrades. **Region**, **Bundle Version**, and **Account Type** filters are available on the following pages:

- **Show Install Base page:** This page is available when you click the **Show Install Base** action for a bundle for a bundle on the Saved Bundles page, at Customization > SuiteBundler > Create Bundle > List.
- **Initiate Upgrade page for managed bundles:** This page is available when you click **Upgrade Install Base** for a bundle on the Managed Bundles page, at Customization > SuiteBundler > Managed Bundles. A **NetSuite Version** filter is also available on this page.
- **Push Bundle to Accounts page:** This page is available when you click the **Push** action for a bundle on the Saved Bundles page, at Customization > SuiteBundler > Create Bundle > List. Note that this page lists all accounts to which the currently logged in user has administrator access.
- **Copy Bundle to Accounts page:** This page is available when you click the **Copy** action for a bundle on the Saved Bundles page, at Customization > SuiteBundler > Create Bundle > List.

Currently, options in the **Region** dropdown list may include: US, UK, and APAC. The regions that are available depend on the regions where your bundle install base accounts are located, (or for push installs, on the regions where the accounts to which you have administrator access are located).

Options in the **Bundle Version** dropdown list include the version numbers that have been assigned to the selected bundle.

Options in the **Account Type** dropdown list include Test Drive Account and Production Account.

Options in the **NetSuite Version** dropdown list (Initiate Upgrade page for managed bundles only) include the current leading and lagging versions.

Editing a Bundle

After you have created a new bundle, you can return to the Bundle Builder to make changes as needed.

To edit a bundle:

1. Go to the Saved Bundles page at Customization > SuiteBundler > Create Bundle > List.
2. In the **Action** list for a bundle, click **Edit**.
3. In the Bundle Builder, you can do the following:
 - On the [Step 1 Bundle Basics](#) page, you can modify basic bundle definitions. You can also delete the bundle. See [Deleting a Bundle](#).
 - On the [Step 2 Bundle Properties](#) page, you can add or change the bundle documentation file, modify the bundle description, and modify bundle-specific terms of service.
 - On the [Step 3 Select Objects](#) page, you can change the objects that are included in the bundle.

- On the [Step 4 Set Preferences](#) page, you can set bundle object preferences to ensure they are set up properly in target accounts, and lock bundle objects so that installed bundle users cannot make changes to these objects.
4. When you are done making changes, click the **Save** button on the Set Preferences page. If you do not want to overwrite the original bundle, you can click **Save As** to save the edited bundle with a different name and a different bundle ID.
- Warning:** You must enter a different name on the first page of the Bundle Builder ([Step 1 Bundle Basics](#) page) to use the **Save As** option.
5. You can click a link on the Finished page to make changes to bundle availability. See [Bundle Availability](#). If you only need to make changes to bundle availability, you do not need to go through the Bundle Builder. On the Saved Bundles page, you can click **Set Availability** in the **Action** list.

If you begin to edit a bundle in the Bundle Builder, go away from the Bundle Builder, and then begin editing the bundle in a new instance of the Bundle Builder, the process is restarted from the first step.

Deleting a Bundle

To delete a bundle, you need to open the bundle in the Bundle Builder. You should not delete a bundle that has already been installed in target accounts.

To delete a bundle:

1. Go to the Saved Bundles page at Customization > SuiteBundler > Create Bundle > List.
2. In the **Action** list for a bundle, click **Edit**.
3. On the [Step 1 Bundle Basics](#) page, click the **Delete** button.

Warning: You should not remove a bundle from the copy chain while the older version of the bundle is installed by some users and that version is deprecated by a newer version of the bundle. Removing a bundle from the copy chain results into users not being able to install the latest version of the bundle. To understand the chaining of bundle copies, see [Ancestry of Copied Bundles](#).

Deleting a Copied Bundle

When you delete a copied bundle, you are directed to the Bundle Install Base page, where the deleted bundle may still be listed. This page does not provide any indication that the deletion is in progress or any updates on the status of the bundle deletion. If the deletion fails, the bundle will remain listed and there is no way to see the Failed status. If the deletion succeeds, the bundle will no longer be listed after a page refresh.

To check the status of the deletion of a copied bundle:

1. In the account for the source bundle, go to the Saved Bundles page at Customization > SuiteBundler > Create Bundle > List.
2. In the **Action** list for a bundle, click **Show Copies**.

This page displays a status of Pending during the time that the deletion is in progress. If the deletion fails, it displays a status of Failed with the related error reference.

Bundle Details

You can view details about a bundle you have created or installed on the Bundle Details page, which is available from the Saved Bundles page and the Installed Bundles page. To view the Bundle Details page from one of these pages, click the bundle name link. To view the Installed Bundles page, go to Customization > SuiteBundler > Search & Install Bundles > List.

The screenshot shows the Oracle NetSuite interface with the title 'Bundle Details'. The main content area displays 'Custom Fields' with the following details:

NAME	NO. INSTALLS
Custom Fields	0
ID	PRODUCT
456	NetSuite OneWorld
COPIED FROM	VERTICAL
VERSION	LANGUAGES
2.3	English (U.S.)
COMPANY	AVAILABILITY
One World	Private
<input type="checkbox"/> MANAGED	ADMIN DOCUMENTATION
AVAILABLE SINCE 7/6/2017	CS NetSuite-Copy.docx

Below the table, there are tabs for 'Overview', 'Components', 'SuiteApp Info', and 'Messages'. The 'Overview' tab is selected. The abstract section contains the text: 'Custom fields for use on entity forms'. The description section contains the text: 'This bundle adds custom fields to entity forms.'

- The **Overview** subtab displays the bundle abstract and description. It also includes a link to any bundle-specific terms of service. See [Bundle Terms of Service](#).
- The **Components** subtab lists bundle objects, including their custom script IDs, referencing objects, and lock status.
 - A bundle may include objects that were not explicitly added by the bundle author, because the Bundle Builder automatically pulls into the bundle any objects on which selected objects depend. These referenced objects display a chain icon. In contrast, objects that have been selected in the Bundle Builder have a check mark icon.
 - In target accounts, the **Components** subtab includes an option to change the setting for whether to hide or show bundle components in lists.



See [Choosing to Hide or Show Bundle Components](#).

- The **SuiteApp Info** subtab displays details about the bundle from SuiteApp.com, if the bundle is linked to that site.
- The **Messages** subtab displays messages that have been sent to this bundle's install base.
 - If this is a bundle you have created, on this subtab you can create and send new messages. See [Sending Bundle Messages](#).
 - If this is a bundle you have installed, on this subtab you can select the users in your account who should receive this bundle's messages as email. See [Subscribing to Bundle Email Messages](#).

Note: The Bundle Details page also displays when you select a bundle to install, for your review before the installation. In this case, this page does not include the Messages subtab.

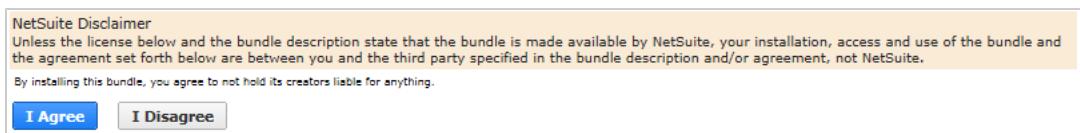
- If this is a bundle you have installed, the Bundle Details page includes an **Audit Trail** subtab that tracks installations, updates, and uninstalls of this bundle in your account. For details, see [Viewing the Bundle Audit Trail](#).
- If this is a managed bundle you have created, the **Email Recipients** field indicates users who should receive upgrade status notifications. By default the user who initiated the upgrade always receives the notifications. For more details, see [Notifications for Managed Bundle Upgrades](#).

Bundle Terms of Service

The terms of service associated with a bundle are available from a link on **Overview** tab of the Bundle Details page.



You can click **View** to display bundle-specific terms of service.



- If this is a bundle you have created, clicking **View** displays the terms of service you have most recently entered in the Bundle Builder. For details, see [Defining Bundle Terms of Service](#).

- If this is a bundle you are considering installing, clicking **View** displays the terms of service you need to accept to proceed with the installation.
- If this is a bundle you have installed, clicking **View** displays the terms of service most recently accepted by a user in your account.
 - After an update, managed bundle users can choose to agree or disagree to terms when they view them here. (The Terms of Service dialog is not displayed before managed bundle updates, because these updates are pushed automatically.)



Note: The terms of service available on the Bundle Details page are specific to the individual bundle, and are in addition to the general SuiteBundler terms of service.

Sending Bundle Messages

For any bundle you have created, you can communicate directly with your install base by sending messages about the bundle. You can create, send, and review messages on the **Messages** subtab of the Bundle Details page.

Bundle messages are displayed on the Bundle Details page in selected target accounts, and may also be sent as email to target account bundle administrators. If you are a target account bundle user, see [Subscribing to Bundle Email Messages](#) for details.



Warning: You cannot send bundle messages to or from Sandbox accounts.

To create and send a bundle message:

1. Go to Customization > SuiteBundler > Saved Bundles > List.
2. Click a bundle name to display its Bundle Details page.
3. On the **Messages** subtab of the Bundle Details page, click **New Message**.
4. If desired, change the email address to be displayed in the **From** field of the message.
 - The default email address in the **From** field for the first bundle message sent from a source account is the email address of the currently logged in user.
 - After this email address has been modified, it is used for later bundle messages from that source account until it is reset again.
 - The **Reply-To** value for each bundle message is set to the email address in the **From** field, and bounced messages are returned to this email address.



Note: For messages created prior to Version 2012 Release 2, the **From** field is blank by default.

5. Enter a subject and message text.

You can enter HTML or text. Rich text editing also is available if your user preference is set to enable it. (See the help topic [Personal Preferences for Appearance](#).)

6. The **Recipients** subtab lists accounts where the bundle is installed. Check the box for each account that you want to receive this message.
 - You can filter the target accounts displayed on the **Recipients** subtab, to help you find the accounts that should receive a message. **Region** and **Bundle Version** filters are available.
7. After you have entered text and selected recipient accounts, do one of the following:
 - If you are not ready to send the message, click **Save as Draft**.

- If you are ready to send the message, click **Save & Send**.

When you send the message:

- It is listed on the Bundle Details page in your account, with the number of install base accounts to which it has been sent, the number of accounts to which it has not been sent, and the date and time when it was last sent.
- It is listed on the Bundle Details page in the target accounts you selected.
- Each administrator and each user with SuiteApp Marketplace permission who are subscribed to this bundle's messages in target accounts receive an email. See [Subscribing to Bundle Email Messages](#).



Important: After you have sent a bundle message, you can no longer edit its subject or text. You can select additional accounts to receive the identical text, but if you want to change the text, you need to create a new message. Also, you cannot resend a message to an account that has already received it.

If you deprecate a bundle, after a target account installs the replacement bundle, its Bundle Details page **Messages** subtab no longer lists any messages sent for the deprecated bundle. Only messages sent for the replacement bundle are displayed. You can continue to send messages from a deprecated bundle, but these are only sent to accounts that have not yet updated to the replacement bundle.

If you delete a bundle in the source account, any messages sent for that bundle are removed from the Bundle Details page **Messages** subtab in all install base accounts.

SuiteApp Installation and Update

If you are the administrator of a NetSuite account or you have the SuiteApp Marketplace permission, you can install any public bundle, as well as any bundle that has been shared with your account.

- When you install a customization bundle, objects included in that bundle are created in the target account.
- When you install a configuration bundle, configuration settings included in the bundle are copied from the source account to the target account, overwriting existing configuration settings. Because the installation of a configuration bundle changes the setup of your account, make sure that you understand the changes that result from installation.

Before you can install a bundle, you need to find it in a NetSuite source account. See [Bundle Searches Overview](#) for a description of bundle search capabilities. See [Installing a Bundle](#) for an overview of the installation process. See [Choosing a Bundle to Install](#) for instructions for finding and installing a bundle in your account.



Tip: Some SuiteApp bundles are available on the SuiteApp Marketplace. For more information, see the help topics [Installing from the SuiteApp Marketplace](#) and [SuiteApp Marketplace](#).

The Installed Bundles page provides details about all of the bundles currently installed in an account, and an action menu. See [Reviewing the Installed Bundles List](#).

When a new version becomes available for a bundle that you have installed, you can update that bundle from a link on the Installed Bundles page. When you update a customization bundle, new bundled objects are added to your account, and updates to bundled objects are copied to your account. When you update a configuration bundle, changes to bundled configuration settings are copied to your account. See [Installed Bundle Updates](#) for instructions for updating an installed bundle.

When you install a bundle that is a managed bundle, you consent to automatic bundle updates pushed by the solution provider. See [Using Managed Bundles](#).

If after installing a customization bundle you decide you do not want the bundle, you can uninstall it from a link on the Installed Bundles page. You cannot uninstall a configuration bundle. For more information, see [Uninstalling a Bundle](#).

SuiteBundler provides options during bundle installations and updates for handling any customization bundle objects that conflict with existing target account objects, meaning they have the same names or script IDs. See [Resolving Conflicting Objects](#).

The Bundle Audit Trail page provides tracking of bundle installations, updates, and uninstalls on the account level. The Audit Trail subtab of the Bundle Details page provides this tracking on a per-bundle level. See [Viewing the Bundle Audit Trail](#).



Important: Bundles are also known as SuiteApps.



Note: You can use SuiteScript to get information about installed bundles and SuiteApps. For more information, see the help topic [N/suiteAppInfo Module](#).

Resolving Conflicting Objects

When you install a customization bundle, the objects in the bundle are checked against preexisting custom objects in your account to identify possible conflicts. In the same manner, when you update

a previously installed customization bundle, any newly added bundle objects are checked against preexisting custom objects in your account. The results of this check are displayed on the Preview Bundle Install page or Preview Bundle Update page. When you review these pages, you are presented with options for how to resolve these conflicts.

For managed bundles, you cannot override bundle installation preferences during bundle installation. For more information, see [Bundle Object Preferences](#).



Important: The process for updating bundles installed into production from sandbox is different from the update process for other bundles, to meet the specialized requirements of bundles developed in sandbox accounts. See [Selective Update of Sandbox Bundle Objects](#).



Note: The process for updating bundles installed into a sandbox account from a development account is the same as the update process for other bundles. The normal bundle update process updates all bundle objects in the target account, overwriting target account objects with source bundle objects. For bundle updates from a development account to a sandbox account, the Preview Bundle Update page always shows the Update action for all objects, even for objects that have not changed in the source account since the last bundle update.

How Conflicts Are Identified

Conflicts can exist when objects in the bundle have the same script IDs or the same names as preexisting objects in the account. Generally, a conflict exists for one of the following reasons:

- The same object already exists in the account, typically because it was already installed as part of another bundle installation.
- A different, unrelated object coincidentally has the same ID or name.

NetSuite identifies conflicts first by the script ID of the object, or when there is no script ID, by the name and type of the object. For example, custom fields, which have script IDs, are identified by their script IDs, whereas custom forms, which do not have script IDs, are identified by their form name and type.

Bundle authors can avoid potential conflicts with target accounts' custom objects by giving each bundle's custom objects names unique to their function in that bundle.

Options for Resolving Conflicting Objects

For each bundle object that is identified as conflicting, you can choose one of the following actions on the Preview Bundle Install page and Preview Bundle Update page:

- **Add and Rename** the bundle object, leaving the preexisting custom object intact, or
- **Replace Existing Object** with the bundle object.

Typically, you would choose Add and Rename if the conflicting ID or name is a coincidence, and you would choose Replace Existing Object if you are sure the bundle object and the preexisting object in the account are the same object. You can still choose Add and Rename if the objects are the same, effectively creating a duplicate object.

Installation of a bundle from a sandbox account to a production account almost always involves conflicting objects. In this case, you usually would choose the Replace Existing Object option.



Warning: For any bundle object conflict where the Preview Bundle Update page displays both a Delete action, and a dropdown to select either Add and Rename, or Replace Existing Object, a bundle update causes deletion of the object and removal of its data from the target account.

Impact of Choosing Add and Rename

If you choose to add and rename a bundle object, a number is appended to the bundle object's name or script ID, to differentiate it from the preexisting custom object. Additionally, all references to the custom object in SuiteScript change to reflect this renaming. For example, if the bundle includes a custom field with the ID **custentity_deptaddr** and the target account has a custom field with the same ID, then the field in the bundle is given the ID **custentity_deptaddr2**, and any of the SuiteScript code in your bundle that referenced custentity_deptaddr is updated automatically to reference custentity_deptaddr2 instead.

Impact of Choosing Replace Existing Object

For a custom field, choosing Replace Existing Object generally does not have any impact on data that has been entered to the existing field, if the bundle field has the same data type as the existing field. This option causes the field definition to be updated to match the definition for the bundled field. If the bundled field has a different data type, data may have to be deleted from the existing field because the bundled field type, for example, INTEGER, cannot store the existing type of data, for example, TEXT. In this case, you should probably choose the Add and Rename option to create a new field, and change the script ID so it is unique.

For custom records that include data and for custom lists, when you choose Replace Existing Object, you can set a bundle preference to Replace Data, Preserve Data, or Merge Data. The Replace Data option overwrites existing target account object data with bundle object data. The Preserve Data option, which is the default, preserves existing data in target account objects. The Merge Data option retains any data added to target account objects after bundle installation and at the same time updating data that was previously installed with the bundle. See [Set Bundle Installation Preferences](#) and [Set Bundle Update Preferences](#).

If you choose to replace a preexisting custom object with a bundle object, the bundle object keeps the same name or script ID, and the preexisting custom object no longer exists in the account. For example, if the bundle includes a custom field with the ID **custentity_deptaddr** and the target account has a custom field with the same ID, then the field in the bundle keeps the ID **custentity_deptaddr** and the preexisting custom field no longer exists in the account.



Warning: If you choose the Replace Existing Object option for a conflicting object during bundle installation, the preexisting custom object is PERMANENTLY replaced by the bundle's object. Even if you choose to uninstall the bundle later, changes to the replaced object remain and the original object is not restored.

Bundle Searches Overview

The Search & Install Bundles page, available at Customization > SuiteBundler > Search & Install Bundles, includes the following two search modes:

- Basic mode. Simple bundle searches across all production locations.
- Advanced mode. Bundle searches that support additional search filters.



Tip: To search for the most popular bundles by the number of installations, leave the Keywords box blank and click Search.

Notes:

- Basic and advanced search results include bundles that are available for you to install, meaning the bundles that have been made public or have been shared with the account to which you are currently logged in. Search results will also include additional bundles that have not been shared with your account but have been marked "Visible by All".
- Bundles returned in search results are sorted by the number of installations in target accounts (No. Installs), in descending order.
- If you use advanced search and the search returns multiple pages of results, you should alter the search criteria to further limit the number of bundles returned by the search.
- Basic search does not search across Sandbox accounts; to search for a bundle in a Sandbox account, you must use advanced search.
- A basic search may return multiple instances of a single bundle, if that bundle is available in multiple NetSuite accounts.
- Support for bundle searches varies across the different types of NetSuite accounts. See [Bundle Support Across Account Types](#).

Basic Mode

The **Basic** search mode, the default, supports simple bundle searches across all production locations. Use a basic search when you are not sure where a bundle is located and you do not have much information about its other characteristics.

Search & Install Bundles							
<input type="button" value="Search"/> Basic Advanced							
Leave the keywords box blank and click Search to view the most popular SuiteApps							
Keywords <input type="text"/>							
<u>Installation Terms of Service</u>							
1 to 25 of 4718 ▾							
NAME	BUNDLE ID	VERSION	MANAGED	COMPANY NAME	CREATED ON	AVAILABILITY	SUITEAPP.COM
Last Sales Activity	53195	1.09.0	Yes	Netsuite Platform Solutions Group - Last Sales Activity	23-APR-14	Public	7,935
Pivot Reports BETA	17447	1.01	Yes	Netsuite - Pivot Reports BETA	11-OCT-11	Public	3,079
Enhanced Sales Center	19656	2.00.0		Netsuite Platform Solutions Group - Enhanced Sales Center	23-JAN-12	Public	2,210
NetSuite Electronic Payments	45178	4.00.1	Yes	Netsuite Platform Solutions Group - NetSuite Electronic Payments	30-OCT-13	Public	1,567
NetSuite for Outlook	26153	1.18.0	Yes	Netsuite - Netsuite for Outlook Integration	02-AUG-12	Public	1,388
International Tax Reports	14864	3.60.0	Yes	Netsuite Platform Solutions Group - International Tax Reports	30-MAY-11	Public	1,085
Advanced Promotions	49247	2.04.0	Yes	Netsuite Platform Solutions Group - Advanced Promotions	26-JAN-14	Public	1,010

Advanced Mode

The **Advanced** search mode supports additional search filters. Use an advanced search when you know the Account ID where a bundle is located or you want to narrow the search based on other bundle characteristics. By default, this search includes all production locations.

Search & Install Bundles

Search

Basic | Advanced

LEAVE THE KEYWORDS BOX BLANK AND CLICK SEARCH TO VIEW THE MOST POPULAR SUITEAPPS

KEYWORDS	PRODUCT
	All
LOCATION	VERTICAL
All Production	All
AVAILABILITY	LANGUAGE
All	All
PUBLISHER ID	

Installation Terms of Service

NAME	BUNDLE ID	VERSION	MANAGED	COMPANY NAME	PUBLISHER ID	CREATED ON	AVAILABILITY	NO. INSTALS
No SuiteApp search performed.								

The following table describes the filters for an advanced search:

Filter	Description
Location	Where the source bundle is located -- choose from the following options: <ul style="list-style-type: none"> ■ All Production: (the Default) Lists all bundles to which you have access across all production accounts. ■ Production Account: Lists bundles to which you have access in a selected NetSuite production account. You need to provide an account ID when you choose this filter. ■ Sandbox Account: Lists bundles to which you have access in a selected sandbox account. You need to provide an account ID when you choose this filter.
Availability	The options are All, Public, Shared, or Private.
Keywords	You can search for bundles by keywords included in the bundle's Name, Bundle ID, Company Name, or Description
Product	The NetSuite product for which the bundle is intended, for example, NetSuite OneWorld.
Vertical	The vertical market for which the bundle is intended, for example, IT Services.
Language	The language translations included in the bundle. For example, a bundle may include translations in French and English of custom records, custom forms, or other elements in the bundle.
Publisher ID	Applicable to bundles created from SDF SuiteApp projects. The publisher ID associated with the SuiteCloud Developer Network (SDN) member who created the SDF SuiteApp project on which the bundle is based.



Note: If you leave the Keywords box blank for an advanced search, NetSuite returns the most popular bundles that meet the filter criteria.

Installing a Bundle



Tip: Some SuiteApp bundles are available on the SuiteApp Marketplace. For more information, see the help topics [Installing from the SuiteApp Marketplace](#) and [SuiteApp Marketplace](#).

If you are an account administrator or you have the SuiteApp Marketplace permission, you can complete the following steps to install a bundle in your account.

1. Go to Customization > SuiteBundler > Search & Install Bundles, then choose the bundle you want to install.

See [Choosing a Bundle to Install](#) for instructions to search for and install the bundle of your choice.

See [Bundle Searches Overview](#) for a description of bundle search functionality.

2. Review the Preview Bundle Install Page.

This page lists the bundle objects that will be installed (or, for configuration bundles, the settings that will be copied). Here, you can indicate how to resolve conflicts between bundle objects and target account objects with the same names or script IDs, set installation preferences for some objects, and lock some objects so target account users cannot change them. For details, see [Reviewing the Preview Bundle Install Page](#).

3. When you are ready, click the Install Bundle button to begin the installation process.

For details, see [Starting the Bundle Installation Process](#).



Note: You should review [Bundle Installation Notes](#) before you begin the process of installing a bundle.

4. After you have installed a bundle, you may want to complete the following tasks:

- [Subscribing to Bundle Email Messages](#)
- [Choosing to Hide or Show Bundle Components](#)



Note: Support for bundle installations varies across the different types of NetSuite accounts. See [Bundle Support Across Account Types](#).



Note: When you install a bundle, you become the owner of the bundle and all its related objects.

Choosing a Bundle to Install

For an overview of the bundle installation process, see [Installing a Bundle](#). Before you choose a bundle to install, you should review [Bundle Installation Notes](#).

To choose a bundle to install:

1. Log in to the account where you want to install the bundle.

Bundles can be installed in production accounts and sandbox accounts.

2. Go to Customization > SuiteBundler > Search & Install Bundles (Administrator).
 3. Search for a bundle to install.
- See [Bundle Searches Overview](#).
4. Optionally, click the **Name** of a bundle to display its Bundle Details page.
 - Click **Documentation** to review documentation for the bundle.
 - The **Overview** subtab displays the bundle abstract, and description. It also displays a link to any bundle-specific terms of service. See [Bundle Terms of Service](#).
 - The **Components** subtab lists bundle objects, including their custom script IDs, referencing objects, and lock status.
 - The **SuiteApp Info** subtab displays details about the bundle from SuiteApp.com, if the bundle has been developed by a NetSuite partner and is available on that site.
 - For more information, see [Bundle Details](#).
 5. When you have decided that you want to install a bundle, click the **Install** button.
 - The **Install** button is only available for bundles that have been made publicly available to all NetSuite accounts, and for bundles that have been shared explicitly with your account. If this button is not available, contact the solution provider to inquire about access to the bundle.
 - If the bundle is a managed bundle, you must agree to allow the bundle author to upgrade the bundle in your account when new updates are available. Managed bundles are indicated in the **Managed** column of the Installed Bundles page. For more information, see [Using Managed Bundles](#).
 - If the bundle has been deprecated, a notification message appears, indicating the replacement bundle that you should install instead. A deprecated bundle is a bundle that has been replaced by a later bundle version.
 - If this bundle is already installed in the account, or this bundle is a copy of a bundle that is already installed, an error message displays.
 6. After you have clicked the **Install** button, the Bundle Details page displays. Click the **Install** button on this page.
 - If the bundle has terms of service that require acceptance, a dialog displays them. Click **I Agree** to proceed.
 - The Preview Bundle Install page displays. See [Reviewing the Preview Bundle Install Page](#).

Reviewing the Preview Bundle Install Page

After you finish [Choosing a Bundle to Install](#) and click the **Install** button for a bundle, the Preview Bundle Install page displays.

Preview Bundle Install						
Cancel		Install Bundle				
Name: Customer Support Bundle Version: 1.1						
BUNDLE CONTENTS	EXISTING OBJECT IN CONFLICT	SCRIPT ID	ACTION	PREFERENCE	INTERNAL ID	LOCK ON INSTALL
Custom Fields						
CRM Fields						
Frequent Support User?		custevent1_1	Add	<input checked="" type="checkbox"/> Hide on Existing Custom Forms	custevent1_1	
Custom Lists/Records						
Lists						
Help Use Occurrence		customlist7	Add		7	
Records						
Training Level		customrecord_level	Add		27	
File Cabinet						
Files						
CaseClosureEmail.js			Add		159	
field_changed.js			Add		70	
Forms						
Entry Forms						
Phone Call Form - Automation			Add		14	
SuiteScripts						
User Event						
ClosedCase		customscript11	Add		213	

On this page, you can review a list of bundle objects to be installed. For customization bundles, you also can do the following:

- Choose Actions for Conflicting Objects
- Set Bundle Installation Preferences
- Review Locked Bundle Objects

i Note: When you have completed your review of the Preview Bundle Install page, you can click the **Install Bundle** button. See [Starting the Bundle Installation Process](#).

Choose Actions for Conflicting Objects

The Preview Bundle Install Page indicates any customization bundle objects that conflict with existing custom objects in the target account. A conflict occurs when a bundle object has either the same script ID, or in some cases, the same name and type, as an existing custom object in the target account. For more information, see [Resolving Conflicting Objects](#).

NetSuite displays conflicting objects, in red, in the **Existing Object in Conflict** and **Script ID** columns. For each conflicting object, choose one of the following from the dropdown list in the **Action** column:

- **Add and Rename** – NetSuite appends a number to the bundle object's name or script ID, to differentiate it from the preexisting custom object. The bundle object is added to account, and the preexisting object remains intact.
- **Replace Existing Object** – The bundle object keeps the same name or script ID. The bundle object is added to the account and the preexisting custom object no longer exists in the account.

If you choose this option for a conflicting object during bundle installation, the preexisting custom object is PERMANENTLY replaced by the bundle's object. Even if you choose to uninstall the bundle later, changes to the replaced object remain and the original object is not restored.

For each custom record that includes data, and each custom list, where this option is chosen, a preference dropdown list is displayed so you can choose whether to replace, preserve, or merge data in the target account object to be replaced. See [Set Bundle Installation Preferences](#).

Set Bundle Installation Preferences

The following preferences may be available on the Preview Bundle Install page:

- **Show or Hide Custom Fields**

For each custom field in a bundle to be installed, you can set a preference indicating whether the field should be shown or hidden on existing custom forms in target accounts. The setting chosen by the bundle author is displayed by default.

Note that this setting does not affect bundled forms, because all custom fields in a bundle are always shown on all custom forms included in that bundle.

- **Replace Data, Preserve Data, or Merge Data for Conflicting Custom Records and Custom Lists**

For each custom record that includes data and for each custom list, when you choose **Replace Existing Object**, you can set a bundle preference to either **Replace Data** or **Preserve Data**. The **Replace Data** option overwrites existing target account object data with bundle object data. The **Preserve Data** option preserves existing data in the target account object. The **Merge Data** option retains any data added to target account objects after bundle installation and at the same time updating data that was previously installed with the bundle. The setting chosen by the bundle author is displayed by default.

Review Locked Bundle Objects

When a bundle contains locked objects, the Preview Bundle Install page displays a lock icon in the **Lock on Install** column.

A few important points to notice about the Lock on Install column on the Preview Bundle Install page:

- A lock icon appears on the row of a locked object in the bundle.
- If no lock icon appears on the row, then the object is not locked. An object may not be locked because:
 - The bundle author chose not to lock the object.
 - Or, because the object is not lockable.

Pages for locked objects do not have buttons that allow changes to the object, such as **Edit**, **Delete**, **Change ID**. Also all list and view pages for these objects display a lock icon. For more information, see [Effects of Locking Different Object Types](#).

Starting the Bundle Installation Process

After you have started the process of installing a bundle in a target account by clicking the **Install Bundle** button on the Preview Bundle Install page, the Installed Bundles page displays.

- If the installation is not complete, the **Status** column displays the percentage of installation progress.
- If the installation is complete, the **Status** column displays a green check.

The time required to install the bundle depends on the size of the bundle. You can continue work in NetSuite during the time that the bundle installs.



Warning: Enabling the **Include Data** option for a bundled custom record object with an extremely large number of custom records may result in bundle installation failure due to network connection problems. To ensure installation success, do not enable the **Include Data** option for the custom record object. Then run a bundle installation script that executes after installation to import custom record object data into the target account through CSV import. For information, see the help topics [SuiteScript 2.x Bundle Installation Script Type](#) and [task.CsvImportTask](#).

You can go back to the Installed Bundles page and check the status of the bundle installation later, at Customization > SuiteBundler > Search & Install Bundles. See [Reviewing the Installed Bundles List](#).

Bundle Installation Notes

You should be aware of the following before you install a bundle in your account.

Difference in NetSuite Versions for Source and Production Accounts

During phasing of a new NetSuite release, bundles developed with the lagging version of NetSuite can be installed into accounts that are already using the leading version. Also, bundles developed with the leading version can be installed into accounts that are still using the lagging version, but be aware of the following limitations:

- When a user in a lagging version account installs a bundle from a leading version account, any object types in the bundle that are not supported in the lagging version are not installed, resulting in potentially serious impact to bundle function.
- A bundle with a SuiteScript that uses a new API available only in the leading version can be installed into a lagging version account, but the script may not function correctly because the lagging version does not support the API.

Restrictions for Release Preview Account Bundles

Be aware that there are limitations on the types of accounts from which you can install bundles into Release Preview accounts. Also be aware that bundle support differs between Release Preview accounts for customer accounts hosted in North American data centers, which remain on the beta domain, and Release Preview accounts for customer accounts hosted in European Union data centers. For more details, see [Bundle Support Across Account Types](#).

Pushed Installations

Bundle authors with administrator access to your account may install bundles directly without any actions required by you. See [Pushing a Bundle to Other Accounts](#).



Important: The **Push** action is not available in sandbox accounts. If a bundle author wants to install a bundle from a sandbox account to your account where they have administrator access, they will need to go to your account to search for and install the bundle, or you can take this action yourself.

Bundled Server SuiteScripts

If an installed bundle includes server SuiteScripts, you might not be able to view the script files, depending on whether the bundle's creator enabled an option in the script file's File Cabinet record to

hide it in bundle installations. You can see the file records for hidden scripts, but you cannot open their .js files to view their contents.

If an installed bundle contains SuiteScripts and there are object conflicts, choosing to add and rename bundle objects requires that the script id references are encapsulated in quotes (" or ') so that the references are correctly renamed in the scripts. For information about the impact of the operation, see [Choose Actions for Conflicting Objects](#) and [Impact of Choosing Add and Rename](#).

Bundle with SSP Applications

As of 2019.1, when you update an installed bundle that includes an SSP application, the script deployments are not updated in the target account. Because script deployments are not updated, the existing touch points are not overwritten and both the Log Level and Status fields are also not updated in the target account. The Log Level and Status fields can be viewed on the Script Deployment page.

Impact of Language Preferences on Bundled Custom Forms

When a bundle that includes a custom form with sublists showing data from custom child records is installed in a target account with a different language preference than used in the source account for the bundle, the labels for the sublists may not be translated correctly. The system attempts to obtain the translations of labels in the language of the target account, but if these translations don't exist, the translated name of the child record type is used instead. A workaround is to manually rename sublists on the custom form in the target account, but when the bundle is updated, these manual changes will be overwritten. Another workaround is to change the language preference of the target account to be the same as the source account.

Updates Required for Some Bundle Objects that Reference Custom Segments

The unified ID has an impact on bundled objects. When the Use as Field ID setting is enabled for custom segments, existing customizations that include these segments may not work as expected. Objects that may be impacted include scripts, workflows, custom Advanced PDF/HTML templates, and saved searches. As of 2019.1, creating a new custom segment automatically uses the unified ID, and the **Use as Field ID** box is not visible. For more information, see the help topic [Creating a Custom Segment](#).

To ensure the successful installation of bundled objects that reference custom segments, view the following guidelines:

Source Account	Target Account	Installation Notes
Custom segment objects created in a NetSuite version prior to 2019.1	Custom segment objects created in a NetSuite version prior to 2019.1	If the Use as Field ID setting is enabled on the segment in the source account, ensure that the objects in the bundle's target account are updated to reference the unified ID for the segment.
Custom segment objects created in NetSuite version 2019.1 or later	Custom segment objects created in a NetSuite version prior to 2019.1	The Use as Field ID setting is enabled on the segment in the source account. Ensure that the objects in the bundle's target account are updated to reference the unified ID for the segment.
Custom segment objects created in a NetSuite version prior to 2019.1	Custom segment objects created in NetSuite version 2019.1 or later	The Use as Field ID setting is enabled on the segment in the target account and cannot be changed.

Source Account	Target Account	Installation Notes
		If the Use as Field ID setting was not enabled on the source account and Replace Existing Object selected to resolve the conflict, existing customizations that include those segments may stop working or may not work as expected. Objects that may be impacted include scripts, workflows, printing templates, and saved searches.
Custom segment objects created in NetSuite version 2019.1 or later	Custom segment objects created in NetSuite version 2019.1 or later	The Use as Field ID setting is enabled on the segment in both the source account and the target account. No change is needed.

Default Values for CSV Files

When CSV files are included in a bundle, there may be a manual step needed to map default values using the Import Assistant after bundle installation. For more information, see the help topic [Including Import Maps and CSV Files in SuiteBundler](#).

Reviewing the Installed Bundles List

After you have installed a bundle, it is listed on the Installed Bundles page, available at Customization > SuiteBundler > Search & Install Bundles > List .

This page includes the following information for each bundle created in the current account:

- Bundle name
- Bundle ID
- Bundle version (If a newer version than the installed version is available, an icon displays that you can click to go to the Preview Bundle Update page and update the bundle to the latest version.)
- Whether the bundle is a managed bundle
- Abstract
- Link to bundle documentation file for administrators (Note that bundle documentation for users may be available in the NetSuite Help Center. See [Finding SuiteApp Help Content in the NetSuite Help Center](#).)
- Status of the install (completed or in progress)
- NetSuite account ID for the account from which the bundle was installed
- Type of account from which the bundle was installed (production or sandbox)
- User name of the user who installed the bundle
- Date and time when the bundle was originally installed in the account

 **Note:** The date and time value is hardcoded to Pacific Time Zone and does not reflect the time zone setting on Home > Set Preferences.

- Date and time when the bundle was last updated

 **Note:** The date and time value is hardcoded to Pacific Time Zone and does not reflect the time zone setting on Home > Set Preferences.

You can use the dropdown menu in the Action column to link to pages for other bundle tasks, including:

- Updating the installed bundle (See [Installed Bundle Updates](#).)
- Uninstalling the bundle (See [Uninstalling a Bundle](#).)

You can click the bundle name to go to the Bundle Details page:

- This page provides information about the bundle, including a link to the bundle documentation file and SuiteApp.com details, if any. See [Bundle Details](#).
- On this page's Components subtab, you can indicate whether objects installed by the bundle should be displayed in your account's list pages. See [Choosing to Hide or Show Bundle Components](#).
- This page also includes a Messages subtab that displays all messages sent by the bundle author. On this subtab you can set up the users who should receive bundle messages as email. See [Subscribing to Bundle Email Messages](#).



Note: You can differentiate objects added to your account through bundle installation from other account objects, because they have a bundle ID listed in the From Bundle column of list pages. See [Identifying Bundle Objects in Target Accounts](#).

You can also use SuiteScript to get information about installed bundles and SuiteApps. For more information, see the help topic [N/suiteAppInfo Module](#).

Finding SuiteApp Help Content in the NetSuite Help Center

Bundle authors may make user help content available in the NetSuite Help Center. This content is available from a page titled Third Party SuiteApps. You can access this documentation by going directly to this page, or through a Help Center search. The Third Party SuiteApps page includes a table of third party SuiteApps. Each SuiteApp listed in the table has a link to a child help page titled with the name of the SuiteApp. This help page links to the PDF file or files containing the SuiteApp documentation, and to the website of the author.

Choosing to Hide or Show Bundle Components

By default, list pages for bundleable objects include objects that have been installed by bundles. You can identify an object that is from a bundle, because the ID of the bundle is displayed in the From Bundle column of the list page. See [Identifying Bundle Objects in Target Accounts](#).

The creator of each bundle can set a preference indicating whether, by default, objects in the bundle should be displayed in target account list pages. After you have installed a bundle, you can review its hide or show setting, and change it if desired.

To change the hide or show setting for bundle objects:

1. Go to Customization > SuiteBundler > Search & Install Bundles > List.
2. Click the name of the bundle to display the Bundle Details page.
3. On the **Components** subtab:
 - Choose **Hide Components** if you do not want bundle objects to be displayed in your account's list pages.
 - Choose **Show Components** if you want bundle objects to be displayed in your account list pages.

This

NAME	ID	REFERENCED BY	LOCK ON INSTALL
field_changed_a.js	2865		
ClientAssistant.js	2866		
field_changed_b.js	2864		

These options are not available for configuration bundles.

Note: If you select **Show Components**, users can set the **From Bundle** filter on a list page to filter out bundle objects. See [Filtering Bundle Objects on List Pages](#).

Subscribing to Bundle Email Messages

For each bundle that you have installed, messages from the bundle author are listed on the **Messages** subtab of the Bundle Details page. To view this page, go to Customization > SuiteBundler > Search & Install Bundles > List, and click the link for a bundle name. For more information, see [Bundle Details](#).

SUBJECT	FROM	DATE RECEIVED
Welcome to Support Solutions!	admin@netsuite.com	7/24/2014 10:41

You can review a message's details by clicking its link in the **Subject** column.

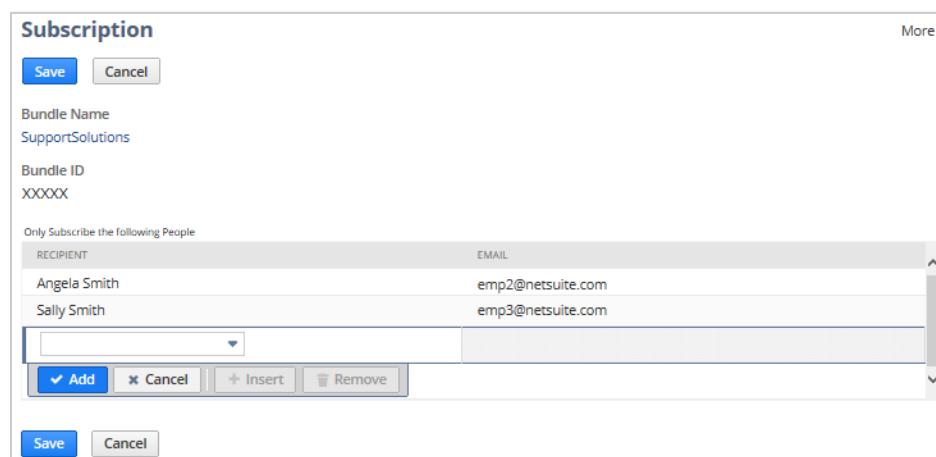
In the **Email Recipients** dropdown list, you can indicate the users in your account who should receive email copies of this bundle's messages.

- The default choice is **All Bundle Admins**, indicating that all administrators and users with the SuiteApp Marketplace permission in your account receive bundle messages as email.
- You can choose **None** to indicate that bundle messages should not be sent as email to anyone in your account.
- You can choose **Custom** to indicate that bundle messages should be sent as email only to selected bundle administrators.

1. To select these users, click **Edit List**.

On the Subscription record, administrators and other users with the SuiteApp Marketplace permission are listed in the **Recipient** dropdown list.

2. Select a user and click **Add**, repeating as necessary to add more recipients.
- Note that you cannot edit email addresses here.
3. When you are done adding users, click **Save**.



The selected users receive email for any future messages sent for this bundle, including email notifications of managed bundle upgrades to target account users.

Note that when you hover over **Edit List**, these recipients are displayed.

Note: Bundle messages are always displayed on the **Messages** subtab of the Bundle Details page, which is accessible to all bundle administrators in the account, no matter what choice is made for email recipients.

Installed Bundle Updates

If a bundle author makes changes to a bundle that has been installed in a target account, an account administrator or another user with SuiteApp Marketplace permission can update the installed bundle.

Updating a bundle creates in the target account all of the custom elements added by the developer since bundle installation. An update also includes revised versions of any objects edited by the bundle author. If you have made changes to objects installed by the bundle, a bundle update generally overwrites these changes so objects match the source. For more details, see [Bundle Update Reference](#).

Complete the following steps to update a bundle:

- Choosing a Bundle to Update
- Reviewing the Preview Bundle Update Page

Target account users cannot update a bundle that the developer has defined as a managed bundle. Updates to managed bundles are performed from the source account by the bundle author. This type of automated bundle update by the bundle author is only supported for managed bundles. Automated updates by solution providers are not supported for other bundles, which you must manually update yourself. See [Using Managed Bundles](#).

 **Warning:** When you remove any object from a bundle and then update the bundle in the target account, the object is not removed from the target account if the object is also included in any other bundle installed in the account.

 **Note:** Support for bundle updates varies across the different types of NetSuite accounts. See [Bundle Support Across Account Types](#).

Choosing a Bundle to Update

You can choose a bundle to update from the Installed Bundles page, at Customization > SuiteBundler > Search & Install Bundles > List. The Action dropdown menu includes an Update option, that you can click to update a bundle.

SuiteBundles to Update Reminder

You may see a reminder in the Reminders portlet indicating that one or more installed bundles have an updated version available.



When the first bundle is installed in an account, this reminder is added for an account's administrators and for users with SuiteApp Marketplace permission. This reminder only displays when at least one installed bundle has an updated version available.

You can click this reminder to go to the Installed Bundles page, where you can click a link to update a bundle.

You can remove this reminder from the portlet by clicking **Set Up**, clearing the box next to SuiteBundles to Update, and clicking Save. Click the Reminders portlet Refresh icon to see the change.

Update Available Icon on Installed Bundles Page

The Installed Bundles page includes an alert icon in the Version column for any bundle that has a later version than the version currently installed in your account. You can click the Update option in the Action dropdown menu to go to the Preview Bundle Update page and update the bundle to the latest version. See [Reviewing the Preview Bundle Update Page](#).

Installed Bundles					
		New	Refresh		
Action	Name	Bundle ID	Version	Managed	Abstract
	Customer Support Bundle	57487	1.1		<div>Update Available</div>
	SupportSolutions	58005	1.0		<div>Version 1.2 has been released</div>
	SuiteSocial	43437	1.06.0	Yes	<div>SuiteSocial is collaboration to more tight interactions and track im</div>

Reviewing the Preview Bundle Update Page

After you click the Update option for a bundle on the Installed Bundles list page, a Preview Bundle Update page opens for the bundle, listing objects to be added and updated.



Important: The process for updating bundles installed into production from sandbox is different from the update process for other bundles, to meet the specialized requirements of bundles developed in sandbox accounts. See [Selective Update of Sandbox Bundle Objects](#).



Note: The process for updating bundles installed into a sandbox account from a development account is the same as the update process for other bundles. The normal bundle update process updates all bundle objects in the target account, overwriting target account objects with source bundle objects. For bundle updates from a development account to a sandbox account, the Preview Bundle Update page always shows the Update action for all objects, even for objects that have not changed in the source account since the last bundle update.

Preview Bundle Update						
		Cancel		Update Bundle		
Name: Customer Support Bundle Current Version: 1.1 New Version: 1.2						
BUNDLE CONTENTS	EXISTING OBJECT IN CONFLICT	SCRIPT ID	ACTION	PREFERENCE	INTERNAL ID	LOCK ON INSTALL
Custom Fields						
Supervisor	Supervisor Name	custevent1_2	Add and Rename <input checked="" type="checkbox"/>	Show on Existing Custom Forms <input checked="" type="checkbox"/>	custevent1_2	
Frequent Support User?		custevent1_1	Update		custevent1_1	
CRM Fields						
Help Use Occurrence		customlist7_3	Update	Preserve Data <input checked="" type="checkbox"/>	236	
Lists						
Training Level (2)		customrecord_level_2	Update	Preserve Data <input checked="" type="checkbox"/>	235	
Records						
CaseClosureEmail.js			Update		2610	
field_changed.js			Update		2611	
File Cabinet						
Files						
Phone Call Form - Automation			Update		40	
Entry Forms						
User Event						
ClosedCase		customscript11_2	Update	Do Not Update Deployments <input checked="" type="checkbox"/>	238	

- For some objects, a lock icon may display in the Lock on Install column, indicating that target account users cannot modify the object.
- For objects that are included in the currently installed version of the bundle, the Action column displays **Update**.
- For objects that are not included in the currently installed version of the bundle, and do not conflict with existing target account objects, the Action column displays **Add**.
- For objects that are not included in the currently installed version and conflict with existing target account objects, the Action column includes a dropdown menu. See [Choose Actions for Conflicting Objects](#).
- For some objects, you can choose setup preferences in the Preference column. See [Set Bundle Update Preferences](#).

When you have finished reviewing this page, click the Update Bundle button to start the update.



Note: If you are required to accept new terms of service for the bundle, a dialog displays these terms. Click I Agree to proceed with the update.

You can continue working in NetSuite as the bundle updates. To check on the progress of the update, go to the list of installed bundles at Customization > SuiteBundler > Search & Install Bundles > List. See [Reviewing the Installed Bundles List](#).

Choose Actions for Conflicting Objects

On the Preview Bundle Update page, conflicting target account objects are displayed, in red, in the Existing Object in Conflict and Script ID columns. For each conflicting object, choose one of the following from the dropdown list in the Action column:

- **Add and Rename** – NetSuite appends a number to the bundle object's name or script ID, to differentiate it from the preexisting custom object. The bundle object is added to account, and the preexisting object remains intact.
- **Replace Existing Object** – The bundle object keeps the same name or script ID. The bundle object is added to the account and the preexisting custom object no longer exists in the account.

If you choose this option for a conflicting object during bundle installation, the preexisting custom object is PERMANENTLY replaced by the bundle's object. Even if you choose to uninstall the bundle later, changes to the replaced object remain and the original object is not restored.

For each custom record that includes data and each custom list where this option is chosen, a preference dropdown list is displayed so you can choose whether to replace, preserve, or merge data in the target account object to be replaced.

For more information, see [Resolving Conflicting Objects](#).

Set Bundle Update Preferences

On the Preview Bundle Update page, review bundle object preferences set by the bundle author and reset as desired. The following preferences may be available:

- Do Not Update Users or Update Users of Published Dashboards

For each dashboard being updated as part of this update, you can set a preference to indicate whether these updates should be applied to the dashboards of users to whom the dashboard has been published. (Updates are always applied to the dashboard definition.)

The default option is not to apply updates to users, allowing an administrator to publish updates to users' dashboards as desired after the bundle update. See the help topic [Applying Changes to Published Dashboards](#).

- Show or Hide Custom Fields

For each custom field being installed for the first time as part of this update, you can set a preference indicating whether the field should be shown or hidden on existing custom forms in target accounts. The setting you chose during the most recent installation or update of this bundle is displayed by default.

Note that this setting does not affect bundled forms, because all custom fields in a bundle are always shown on all custom forms included in that bundle.

- Update or Do Not Update Script Deployments

For each SuiteScript to be updated, you can choose whether to update script deployments in the target account. The setting you chose during the most recent installation or update of this bundle is displayed by default.

If the update deployments option is applied, bundled script deployments completely overwrite target account script deployments, **except for script deployment audience definitions**. Bundle updates

do not impact script deployment audience definitions in target accounts. These audiences are not changed, even if the Update Deployments option is applied.

Note that changes to an icon associated with a dashboard SuiteApp portlet are always copied during a bundle update. This overwrite of the icon in the target account occurs even if the bundle preference is set to prevent the update of script deployments. See [Giving Users Dashboard Access to a SuiteApp](#).

- Replace Data, Preserve Data, or Merge Data for Custom Records and Custom Lists

This preference is available for each custom record with data and each custom list that conflict with existing target account objects and have Replace Existing Object chosen, as well as for each custom record with data and each custom list that need to be updated as part of the bundle update.

You can choose one of the following: Replace Data, Preserve Data, or Merge Data. (The setting you chose during the most recent update of this bundle is displayed by default.)

- The Replace Data option overwrites existing target account object data with bundle object data. This process compares script IDs for records in the bundle and those in the account, to do the following:
 - Delete records from the target account that no longer exist in the bundle's source account.
 - Update records in the target account that also exist in the source account so that values match.
 - Add records to the target account that are new to the source account since the bundle was previously updated in the target account.
- The Preserve Data option preserves existing data in the target account object.
- The Merge Data combines bundled object data with target account object data. As part of this merge:
 - Any records that were added to the object in the target account after bundle installation are preserved.
 - Any records that were added to the object in the bundle source account, after installation in the target account, are added to the target account object.
 - Any records in the target account object that were added as part of the bundle installation and remain in the bundled account object are handled as follows:
 - If the values are the same in both objects, target account object values are preserved as is.
 - If the values are different, target account object values are updated to match bundled account object values.

NetSuite internal IDs are used for this comparison and update of values.

Merge Data Example

For example, consider a custom list called Colors that has been installed as part of a bundle in a target account. At the point in the time when a bundle update is to occur, this list has different data in the target account and the bundle:

- The values in the existing target account Colors list are: red, orange, yellow, green, blue, and purple (installed by the bundle), and white (added in the target account after bundle installation).
- The values in the bundled Colors list are: red, orange, yellow, green, blue, indigo, violet (changed from previous value of purple), black, and gray.

Results of updating the bundle would vary according to the option chosen for the custom list data preference, as shown in the following table:

Option Chosen	Target Account Colors List Data After Bundle Update
Preserve Data	red, orange, yellow, green, blue, purple white

Option Chosen	Target Account Colors List Data After Bundle Update
(Do not change target account object data.)	
Replace Data (Overwrite target account object data with bundled object data.)	red, orange, yellow, green, blue, indigo, violet, black, gray
Merge Data (Add bundled object data to target account object data, eliminating any duplicates.)	<p>red, orange, yellow, green, blue, indigo, violet, white, black, gray</p> <p>Note merge handling of data values:</p> <ul style="list-style-type: none"> ■ red, orange, yellow, green, blue remain as is ■ purple is updated to violet ■ white is preserved ■ indigo, black, and gray are added

i Note: Bundle updates do not change the values of script deployment parameters that have a user or company preference set, even if the bundle preference for the script is set to Update Deployments. For more information, see the help topic [Script Parameter Preferences and Bundles](#).

Bundle Update Reference

In general, updating a bundle causes unlocked bundle objects in the target account to be updated to match source account bundle objects. If you change any of the unlocked elements of an installed bundle, a bundle update may overwrite or merge the changes. If a bundle update overwrites target account objects, you must redo the changes after each bundle update.

The following table describes possible changes to target account objects during a bundle update:

Bundle Object	Description
Custom field names	<p>Changes to custom field names are copied into the source account.</p> <p>For example, you install a bundle that includes a custom record with a field labeled Name, which is set to Include Data. You change the name of the Name field to Company. The bundle creator revises the bundle and then adds it to a deployment account.</p> <p>When you install the updated bundle, the field label changes from Company back to Name.</p>
Inactive property on custom fields	<p>You can use SuiteBuilder to set custom fields to Inactive instead of deleting the custom field. However, if a custom field in a bundle is inactive, SuiteBundler may not set the custom field to inactive in the target account when you update the bundle.</p> <p>If the custom field is inactive in a bundle but active in the target account, and you update the bundle, the custom field remains active in the target account after the bundle update.</p> <p>For more information about the Inactive property for custom fields, see the help topic Inactivating a Custom Field.</p>
Custom form preferences	<p>Generally, If a bundle update includes any custom forms that are set as preferred for one or more roles, these preferences overwrite any custom forms previously set as preferred in the target account.</p> <p>However, a custom form set as preferred for a custom record in a bundle only carries over after a bundle update if the custom record did not previously have a preferred</p>

Bundle Object	Description
	form set in the target account. If after bundle installation a preferred form was set for the custom record in the target account, a bundle update does not change this preferred form.
Access Type property on custom records	<p>If the Access Type value for a bundled custom record is edited in a target account, that new value is kept after a bundle update, if the custom record is not locked in the bundle.</p> <p>For example, you install a bundle that contains a custom record with an Access Type of Use Permission List. In the target account, you change the access type for the custom record to No Permission Required. A bundle update does not overwrite the target account value of No Permission Required with the bundle value of Use Permission List.</p> <p>If a custom record is locked in a bundle, a bundle update overwrites the Access Type value in the target account with the value from the bundle.</p>
Permissions for standard roles on custom records, custom transactions, and custom segments	<p>Changes to standard role permissions for custom records, custom transactions, and custom segments in a bundle may not be reflected in a target account after a bundle update.</p> <p>There is no way to determine whether a difference in permissions between the bundle and the target account is a result of a change in the bundle or of a manual change in the target account. To avoid overwriting changes made in the target account, the bundle update does not modify these permissions.</p>
Workflows	<p>The update of a workflow in a target account depends on the difference between the states and transitions in the source and target accounts.</p> <p>For more information, see Workflow Handling on Bundle Update.</p> <p>Workflow instances and history logs from the source account are not copied to target accounts when a bundle is updated.</p> <p>Workflow release status is not updated in the target account when a bundle is updated or copied. You can set your preferred workflow release status in the target account manually.</p>
Audiences	<p>Generally, bundle updates implement an intelligent merge of bundle object audience definitions in source and target accounts.</p> <p>If the Select All box is checked for employees, customers, vendors, or partners in the source object audience, this selection overwrites the target object audience. If the Select All box is not checked for the source object audience, employees, customers, vendors, and partners are not deleted from the target object audience. However, the target object audience may still be overwritten during the merge.</p> <p>For details about how the merge is done, see Object Audience Handling on Bundle Update.</p>
Hierarchical custom record data	<p>SuiteBundler updates the custom record data in the target account depending on the bundle preference that you specify when you update an existing bundle.</p> <p>See Hierarchical Custom Record Data Handling on Bundle Update.</p>
Center category links	<p>If a custom category in the target account came originally from a bundle, a bundle update removes all existing custom record links from that custom category in the target account and adds any custom record links in the updated bundle to that custom category in the target account.</p> <p>If a user in the target account creates a custom category in the target account with links to custom records, a bundle update does not remove these links from the custom category, even if they link to custom records that came initially from the bundle.</p>

Bundle Object	Description
	<p>A bundle update does not remove any links to standard categories from the target account, because it is not possible to determine whether these links originated from the bundle or not.</p> <div style="border: 1px solid orange; padding: 10px;">  Important: Be aware that the handling of bundled links described above may result in unexpected links in the target account after bundle update. For example, a change of links in the source account may result in these changed links being installed in the target accounts when the bundle is updated, whereas existing links in the target account are preserved. In this case, duplicate links may occur in target accounts. </div>
SSP application files	<p>During a bundle update, conflicts between bundle files and target account files in the SSP Application folder are detected. In case of such a conflict, the bundle update deletes the target account file and replaces it with the file from bundle.</p>

Exceptions to Bundle Update Overwrites

The following table describes exceptions to the bundle overwrite rules:

Bundle Object	Exception
Custom field display order	<p>A bundle update does not impact the display order of custom fields in a target account. The display order is left as is after a bundle update. It is not changed to match the display order of custom fields in the source account.</p>
Script parameters	<p>Script parameters defined as user, company, or portlet preferences are not overwritten in the target account unless the type of the preferences changes in the bundle.</p> <p>If the default value of the preference in a script deployment changes in the bundle, the default is carried over without affecting the preferences specified in the target account.</p>
Script deployment audience definitions	<p>Script deployment audience definitions in target accounts are not overwritten by definitions in an updated bundle, even if the Update Deployments option is selected for the update.</p>
Custom lists	<p>Custom lists updated by a bundle author are replicated in the target account. If the developer removed items from the list, then this item will also be removed from your account.</p> <p>To preserve Custom List options in the target, choose the Update Data or Preserve Data preference. See Installed Bundle Updates.</p>
Custom records converted to custom lists	<p>For objects installed from bundles as custom lists and converted in target accounts to custom records, the record is not converted back to a custom list when the bundle is updated, even if the object is locked as a custom list in the source account.</p> <p>Similarly, you cannot convert custom records to custom lists in the NetSuite UI.</p>

Guidelines for Changing Bundle Objects in a Target Account

You should use the following guidelines when editing bundle objects in the target account:

- If you plan to update a bundle with changes from source, do not update bundle objects such as forms, scripts, or custom fields, because updates will overwrite these changes.
- If you want to make changes to a custom object from an installed bundle, create a copy of the object before making the changes and make changes to the copy, so that future updates do not overwrite the changes.

Workflow Handling on Bundle Update

SuiteBundler updates bundled workflows in a target account using the following rules:

- If states or transitions are modified in an unlocked workflow in the source account, NetSuite updates states and transitions previously installed in the target account with the states and transitions from the source account. NetSuite also adds any new states or transitions in the source account to the target account. If a state or transition with the same name exists in the source and target accounts, NetSuite duplicates the state or transition in the target account.

For example, after you install a bundle, you create a new state "state10" in both source and target account and then update the bundle. In the target account workflow, two states named "state10" will appear, one created directly in the target account and the other installed by the bundle update.

- If states or transitions, or any aspect of the workflow objects, are modified in the target account, updating the bundle will revert the changes back to what the source dictates.
- If states or transitions are deleted in a target account, they will be replaced during a bundle update.
- If you delete any state or transition in the source, but use them in the target, you will be unable to remove these states or transitions in the target account if the workflow is running in the target account and is currently referencing that state. Bundle authors should advise their customers to make these states or transitions inactive if they are not meant to be used any longer.

Workflow instances and history logs are not copied to target accounts when a bundle is updated.

Workflow release status is not updated in the target account when a bundle is updated or copied. You can set your preferred workflow release status in the target account manually.



Note: For more information, see the help topic [Bundling a Workflow](#).

Merge Handling for Reinstallation of Bundled Workflow in Target Production Account from Sandbox

The following handling prevents duplication of workflow states in cases where a bundled workflow is installed in a target account, and the same workflow is later installed again in that same account. For example, this case would apply to installation of a bundled workflow from Sandbox to Production where the sandbox account is a direct clone of the target production account.

When the same workflow is reinstalled in the target account and Merge Data is selected during installation:

- Workflow states are merged based on matching script IDs.
- Actions are merged based on matching script IDs and workflow state IDs.
- Transitions are merged based on matching script ID of source and target state and database ID, meaning merge occurs only if the transition was originally created in sandbox as result of a Production refresh. If a transition was manually created in both Sandbox and Production, it is not merged.



Important: In all other cases, such as when a bundle is installed from a production account into another production account, or from a sandbox account to a non-owning production account, states, actions, and transitions are duplicated without regard to script IDs.

Object Audience Handling on Bundle Update

Bundle updates implement an intelligent merge of bundle object audience definitions in source and target accounts. Object types that have audiences include script deployments, reports, saved searches, and CSV import maps:

- Target account audience definitions for custom roles included in the bundle are updated to match the source account.
- Target account audience definitions for custom roles not included in the bundle are preserved unchanged.
- Target account audience definitions for standard roles are merged with source account audience definitions for these roles, so that any definitions from the source account are added.
- Target account audience definitions not related to roles are preserved unchanged.



Note: If the **All Roles** option is checked in the source bundle, be aware that the intelligent merge does not enable this option in the target account when the bundle is updated.

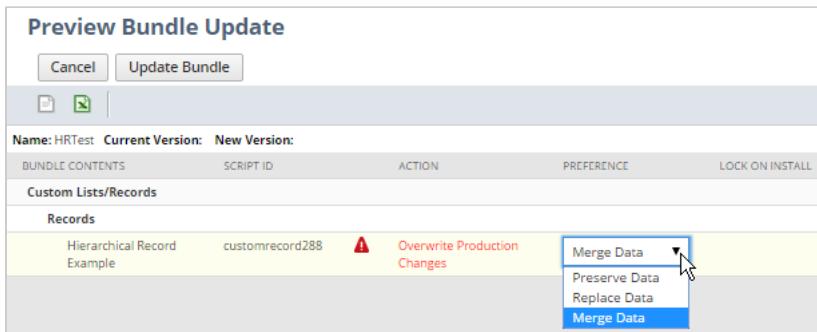
Hierarchical Custom Record Data Handling on Bundle Update

You can create hierarchies between records of the same custom record type in SuiteBuilder. You can then use SuiteBundler to bundle the custom record types and the custom record data. When you bundle the custom record data, SuiteBundler also includes any parent-child hierarchies created between the custom records.

SuiteBundler updates the custom record data in the target account depending on the bundle preference that you specify when you update an existing bundle:

- Merge Data. Combine bundled record data with target account record data. See [Merge Data Examples](#).
- Preserve Data. Keep existing data in the target account. See [Preserve Data Examples](#).
- Replace Data. Overwrite existing target account record data with bundled record data. See [Replace Data Examples](#).

The following screenshot shows the preferences for handling record data when updating an installed bundle in a target account:



For more information, see one of the topics in the following table:

For more information about ...	See ...
Using the Hierarchy option in SuiteBuilder to create hierarchical custom record types.	Creating Custom Record Types
Updating a bundle.	Installed Bundle Updates
Bundle update preferences.	Set Bundle Update Preferences

Merge Data Examples

SuiteBundler uses the following rules when you update a bundle that contains record data with parent-child relationships and use the **Merge Data** option:

- Relationships removed in the source account are also removed in the target account.
- Relationships in the target account that include non-bundle custom records may be overwritten, if they conflict with the parent-child relationships in the source account.
- If you delete a parent record in the source account, SuiteBundler retains the parent record and any relationship to child records in the target account.
- SuiteBundler adds any new records and the associated parent-child relationships, added in the source account, to the target account.
- SuiteBundler retains any new target account records and parent-child relationships if they do not conflict with records and relationships from the source account.



Note: The examples in this section assume that the source bundle has already been installed one time in the target account.

The following table includes bundle update examples for the **Merge Data** update option:

Change Made In Account ...	Description of Changes	Effect on Source Account	Effect on Target Account
Source	<p>Remove parent-child relationship a : b in the source account.</p> <p>After upgrade, relationship is removed in the target account.</p>	Before Edit: After Edit: 	Before update: After update:
Source and target	<p>In the source account, rename parent record, b to p, with a child record.</p> <p>In the target account, edit parent-child relationship, b : c to b : n, where n is a non-bundle record.</p> <p>After upgrade, parent-child relationship b : n is overwritten in the target account.</p>	Before Edit: After Edit: 	After target edit and before update: After update:

Change Made In Account ...	Description of Changes	Effect on Source Account	Effect on Target Account
Source	<p>In the source account, delete parent record e in a parent-child relationship e : f.</p> <p>In the target account, the parent-child relationship is retained, although the parent record and relationship was removed in the source account.</p>	Before Edit: After Edit: 	Before and after update:
Source and target	<p>In the source account, add two new records with the parent-child relationship v : w.</p> <p>In the target account, add two new records with the parent-child relationship y : z.</p> <p>After upgrade, the two new records and the relationship are copied to the target account. The two new records and the relationship remain in the target account.</p>	Before edit: After edit: 	Before Edit: After edit: After update:

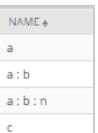
Preserve Data Examples

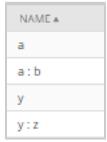
SuiteBundler uses the following rules when you update a bundle that contains record data with parent-child relationships and use the **Preserve Data** option:

- Relationships removed in the source account remain in the target account.

- Relationships in the target account that include non-bundle custom records that conflict with the parent-child relationships in the source account stay in the target account.
- If you delete a parent record in the source account, SuiteBundler retains the parent record and any relationship to child records in the target account.
- SuiteBundler does not add any new records and the associated parent-child relationships from the source account to the target account after the first install.
- SuiteBundler retains any new target account records and parent-child relationships.

The following table includes bundle update examples for the **Preserve Data** update option:

Change Made In Account ...	Description of Changes	Effect on Source Account	Effect on Target Account
Source	<p>Remove parent-child relationship a : b in the source account.</p> <p>After upgrade, the relationship remains in the target account.</p>	Before Edit:  After Edit: 	Before and after update: 
Source and target	<p>In the source account, rename parent record, b to p, with a child record c.</p> <p>In the target account, edit parent-child relationship, b : c to b : n, where n is a non-bundle record.</p> <p>After upgrade, renamed record is not updated and parent-child relationship b : n remains in the target account.</p>	Before Edit:  After Edit: 	After target edit and before update:  After update: 
Source	<p>In the source account, delete parent record e in a parent-child relationship e : f.</p> <p>In the target account, the parent-child relationship is retained, although the parent record and relationship was removed in the source account.</p>	Before Edit:  After Edit: 	Before and after update: 

Change Made In Account ...	Description of Changes	Effect on Source Account	Effect on Target Account
			
Source and target	<p>In the source account, add two new records with the parent-child relationship v : w.</p> <p>In the target account, add two new records with the parent-child relationship y : z.</p> <p>After upgrade, the two new records in the target account and the relationship remains but the two new records and the relationship are not copied to the target account.</p>	Before edit:  After edit: 	Before Edit:  After edit and bundle update: 

Replace Data Examples

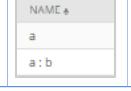
SuiteBundler uses the following rules when you update a bundle that contains record data with parent-child relationships and use the **Replace Data** option:

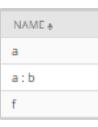
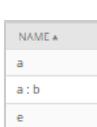
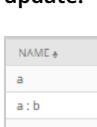
- Relationships removed in the source account are also removed in the target account.
- Relationships in the target account that include non-bundle custom records are overwritten, and the non-bundle records are removed.
- If you delete a parent record in the source account, SuiteBundler also removes the parent record and any relationship to child records in the target account.
- SuiteBundler adds any new records and the associated parent-child relationships added in the source account to the target account.
- SuiteBundler removes any new target account records and parent-child relationships for the custom record type.



Note: The examples in this section assume that the source bundle has already been installed one time in the target account.

The following table includes bundle update examples for the **Replace Data** update option:

Change Made In Account ...	Description of Changes	Effect on Source Account	Effect on Target Account
Source	<p>Remove parent-child relationship a : b in the source account.</p> <p>After upgrade, relationship is removed in the target account.</p>	Before Edit: 	Before update: 

Change Made In Account ...	Description of Changes	Effect on Source Account	Effect on Target Account
		After Edit: 	After update: 
Source and target	<p>In the source account, rename parent record, b to p, with a child record.</p> <p>In the target account, edit parent-child relationship, b : c to b : n, where n is a non-bundle record.</p> <p>After upgrade, parent-child relationship b : n is overwritten in the target account and non-bundle node n is removed.</p>	Before Edit:  After Edit: 	After target edit and before update:  After update: 
Source	<p>In the source account, delete parent record e in a parent-child relationship e : f.</p> <p>In the target account, the parent-child relationship is removed, and the parent record is removed.</p>	Before Edit:  After Edit: 	Before update:  After update: 
Source and target	<p>In the source account, add two new records with the parent-child relationship v : w.</p> <p>In the target account, add two new records with the parent-child relationship y : z.</p> <p>After upgrade, the two new records and the relationship are copied to the target account. The two new records and the relationship in the target account are removed.</p>	Before edit:  After edit:	Before Edit:  After edit:

Change Made In Account ...	Description of Changes	Effect on Source Account	Effect on Target Account															
		<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>NAME ↴</td></tr> <tr><td>a</td></tr> <tr><td>a : b</td></tr> <tr><td>v</td></tr> <tr><td>v : w</td></tr> </table>	NAME ↴	a	a : b	v	v : w	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>NAME ↴</td></tr> <tr><td>a</td></tr> <tr><td>a : b</td></tr> <tr><td>y</td></tr> <tr><td>y : z</td></tr> </table> <p>After update:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>NAME ↴</td></tr> <tr><td>a</td></tr> <tr><td>a : b</td></tr> <tr><td>v</td></tr> <tr><td>v : w</td></tr> </table>	NAME ↴	a	a : b	y	y : z	NAME ↴	a	a : b	v	v : w
NAME ↴																		
a																		
a : b																		
v																		
v : w																		
NAME ↴																		
a																		
a : b																		
y																		
y : z																		
NAME ↴																		
a																		
a : b																		
v																		
v : w																		

System Notes Behavior for Bundle Updates of Deleted Custom Records or Custom Record Entries

For bundle updates of bundled custom records with deleted custom record entries in a source account, there are different scenarios about what happens to system notes for changes on the custom record entries in a target account.



Note: System notes get deleted for entries that are deleted from a bundle.

The following table describes the different bundle update scenarios and their effect on system notes for changes on the custom record entries:

	Description of Changes	Effect on System Notes for the Preserve Data Option	Effect on System Notes for the Merge Data Option	Effect on System Notes for the Replace Data Option
Scenario 1	<p>Delete the custom record entries from a bundled custom record in a source account.</p> <p>Update the bundle in a target account.</p>	System notes for changes on the custom record entries are retained in a target account.	The custom record entries and system notes for changes on the custom record entries that got deleted from a source account, get deleted from a target account.	When custom record entries get deleted from a source account, system notes for changes on the custom record entries get deleted from a target account as well.
Scenario 2	<p>For custom records, that are included in multiple bundles:</p> <p>Delete a bundled custom record in a source account.</p> <p>Create a new custom record with the same ID as the previously deleted custom record.</p> <p>Edit the bundle in a source account by</p>	System notes for changes on the custom record entries are retained in a target account.	The custom record entries and system notes for changes on the custom record entries that got deleted from a source account, get deleted from a target account.	When custom record entries get deleted from a source account, system notes for changes on the custom record entries get deleted from a target account as well.

	Description of Changes	Effect on System Notes for the Preserve Data Option	Effect on System Notes for the Merge Data Option	Effect on System Notes for the Replace Data Option
	<p>adding the newly created custom record.</p> <p>Update the bundle in a target account.</p>			
Scenario 3	<p>For custom records, that are included in one bundle:</p> <p>Delete a bundled custom record in a source account.</p> <p>Create a new custom record with the same ID as the previously deleted custom record.</p> <p>Edit the bundle in a source account by adding the newly created custom record.</p> <p>Update the bundle in a target account.</p>	<p>System notes for changes on the custom record entries get deleted from a target account.</p>	<p>System notes for changes on the custom record entries get deleted from a target account.</p>	<p>System notes for changes on the custom record entries get deleted from a target account.</p>

Using Managed Bundles

Managed bundles support automated updates by solution providers. Because the solution provider is responsible for the bundle update process, you do not have to take any action to ensure that a managed bundle installed in your production account is always the most up-to-date version.

The list of installed bundles, at Customization > SuiteBundler > Search & Install Bundles > List, has a Managed column that indicates which bundles are managed.

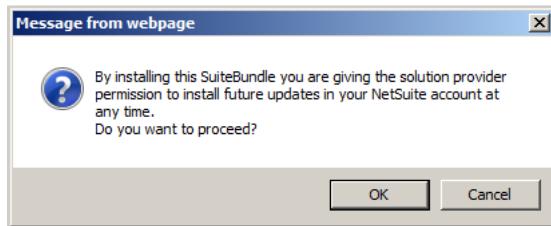
Installed Bundles					
ACTION	NAME	BUNDLE ID	VERSION	MANAGED	ABS
	Customer Feedback	1052	2.1	No	Includ cus rec sav set new for cus fee
	Customer Survey	2792	1.0	No	
	Equip Maintenance	3396		No	

You cannot manually update a managed bundle installed in a production account by selecting the Update option in this page's Action menu. Managed bundle updates are pushed into your account at the discretion of the solution provider.

In addition, you cannot override the managed bundle object preferences set by the solution provider. If a non-managed bundle installed in your account later becomes managed, all preferences previously selected in your account are ignored during bundle updates. See [Set Bundle Installation Preferences](#) and [Set Bundle Update Preferences](#).

Installing a Managed Bundle

You can install a managed bundle in the same way that you install any other bundle in your account, but before you proceed with installation, you are presented with a popup message. This message informs you that by installing the bundle, you are giving the solution provider permission to install future updates at any time.



Updating a Bundle that was Converted to Managed

In some cases, a bundle may not be a managed bundle when you first install it, but later is converted to be a managed bundle. In this case, the next time you click **Update**, you are informed that the bundle is now a managed bundle, and you must agree to let the solution provider manage future bundle updates before you can update the bundle. If you no longer want the managed bundle, you can uninstall it and receive no further upgrades.

User for Managed Bundle Update Audit Trail

The user shown in the bundle audit trail as executing a managed bundle update in your account is an arbitrary designation. The managed bundle update process selects a user in the employee list to be the entry in the **Installed By** column of the audit trail. This selection is the administrator that has the lowest internal ID.

Managed Bundle Upgrade Notifications to Target Account Users

SuiteBundler sends email notifications of managed bundle upgrades to target account users. Email recipients can be designated on the Bundle Details page.

When a managed bundle is successfully upgraded in your account, an email message containing the bundle name, bundle ID, bundle version, and account ID is sent. No message is sent when an upgrade fails.

The recipients for this email message are based on the **Email Recipients** setting on the Messages subtab of the Bundle Details page in your account. The default choice is **All Bundle Admins**, indicating that all administrators and users with the SuiteApp Marketplace permission in the account are recipients. A choice of **None** opts out of receiving this email. A choice of **Custom** allows the entry of specific users to receive the email.



Note: The **Email Recipients** setting applies to all bundle messages, not only managed bundle upgrade notifications. For more information, see [Subscribing to Bundle Email Messages](#).

Uninstalling a Bundle

You can uninstall a customization bundle at any time after installing it.

You cannot uninstall a configuration bundle. For more information, see [Configuration Bundles](#).

Uninstalling a managed bundle ends all upgrades to that bundle in your account.



Warning: Uninstalling a bundle may result in data loss. When you uninstall a customization bundle, all objects included in that bundle, including any data in these objects, are deleted. Additionally, any custom records you created from the custom record types in the bundle are deleted. To avoid potential data loss, do not uninstall the bundle.

To uninstall a bundle:

1. Log in to the account in which you installed the bundle.
2. Go to Customization > SuiteBundler > Search & Install Bundles > List.
3. Choose **Uninstall** from the **Action** dropdown list for the bundle you want to uninstall.
4. In the confirmation popup, click **OK**.

NetSuite deletes the bundle, and the bundle's customization objects, from your account. Be aware of the following objects that are exceptions: objects referenced by other bundles installed in the account, and objects that were set to replace an existing object during bundle installation are not deleted.



Warning: If when you installed this bundle, you set objects in the bundle to **Replace Existing Object**, then these objects cannot be uninstalled with this procedure. After uninstalling the bundle, the objects that you replaced remain in your account in the same configuration as they were in after installation.

Note about Uninstalling Bundled Custom Records Used for Workflows

The following changes are made to workflows related to a custom record type installed as part of a bundle when that bundle is uninstalled:

- The bundle uninstall deletes any workflows that have been deployed to that record type.
- The bundle uninstall deletes any actions that point to that record type but are deployed to different record type. (Note that in these cases, the actions are deleted, but not the workflows.)

After a bundle uninstall, you should review your workflows for possible changes.

Viewing the Bundle Audit Trail

NetSuite supports tracking of all bundle installations, updates, and bundle uninstallations in your account. These actions are recorded on the read-only Bundle Audit Trail page, available at Customization > SuiteBundler > Bundle Audit Trail.

Audit trail results are always sorted in reverse chronological order, starting with the most recent action. You can filter the results by Action, Bundle ID, User, and time interval and you can export these results to a CSV file.

As of 2017.1, the SuiteBundler Audit Trail permission provides access to the Bundle Audit Trail page, but not to other SuiteBundler tasks. Prior to this release, the SuiteApp Marketplace permission was

required to see the Bundle Audit Trail page. The SuiteBundler Audit Trail permission is useful for auditors. This permission includes only one level, Full. This level is required to view the Bundle Audit Trail page. Users without this permission cannot access this page. By default, all account administrators have this permission.

Bundle Audit Trail											
<input type="button" value="Refresh"/> <input type="checkbox"/> FILTERS											
Action	Bundle ID	User	VERSION	ACTION	START DATE	END DATE	STATUS	USER	SOURCE	NOTES	ERROR REFERENCE
XXXXXX	International Tax Reports		3.54.0	Managed Upgrade	06/24/2014 8:36 am	06/24/2014 8:37 am	Success	Wolfe, S	XXXXXXX		
XXXXXX	Tax Audit Files		1.5.0	Managed Upgrade	06/24/2014 5:22 am	06/24/2014 5:30 am	Failure	QA Test	XXXXXXX	hs1zt4z1diy0n93wq7k8	
XXXXXX	Payment Gateway		1.24.0	Managed Upgrade	10/23/2013 4:16 am	10/23/2013 4:16 am	Success	Wolfe, S	XXXXXXX		
XXXXXX	International Tax Reports		3.53.0	Managed Upgrade	10/21/2013 6:25 am	10/21/2013 6:26 am	Success	QA Test	XXXXXXX		

The following table describes the bundle audit trail columns:

Column Name	Description
Bundle ID	Unique system-defined identifier for the bundle.
Name	Name for bundle. Defined by the bundle author.
Version	Version number for bundle. Defined by the bundle author.
Action	Action performed by the bundler. Includes the following actions: Install, Update, Uninstall, or Managed Upgrade for managed bundles.
Start Date	Date and time the bundle action began. i Note: The date and time value is hardcoded to Pacific Time Zone and does not reflect the time zone setting on Home > Set Preferences.
End Date	Date and time the bundle action ended. i Note: The date and time value is hardcoded to Pacific Time Zone and does not reflect the time zone setting on Home > Set Preferences.
Status	Success or failure of the bundle action.
User	Name of the user who performed the bundle action.
Source	Location from which bundle was installed or updated.
Notes	Additional bundle details. For example, a bundle that replaces a deprecated bundle displays "Deprecated bundle <bundle_ID>."
Error Reference	The internal error database number for any error that occurs during bundle installation, update, or uninstallation. You can provide this number to NetSuite Customer Support when troubleshooting a failed bundle installation, update, or uninstallation.

Bundle-Specific Audit Trail

The **Audit Trail** subtab of each Bundle Details page tracks installations, updates, and uninstalls of that specific bundle. This information can be filtered by **Action**.

Bundle Details

Advanced Promotions

[Back](#)

Name	Advanced Promotions	No. Installs	1,017
ID	49247	Product	NetSuite OneWorld
Copied From		Vertical	
Version	2.04.0	Languages	English (U.S.)
Company	XXXXXXX	Availability	Public
<input checked="" type="checkbox"/> Managed		Documentation	helpcenter.nl?topic=DOC_AdvancedPromotions_SuiteApp
Available Since	2/12/2014		

[Overview](#) [Components](#) [SuiteApp Info](#) [Messages](#) [Audit Trail](#) [☰](#)

Action	NAME	VERSION	ACTION	START DATE	END DATE	STATUS	USER	NOTES
- All -	Advanced Promotions	2.04.0	Managed Upgrade	7/22/2014 5:47 am	7/22/2014 5:48 am	Success	admin	
	Advanced Promotions	2.03.0	Managed Upgrade	5/20/2014 12:17 pm	5/20/2014 12:18 pm	Success	admin	
	Advanced Promotions	2.02.1	Managed Upgrade	3/20/2014 3:53 am	3/20/2014 3:53 am	Success	admin	
	Advanced Promotions	2.02.0	Managed Upgrade	2/12/2014 5:47 am	2/12/2014 5:48 am	Success	admin	Deprecated bundle 31,356

i Note: The Bundle Audit Trail is available to account administrators and to other users with the SuiteApp Marketplace permission.

Identifying Bundle Objects in Target Accounts

To allow you to identify the objects in your account that came from bundles, list pages containing these objects include a **From Bundle** column. For objects that were added through a bundle installation, update, or copy, this column displays the bundle ID. For other objects, this column is blank.

Record Types		
New Type		
FILTERS		
<input type="checkbox"/> Show Inactives		
EDIT ▾	FROM BUNDLE	ID
AP Eligible Customer	49247	customrecord
AP Eligible Order	49247	customrecord
AP Minimum Purchase	49247	customrecord
AP Promotion Discount	49247	customrecord
AP Promotion Shipping Method	49247	customrecord
AP Promotion Shipping Price	49247	customrecord
AP Promotion Tier	49247	customrecord

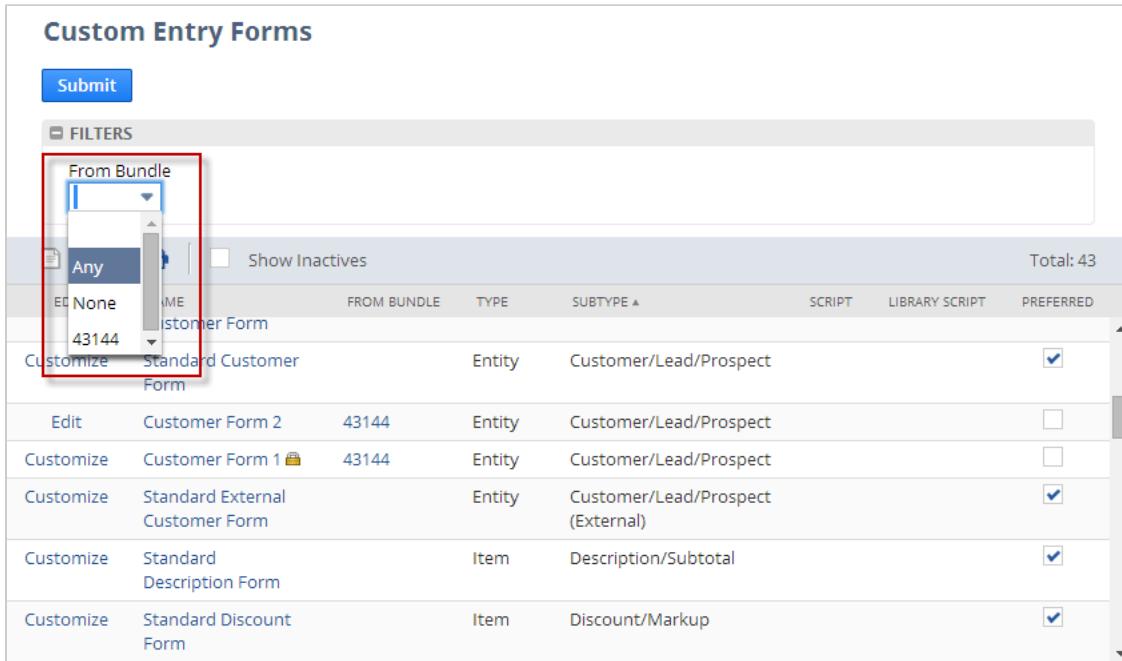
- Each value in the **From Bundle** column is a link that you can click to go to the Bundle Details page. See [Bundle Details](#).

- You can use the **From Bundle** footer filter to alter the bundle objects that are displayed in a list page. See [Filtering Bundle Objects on List Pages](#).
- Objects from a bundle do not display in list pages if the **Hide Components** setting is enabled for the bundle. See [Choosing to Hide or Show Bundle Components](#).
- In cases where an object came from more than one bundle, this column may display multiple bundle IDs. The bundle most recently updated in the account is listed first, and the link goes to the Bundle Details page for that bundle.
- In cases where an object came from a bundle that was copied rather than installed into the account, this column does not display the bundle ID for the source bundle. If you click the link to go to the Bundle Details page, note that this page is for the copied bundle, so it has a different ID than the source bundle listed in the **From Bundle** column.

Filtering Bundle Objects on List Pages

Any NetSuite list page for a bundleable object type may display objects that have been installed by a bundle. You can use the **From Bundle** dropdown list in the **Filter** area to filter the objects from bundles that are displayed on the page. This filtering capability can make object lists more manageable, allowing you to shorten lists and find specific objects.

Whether a bundle's objects are displayed in list pages depends on the hide or show setting for bundle components. This setting is defaulted according to the option chosen by the bundle author. In your account, this setting can be changed on the Bundle Details page. See [Choosing to Hide or Show Bundle Components](#).



The screenshot shows a list of custom entry forms. At the top, there is a 'Submit' button and a 'FILTERS' section. The 'From Bundle' dropdown menu is open, showing three options: 'Any', 'None', and '43144'. A red box highlights this dropdown menu. Below the filters, there is a table with columns: Edit, Name, FROM BUNDLE, TYPE, SUBTYPE, SCRIPT, LIBRARY SCRIPT, and PREFERRED. The table contains several rows of data, with the last few rows having checked boxes in the PREFERRED column.

Edit	Name	FROM BUNDLE	TYPE	SUBTYPE	SCRIPT	LIBRARY SCRIPT	PREFERRED
Customer Form			Entity	Customer/Lead/Prospect	<input checked="" type="checkbox"/>		
Customer Form 2	43144		Entity	Customer/Lead/Prospect	<input type="checkbox"/>		
Customer Form 1	43144		Entity	Customer/Lead/Prospect	<input type="checkbox"/>		
Standard External Customer Form			Entity	Customer/Lead/Prospect (External)	<input checked="" type="checkbox"/>		
Standard Description Form			Item	Description/Subtotal	<input checked="" type="checkbox"/>		
Standard Discount Form			Item	Discount/Markup	<input checked="" type="checkbox"/>		

The **From Bundle** filter includes the following options:

- Any** - Select this option to display any objects that have been installed by bundles. Objects created within the account are not displayed.
- None** - Select this option to display only objects created within the account. Objects installed by bundles are not displayed.

- **<Bundle ID>** - ID for each bundle that has been installed in the account - Select a bundle ID to display only objects that have been installed by that specific bundle.

The **From Bundle** filter is available on list pages for all object types that have at least one object installed by a bundle, where that bundle is not marked to hide components. For example, if an account contains custom lists, custom records, and saved searches installed by bundles, and none of these bundles are marked to hide components, the **From Bundle** filter is available on Custom List, Custom Record, and Saved Search list pages. However, if an account contains custom roles installed by a bundle, but that bundle is marked to hide components, these roles are not displayed in the roles list page and the **From Bundle** filter is not available on this page.

SuiteApp Development Process with SuiteBundler



Note: SuiteBundler is still supported, but it will not be updated with any new features.

To take advantage of new features for packaging and distributing customizations, you can use the Copy to Account and SuiteCloud Development (SDF) features instead of SuiteBundler.

Copy to Account is an administrator tool that you can use to copy custom objects between your accounts. The tool can copy one custom object at a time, including dependencies and data. For more information, see the help topic [Copy to Account Overview](#).

SuiteCloud Development Framework is a development framework that you can use to create SuiteApps from an integrated development environment (IDE) on your local computer. For more information, see the help topic [SuiteCloud Development Framework Overview](#).

The SuiteBundler feature offers flexibility in how you distribute bundles to other NetSuite accounts. Whether you are an internal developer creating customizations for your company, an independent software developer (ISV) distributing solutions to your customers, or an administrator who wants to make a bundle available to other users at your company, NetSuite provides different methods for you to use.

At the core of this SuiteApp development process is the development account. A NetSuite development account is an account, isolated from production, in which you can develop and test new applications and customizations without worrying about affecting your production account. Each development account has the same features and modules as a production account, but it does not contain production data. For more information about development accounts, see the help topic [NetSuite Development Accounts](#).

You can use separate development accounts for developing, testing, and releasing customizations. Use SuiteBundler in conjunction with the customizations created in a development account to package the SuiteApp as a bundle for distribution.

Support for bundle operations varies across the different types of NetSuite accounts, including development accounts. See [Bundle Support Across Account Types](#).



Important: This section contains best practices for managing SuiteApp development with development accounts in conjunction with sandbox accounts. You should adhere to these best practices. You can, however, adapt the best practices to your own specific requirements.

You can use SuiteCloud Development Framework (SDF) to develop your SuiteApp. SDF includes support for SuiteApp projects, self-contained, standalone projects that enable SuiteCloud Developer Network (SDN) members to develop and deploy SuiteApps to their NetSuite accounts. SuiteBundler is used to bundle and share SDF SuiteApps. You can create a bundle that contains all objects from an SDF SuiteApp project without manually adding all of the objects to the bundle in the Bundle Builder. For information about using SDF, see the help topic [SuiteCloud Development Framework Overview](#).

The following table lists the topics where you can get more information about these processes:

Topic	Description
SuiteApp Development Terminology	Terminology used to describe the bundle development process.
Which Method Should I Choose?	Description of the different methods used to develop SuiteApps with development accounts. Includes information to decide on the best method for your requirements.
Additional SuiteApp Development Guidelines	Guidelines to use during SuiteApp development.
Single Development Account Method	How to use a single development account to develop SuiteApps.
Multiple Development Account Methods	<p>How to use multiple development accounts in a SuiteApp development environment. The better approach depends on how many simultaneous versions of an individual SuiteApp that you want to develop:</p> <ul style="list-style-type: none"> <li data-bbox="545 671 1073 699">■ Single Development Account Environment Method <li data-bbox="545 705 1099 732">■ Multiple Development Account Environments Method

SuiteApp Development Terminology

This section uses the following terminology when describing the SuiteApp development process:

Term	Description
development account	Account in which you can develop and test new applications and customizations. For a more detailed description of development accounts, see the help topic NetSuite Development Accounts .
development account environment	Set of development accounts used as an environment in which to develop SuiteApps. Each development account is used for a specific purpose in the SuiteApp development process.
target account	NetSuite account into which an administrator installs a bundle.
source account	NetSuite development account from which a bundle is installed into another account.
sandbox account	<p>NetSuite can provision one or more sandbox accounts for use with a production account. For more information, see the help topic NetSuite Sandbox.</p> <p>NetSuite account used for the integration and user acceptance testing of customizations created in a development account environment before installation in a production account.</p>
release account	NetSuite development account dedicated to the release of bundles. A release account can serve as the source account of bundles that customers install into their sandbox or production accounts.
test account	NetSuite development account dedicated to the testing of bundles. Use a testing account to perform QA on NetSuite customizations that were developed in a development account.
develop account	NetSuite development account dedicated to the development of customizations, independent of production customizations or production data.

Which Method Should I Choose?

The following table lists the different development methods and the relative benefits:

Method	Description	Benefits	Drawbacks
Single Development Account Method	<p>Uses a single development account.</p> <p>The single development account is used for developing, testing, and releasing a SuiteApp.</p>	<ul style="list-style-type: none"> ■ Size. Appropriate for smaller customization projects. ■ Release and update management. Release or update production-ready bundles. 	<ul style="list-style-type: none"> ■ Bug fixes. Difficult to release bug fixes for older versions of the bundle. ■ Future releases. If customers do not upgrade in a timely manner to the new version of a bundle, then the capability to jump versions during a future upgrade becomes problematic. For example, it may not be possible to upgrade from version 1.0 of a bundle to version 3.5 of the same bundle. ■ No difference between development and release. No de-coupling of development and release account types. Changes to objects in a bundle in the development account are included in subsequent installations or updates by other accounts. ■ Consistency. No consistent copy of current release version of the bundle. ■ Changed objects affect bundle components. Any change to an object in a bundle affects the behavior of the bundle. So, a change to a single object in a bundle could be installed as an update. ■ No team development. Multiple developers need to work in the same environment.
Single Development Account Environment Method	<p>Uses three development accounts to develop one version of a bundle at a time.</p> <p>A bundle developer works in one account, stages the bundle for release in a second account, and tests in a third account.</p>	<ul style="list-style-type: none"> ■ Separate accounts. Decouples development, test, and release accounts. ■ Avoids bundle overwrites. Inadvertent overwriting of bundle content more unlikely without using sandbox for testing and release. ■ Testing with production data. Uses sandbox account for user acceptance and integration testing with production data. 	<ul style="list-style-type: none"> ■ Bug fixes. Difficult to release bug fixes for older versions of the bundle. ■ No multiple version development. Supports development of only one version of a SuiteApp at a time. ■ Team development. Multiple developers working on a single version of a SuiteApp would be accessing the same account. ■ SuiteApp versioning. Can be difficult to manage SuiteApp versioning between development cycles.
Multiple Development Account	Uses two environments, where each environment contains three development accounts.	<ul style="list-style-type: none"> ■ Bug fixes. Can fix bugs in one environment and at the same time develop 	<ul style="list-style-type: none"> ■ Team development. Multiple developers working on a single version of a SuiteApp would be accessing the same account.

Method	Description	Benefits	Drawbacks
Environments Method	<p>Used to simultaneously develop multiple versions of a single SuiteApp.</p> <p>In each environment, a bundle developer works in one account, stages the bundle for release in a second account, and tests in a third account.</p>	<p>new version in a separate environment.</p> <ul style="list-style-type: none"> ■ Separate code bases. Bundle being developed separate from release version. ■ Separate accounts. Decouples development, test, and release accounts. ■ Avoids bundle overwrites. Inadvertent overwriting of bundle content more difficult without using sandbox for testing and release. ■ Testing with production data. Uses sandbox account for user acceptance and integration testing with production data. 	<ul style="list-style-type: none"> ■ SuiteApp versioning. Can be difficult to manage SuiteApp versioning between development cycles.

Additional SuiteApp Development Guidelines



Important: Use the following guidelines when developing SuiteApps with the SuiteBundler development methods for development accounts.

- NetSuite follows a phased release schedule for each new version. The phasing process from initial release until the point where all customers are using the new release occurs over several weeks. During the phasing periods, developers, particularly ISVs, must coordinate the installation and update of bundles to ensure that bundles requiring new features are installed into upgraded accounts.
For more information, see [Bundle Support during NetSuite Release Phasing](#).
- You can use managed bundles to streamline the process of updating and supporting your bundles. For details, see [Managed Bundles Overview](#). ISVs must join the SuiteCloud Developer Network to obtain the appropriate development and testing accounts. See the [SuiteCloud Developer Network](#) page on the NetSuite website.
- The development methods described in this section are not intended for bundles developed in sandbox accounts. For information about developing bundles in sandbox accounts, see [Sandbox Bundle Deployment Models](#).
- If you use SuiteScript 2.0 from a development account, make sure that you review and set an appropriate module scope for scripts using a JSDoc tag. To successfully import a SuiteScript 2.0 script created in a development account into another account (such as your NetSuite production or sandbox account), the script must include the following JSDoc Tag:

```

1 /**
2  * @NModuleScope Public

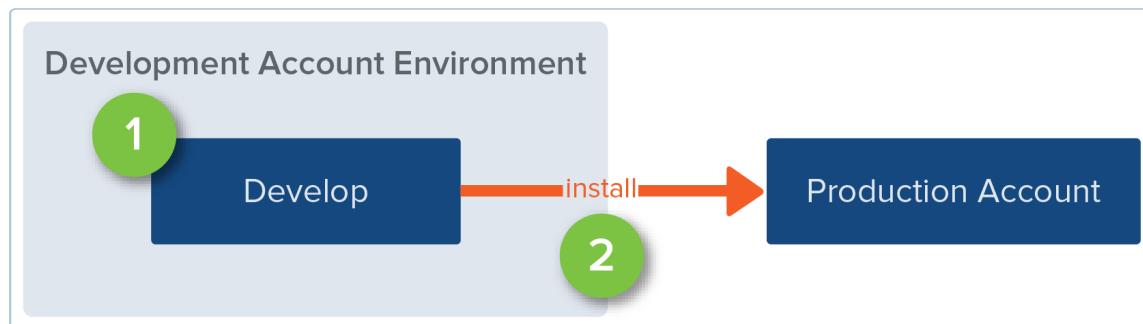
```

For more information, see the help topics [SuiteScript 2.x JSDoc Validation](#) and [NetSuite Development Accounts and Module Scope](#).

Single Development Account Method

You can use an environment with a single development account to develop one version of a bundle at a time. A single development account is used for developing, testing, and releasing a SuiteApp.

The following diagram shows the process flow for the bundle development:



Process

The following table describes the basic steps for developing a SuiteApp using the single development account method:

Step Number	Description
1	<p>Develop the customization in the develop account. Use SuiteBuilder and SuiteScript to develop the customization. For more information, see the help topic Customize Your NetSuite Account.</p> <p>Then, in the develop account, use the Bundle Builder to create a bundle with the customization. See Creating a Bundle with the Bundle Builder.</p>
2	<p>Install the bundle into the production account.</p> <p>New versions of directly installed bundles can be propagated to other accounts by using one of the following methods:</p> <ul style="list-style-type: none"> ■ Notifying administrators, who have installed your bundle, to manually update it from the Installed Bundles page. For more information, see Installed Bundle Updates. ■ Pushing updated bundles to the target accounts. This method requires that you have administrator access to the target accounts. For more information, see Pushing a Bundle to Other Accounts.

Single Development Account — Benefits and Drawbacks

The following table lists the benefits and drawbacks to this approach:

Benefits	Drawbacks
<ul style="list-style-type: none"> ■ Size. Appropriate for smaller customization projects. 	<ul style="list-style-type: none"> ■ Bug fixes. Difficult to release bug fixes for older versions of the bundle. ■ Future releases. If customers do not upgrade in a timely manner to the new version of a bundle, then the capability to jump versions during

Benefits	Drawbacks
<ul style="list-style-type: none"> ■ Release and update management. Release or update production-ready bundles. 	<ul style="list-style-type: none"> a future upgrade becomes problematic. For example, it may not be possible to upgrade from version 1.0 of a bundle to version 3.5 of the same bundle. ■ No difference between development and release. No decoupling of development and release environments. Changes to objects in a bundle in the development account are included in subsequent installations or updates by other accounts. ■ Consistency. No consistent copy of current release version of the bundle. ■ Changed objects affect bundle components. Any change to an object in a bundle affects the behavior of the bundle. So, a change to a single object in a bundle could be installed as an update. ■ No team development. Multiple developers need to work in the same environment.

Multiple Development Account Methods

You can use the following methods to develop SuiteApps using multiple development accounts. These accounts may all be within a single development account environment, or may be in multiple development account environments.

- [Single Development Account Environment Method](#)
- [Multiple Development Account Environments Method](#)
- [Bug Fixing Methodology](#)

Using Sandbox Accounts

In the above methods, you should use a sandbox account for integration and user acceptance testing. For example, you can use a separate sandbox account to test bundles for the following reasons:

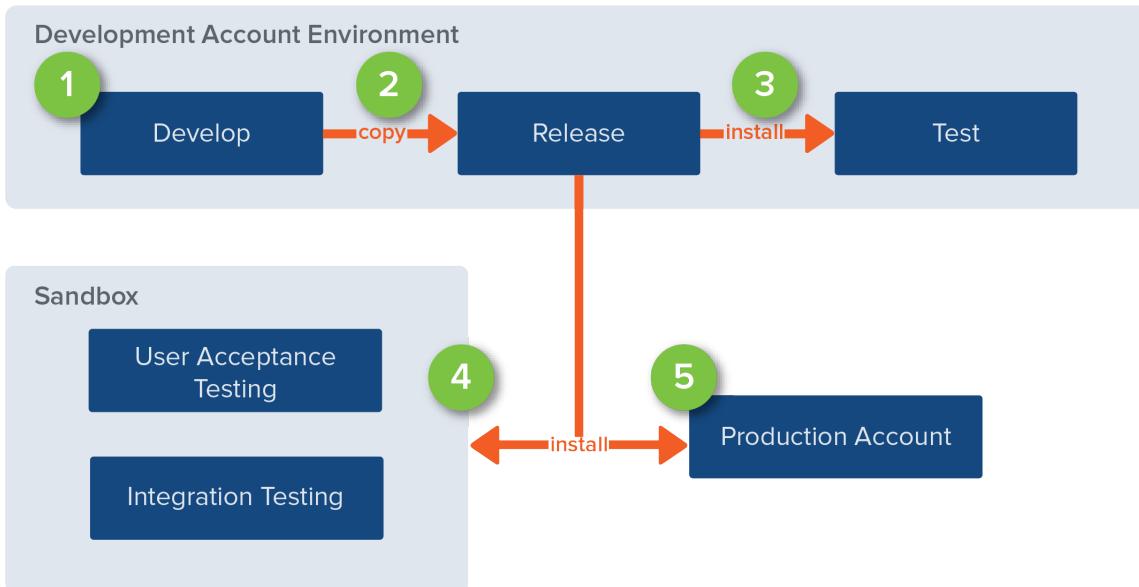
- SuiteApp contains a significant amount of SuiteScript.
- SuiteApp requires specific account setup and configuration.
- SuiteApp includes dependencies among objects, such as dashboards, forms, and custom records.

For information about working with bundles in sandbox accounts, see [SuiteApps and Sandbox Accounts](#).

Single Development Account Environment Method

You can use a single development account environment to develop one version of a bundle at a time. Three development accounts are used for developing, testing, and releasing a single version of a SuiteApp. In this method, a bundle developer works in one account, stages the bundle for release in another account, and tests in a third account.

The following diagram shows the process flow for the bundle development:



Process

The following table describes the basic steps for developing a SuiteApp with a single development account environment:

Step Number	Description
1	<p>Develop the customization in the develop account. Use SuiteBuilder and SuiteScript to develop the customization. For more information, see the help topic Customize Your NetSuite Account.</p> <p>Then, in the develop account, use the Bundle Builder to create a bundle with the customization. See Creating a Bundle with the Bundle Builder.</p>
2	<p>Copy the bundle to the release account. Copy a bundle to a release account to support your release process. See Copying a Bundle to Other Accounts.</p> <p>Note: This account should only contain copied bundles, and not any objects created as part of the customization.</p>
3	<p>Install the bundle from the release account into the test account. You can create sample data to support testing of the customization. See Installing a Bundle.</p> <p>If you want to make changes to the customization during testing, make the changes on the develop account and repeat steps 1 to 3 above.</p>
4	<p>After the bundle has been tested on the test account and all desired changes are incorporated into the bundle and copied to the release account, install the bundle into a sandbox account for additional integration and user acceptance testing with production data.</p> <p>If you want to make changes to the customization after testing in the sandbox, make the changes on the develop account and repeat steps 1 to 4 above.</p>
5	After the bundle has been completely tested in the sandbox account, install the bundle from the release account into the production account.

Step Number	Description
	<p>Note: At this point, you can begin developing a second version of the bundle on the development account, using the objects from the previous version as a starting point. If you want multiple simultaneous versions of the same bundle, see Multiple Development Account Environments Method.</p>

Single Development Account Environment Method — Benefits and Drawbacks

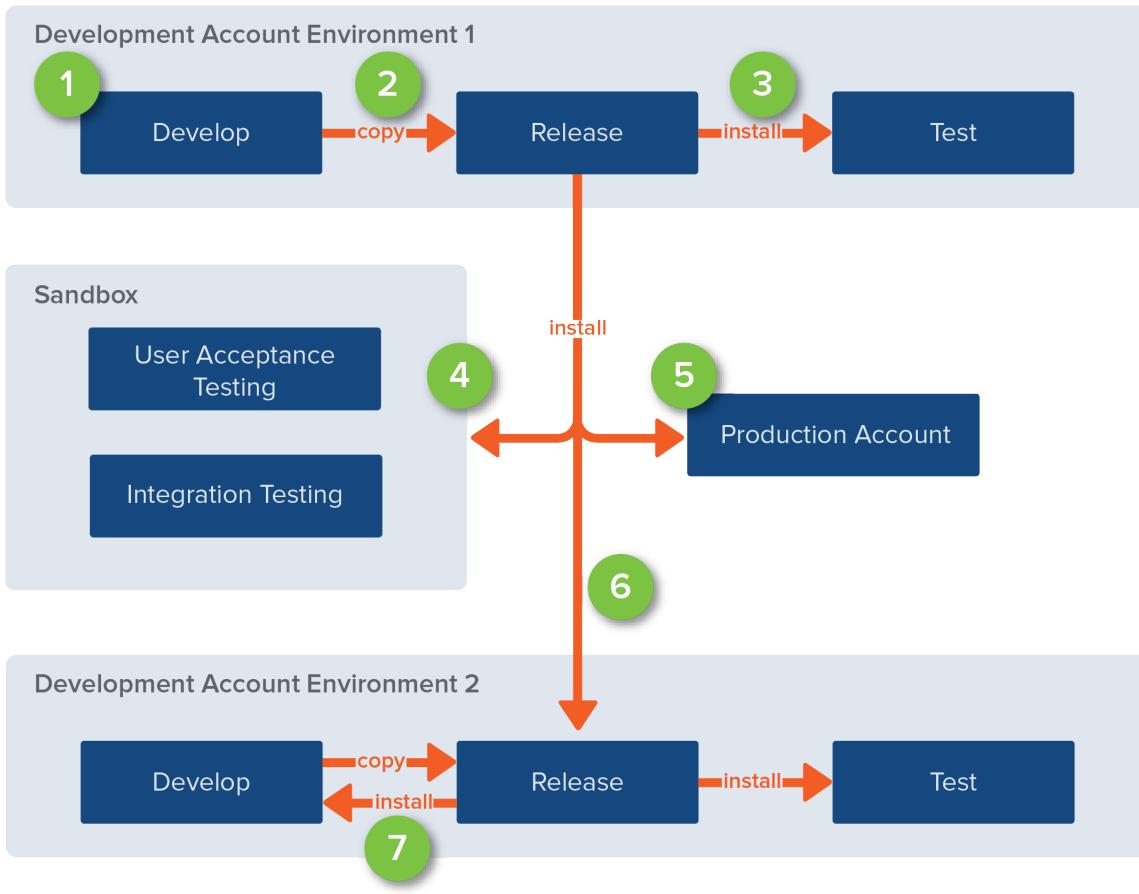
The following table lists the benefits and drawbacks to this approach:

Benefits	Drawbacks
<ul style="list-style-type: none"> ■ Separate accounts. De-couples development, test, and release accounts. ■ Avoids bundle overwrites. Inadvertent overwriting of bundle content does not occur. ■ Testing with production data. Uses sandbox account for user acceptance and integration testing with production data. 	<ul style="list-style-type: none"> ■ Bug fixes. Difficult to release bug fixes for older versions of the bundle. ■ No multiple version development. Supports development of only one version of a SuiteApp at a time. ■ Team development. Multiple developers working on a single version of a SuiteApp would be accessing the same account. ■ SuiteApp versioning. Can be difficult to manage SuiteApp versioning between development cycles.

Multiple Development Account Environments Method

You can use multiple development account environments to develop multiple versions of a SuiteApp. Three development accounts are used for developing, testing, and releasing a single version of a SuiteApp. In this method, a bundle developer works in one account, stages the bundle for release in another account, and tests in a third account. A second environment is used to develop subsequent versions of the SuiteApp.

The following diagram shows the process flow for the bundle development:



Note: The following steps show the basic process for developing, for example, a leading and lagging version of one customization. However, you can use additional development account environments to develop multiple versions of multiple customizations simultaneously.

Process

The following table describes the basic steps for developing a multiple versions of a single SuiteApp with multiple development account environments:

Step Number	Description
1	<p>Develop the customization in the develop account on Development Account Environment 1. Use SuiteBuilder and SuiteScript to develop the customization. For more information, see the help topic Customize Your NetSuite Account.</p> <p>Then, in the develop account, use the Bundle Builder to create a bundle with the customization. See Creating a Bundle with the Bundle Builder.</p>
2	<p>Copy the bundle to the release account. Copy a bundle to the release account to support your release process. See Copying a Bundle to Other Accounts.</p> <p>Note: This account should only contain copied bundles, and not any objects created as part of the customization.</p>
3	<p>Install the bundle from the release account into the test account. You can create sample data to support testing of the customization. See Installing a Bundle.</p>

Step Number	Description
	If you want to make changes to the customization after testing, make the changes on the develop account and repeat steps 1 to 3 above.
4	After the bundle has been tested on the test account and all desired changes are incorporated into the bundle and copied to the release account install the bundle into a sandbox account for additional integration and user acceptance testing with production data. If you want to make changes to the customization after testing, make the changes on the develop account and repeat steps 1 to 4 above.
5	After the bundle has been completely tested in the sandbox account, install the bundle into the production account. You can now begin development on a second version of the bundle in Development Account Environment 2 .
6	Copy the bundle from the release account on Development Account Environment 1 to Development Account Environment 2 .
7	Install the bundle from the release account to the develop account . You can now begin development of a second version of the bundle, using the customization objects in the first version of the bundle as a starting point.

Multiple Development Account Environments Method — Benefits and Drawbacks

The following table lists the benefits and drawbacks to this approach:

Benefits	Drawbacks
<ul style="list-style-type: none"> ■ Bug fixes. Can fix bugs in one environment and at the same time developing new version in a separate environment. ■ Separate code bases. Bundle being developed separate from release version. ■ Separate accounts. De-couples development, test, and release accounts. ■ Avoids bundle overwrites. Inadvertent overwriting of bundle content does not occur. ■ Testing with production data. Uses sandbox account for user acceptance and integration testing with production data. 	<ul style="list-style-type: none"> ■ Team development. Multiple developers working on a single version of a SuiteApp would be accessing the same account. ■ SuiteApp versioning. Can be difficult to manage SuiteApp versioning between development cycles.

Bug Fixing Methodology

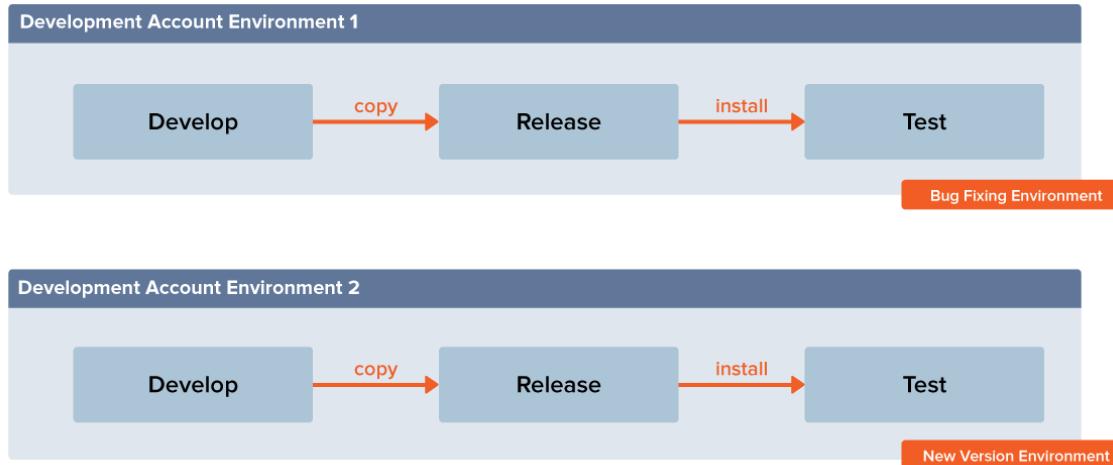
The ability to fix bugs and develop a new version of a bundle, and at the same time hosting the previous, stable version meets an independent software vendor's (ISV's) development, release, and customer support needs. This method provides the mechanism that an ISV can use to both maintain customers who have installed the bundle and continue to develop a new version.

A developer can fix bugs in the develop account and test the resolution in the test account in a development account environment. The developer then copies the bundle to the release account. Then, administrators of target accounts can install or update the bundle, which contains the bug fixes.



Important: By decoupling the bundles in the release account from the develop account, changes can be made to the bundle, tested, and then released, without impacting the development of the new version of the bundle in the second development environment.

The following diagram shows **Development Account Environment 1** being used as the bug fixing environment and at the same time the new version of the bundle is developed in **Development Account Environment 2**:



SuiteApps and Sandbox Accounts

NetSuite sandbox accounts provide areas for the development and testing of bundles without disrupting day-to-day business activities in your NetSuite production account. You can use one or more sandbox accounts for development and testing of your own bundles as well as for testing bundles from external sources such as independent software vendors (ISVs).

Sandbox accounts provide:

- a safe place to test customizations to NetSuite, such as bundles
- a realistic starting point for development, because they initially mirror your production account, including its configuration, data, and customization

For general information about using sandbox accounts, see the help topic [NetSuite Sandbox](#). The NetSuite Development account is an alternative to using Sandbox. For details, see the help topics [The Development Account](#), [NetSuite Development Accounts](#), and the [SuiteApp Development Process with SuiteBundler](#).



Note: Support for bundle operations varies across the different types of NetSuite accounts, including sandbox accounts. See [Bundle Support Across Account Types](#).

Sandbox Bundle Deployment Models

Review the following for brief descriptions of sandbox bundle deployment models:

- [Single Sandbox Bundle Deployment Model](#)
- [Two Sandbox Bundle Deployment Model](#)

Limitations for Bundles in Sandbox Accounts

Some features are limited in their functionality in sandbox accounts, particularly features related to payroll and credit card processing. When you are developing or testing bundles, it is important that you account for these limitations in the sandbox account. For a complete list of limitations, see the help topic [Features Available for Testing in a Sandbox](#).

Also, be aware of the following:

- You cannot deprecate a bundle that has been created in a sandbox account. However, if a bundle that has been installed in a sandbox account has been deprecated, the bundle in the sandbox account can be updated with the replacement bundle.
- The Copy action is not supported in sandbox accounts.
- Unlike managed bundles installed in production accounts, managed bundles installed in sandbox accounts must be updated manually. For more information about managed bundles, see [Managed Bundles Overview](#).
- The Push action is not available in sandbox accounts. If you want to install a bundle from a sandbox account to another account where you have administrator access, you will need to go to that target account to search for and install the bundle.
- You should not check the Hide In SuiteBundle box for a script to be included in a bundle developed in sandbox. Checking this box for a script in the source sandbox account for the bundle can eventually lead to the script being hidden in that account. After the bundle is installed in production and the

sandbox account is later refreshed, the production account setting for this option is copied to the sandbox account. This copy causes the Hide in SuiteBundle box to be checked and disabled in the source sandbox account, preventing bundle developers from accessing the script. For information about this option, see [Protecting Your Bundled Server SuiteScripts](#).

- You should not lock bundle objects to be included in a bundle developed in sandbox. Locking objects in the source sandbox account for the bundle can eventually lead to the objects being locked in that account. After the bundle is installed in production and the sandbox account is later refreshed, the production account settings for locked objects are copied to the sandbox account. This copy causes the objects to be locked in the source sandbox account, with their lock options disabled, preventing bundle developers from editing these objects. For information about setting lock preferences, see [Locking Objects in Customization Bundles](#).
- Support for bundle operations varies across the different types of NetSuite accounts, including sandbox accounts. See [Bundle Support Across Account Types](#).



Warning: Users are strongly cautioned NOT to rebundle objects that are installed in sandbox from other bundles. If such objects are rebundled, users are also cautioned not to install these bundles in production. Installing bundles with rebundled sandbox objects in a production account has a particularly detrimental impact on SuiteApps, including managed bundles from NetSuite or partner-bundled solutions. These production rebundles can result in an unexpected state of the bundle and account data, and actions to resolve such a state may result in data loss. Users can reduce the risk of such rebundling unintentionally occurring by avoiding the use of the Bundle All option. Manually selecting components ensures that all components are not added by default.

Specialized Features for Sandbox Bundle Development

Sandbox bundle development involves issues that you do not encounter for bundles installed from an external (ISV) account into your production account. Because your sandbox account is a copy of your production account, the objects included in a bundle created in sandbox are likely to also exist in production. Because of this copying, bundle updates from sandbox to production usually involve significantly more object conflicts than updates of ISV bundles. Refreshes of a sandbox account can result in overwriting of bundle objects and loss of data, due to changes made to bundle objects in production. See [Sandbox Refresh Impact on Bundles](#).

SuiteBundler includes the following features that address unique issues for bundles developed in sandbox accounts:

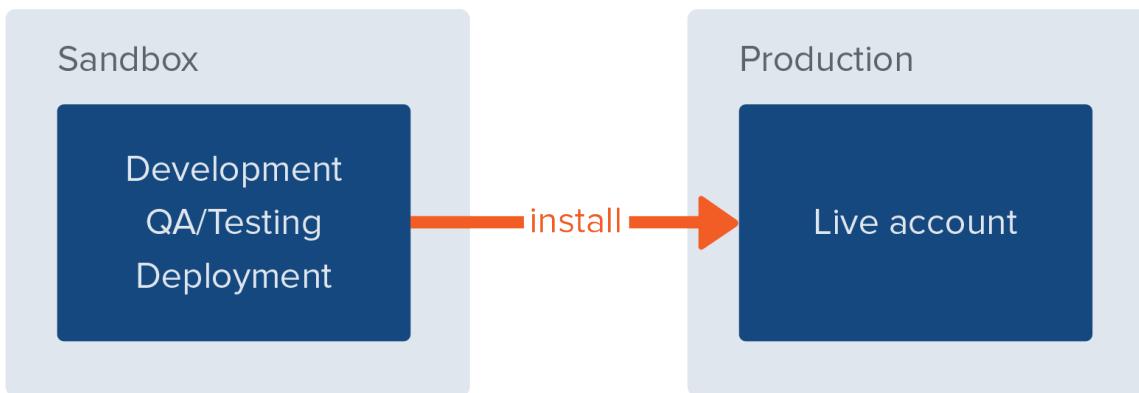
- [Selective Update of Sandbox Bundle Objects](#)
- [Dissolving Bundles Created in Sandbox](#)

Single Sandbox Bundle Deployment Model

You can develop and test bundles using a single sandbox account. This simplified model is appropriate for smaller customization projects. With this model, administrators and testers can work on bug fixes and customizations in sandbox without disrupting normal business activities in production. However, this model includes the risk that development or testing activities could interfere with each other.

After you develop and test a bundle in the same sandbox account, you can safely install it in your production account.

You can also test bundles created by administrators in other companies or by ISVs in a single-sandbox environment prior to installing that bundle in your production account.



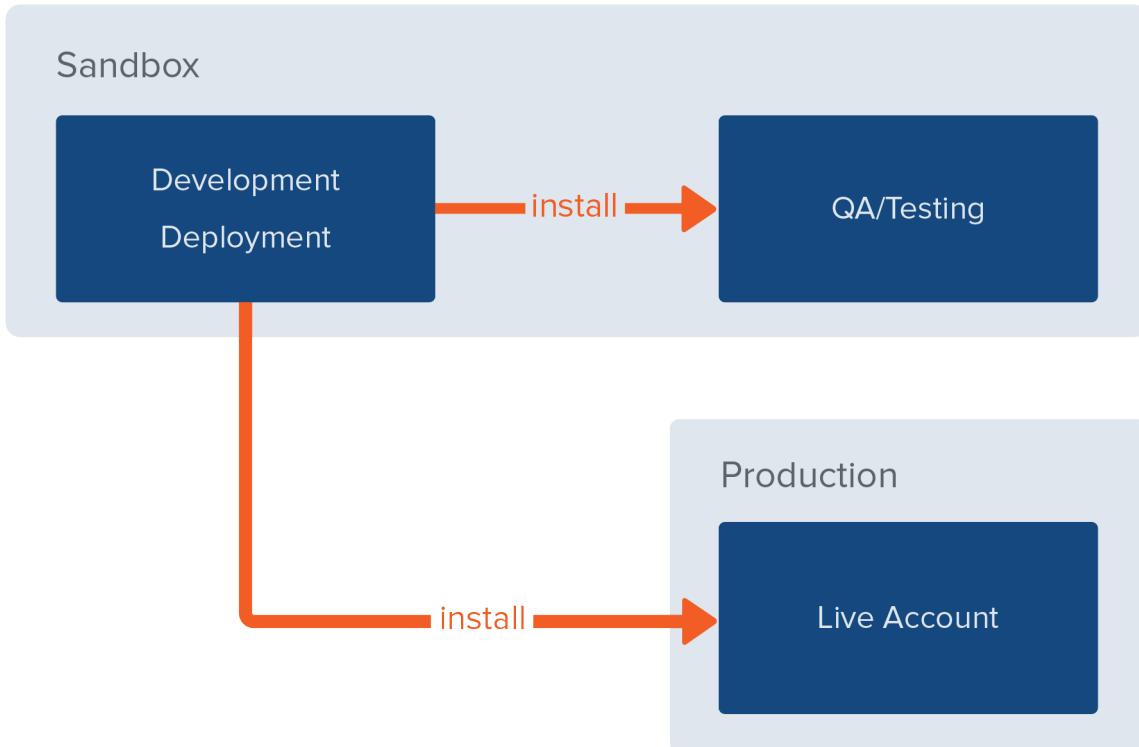
For information about installing a bundle, see [Installing a Bundle](#).

Two Sandbox Bundle Deployment Model

If your company has two sandbox accounts, you can dedicate separate sandboxes for development and testing. With this model, you create a bundle in a development (DEV) sandbox, install the bundle in another sandbox for testing (QA), and then deploy the tested bundle from the DEV sandbox into your production account.

Because this model provides a dedicated sandbox account for testing, it ensures that development and testing activities do not interfere with each other. Administrators can continue work on future versions of the bundle or other customizations in the DEV sandbox account at the same time that the bundle is tested in the QA sandbox account.

Important: If you use a two sandbox model, be sure to deploy from the development sandbox account into production and never from the QA sandbox account. If you deploy from the QA account, the development sandbox account will lose the bundle definition after being refreshed from production.



Sandbox Refresh Impact on Bundles

A refresh of a sandbox account overwrites the sandbox account contents with the related production account contents. The intent is that, after the refresh is complete, the sandbox account mirrors the related production account. However, the handling of bundles during refreshes may result in some differences between bundles in the sandbox account and bundles in the production account.

A sandbox refresh has the following impacts on bundles:

- A bundle definition for a bundle created in a sandbox account may not be preserved in the sandbox account after a refresh.
 - If the bundle was installed in the related production account before a refresh, NetSuite maintains the bundle definition in the sandbox account after the refresh. NetSuite also maintains the link between the bundle's source and target accounts.
 - If the bundle was not installed in the related production account before a refresh, the bundle definition is not maintained in the sandbox account after the refresh.
 - You also must take steps to preserve any changes made to a sandbox account between the latest production snapshot at the time of the refresh request and the time the refresh is activated. See [Preserving Changes Made to Sandbox Accounts During the Refresh Process](#).



Warning: Bundles developed in sandbox accounts should be used only for account customizations in linked production accounts. A refresh of a sandbox account used as a source for bundle installation may result in broken bundle operation in target accounts other than the linked production account. To avoid this issue, do not install bundles in your production account from sandbox accounts other than your own. And do not allow installation of bundles developed in your sandbox account to others' production accounts.

- A bundle definition for a bundle created in a production account before a refresh is not transferred to the related sandbox account during a refresh.
 - The bundle definition is NEVER transferred from production to sandbox because a bundle can have only one source account. If the bundle definition were transferred from production to sandbox during a refresh, both production and sandbox accounts would be a source account of this bundle. It is not feasible to have multiple source accounts for a single bundle.
 - This type of bundle will not be listed on the Saved Bundles page in the sandbox account after a refresh. The Saved Bundles page in a sandbox account only lists bundles that were recreated, that have a bundle definition in the sandbox account. The Installed Bundles page lists all bundles installed from other accounts into the sandbox account, including bundles copied to the sandbox account as part of a refresh.
- If any objects from a bundle created in a sandbox are deleted from the corresponding bundle in the production account, when the sandbox account is later refreshed, these deletions have an impact on the bundle in the sandbox account. As part of the refresh, the objects that were deleted in production are deleted from the sandbox account, and are removed from the sandbox bundle definition



Warning: After a sandbox account has been refreshed from production, it cannot be restored to its prior state. For more information about sandbox accounts, see the help topic [NetSuite Sandbox](#). For more information about sandbox refreshes, see the help topic [Using the Sandbox Accounts Page](#).

Support for bundle operations varies across the different types of NetSuite accounts, including sandbox accounts. See [Bundle Support Across Account Types](#).

Preserving Changes Made to Sandbox Accounts During the Refresh Process

Before you activate a sandbox refresh, you must be sure to preserve any sandbox account changes that occurred during the refresh process. These updates will not be included in the newly activated sandbox. The contents of a refreshed sandbox are based on the latest snapshot of production account contents captured prior to the refresh request. For more information about sandbox refresh processing, see the help topic [Requesting a Refresh](#).



Warning: As the bundle author, ensure your users do not install any bundles from the sandbox during the sandbox refresh process. This applies to your customers or users at your company. Users installing bundles from the sandbox can later result in incorrect merging of different objects. To change bundle availability for your users, see [Bundle Availability](#).

There are two options available to preserve sandbox account changes:

- You can import updated sandbox account objects to a SuiteCloud project before activating the new sandbox. After activation, you can use SDF to deploy the updated objects to the new sandbox. This

action is the preferred preservation method. For details, see [Using SDF to Preserve Sandbox Account Changes](#).

- You can bundle updated sandbox account objects and install them in the related production account before activating the new sandbox. After activation, you can use SuiteBundler to install the updated objects to the newly activated sandbox account. For details, see [Using SuiteBundler to Preserve Sandbox Account Changes](#).



Warning: Do not click the **Activate** button for the sandbox refresh until you have taken action to preserve all of the objects that have been updated during the refresh process. Otherwise the updates to these objects will be lost after the new sandbox is activated. You can preserve changes through import to a SuiteCloud project, or through bundling and installation in the related production account. If there have not been any updates since the refresh request was initiated, no preservation action is needed.

Using SDF to Preserve Sandbox Account Changes

SDF users can import updated sandbox account objects into a SuiteCloud project before the activation of a sandbox refresh, and then deploy those objects to the sandbox account after the refresh has been activated.

Follow these steps to use SDF to preserve sandbox account changes:

1. When you receive the email notification that the sandbox refresh is ready to be activated, identify the changes that were made since the initiation of the refresh request.
The changes are those you have made since the snapshot timestamp. To view the snapshot timestamp, go to Setup > Company > Sandbox Accounts. The snapshot timestamp is located in the **Snapshot Date** field.
2. Preserve the changes by importing the desired custom objects into a SuiteCloud project. See the help topic [Account Component Imports to SuiteCloud Projects](#).
3. Activate the sandbox refresh. See the help topic [Requesting a Refresh](#).
4. After the sandbox refresh has completed, deploy the custom objects from the SuiteCloud project to the refreshed sandbox account. See the help topic [Deployment Logs for SuiteCloud Projects](#).

Using SuiteBundler to Preserve Sandbox Account Changes

You can preserve updated sandbox account objects by packaging them in a bundle and installing the bundle to a related production account before activating the refresh. Later you can install the objects into the sandbox account after the refresh has been activated.

Follow these steps to use SuiteBundler to preserve sandbox account changes:

1. When you receive the email notification that the sandbox refresh is ready to be activated, identify the changes that were made since the initiation of the refresh request.
The changes are those you have made since the snapshot timestamp. The snapshot timestamp is included in the email notification.
2. Bundle the updated sandbox objects that you have identified and install the bundle into a related production account. See [Creating a Bundle with the Bundle Builder](#), [Selective Update of Sandbox Bundle Objects](#), and [Installing a Bundle](#).
3. Activate the sandbox refresh (see the help topic [Requesting a Refresh](#)).

- After the sandbox refresh has completed, bundle the updated objects from production and install the bundle into the refreshed sandbox account.

Selective Update of Sandbox Bundle Objects

The process for updating bundles installed into production from sandbox is different from the update process for other bundles, to meet the specialized requirements of bundles developed in sandbox accounts.

The normal bundle update process updates all bundle objects in the target account, overwriting target account objects with source bundle objects. The Preview Bundle Update page lists bundle objects to be updated. This page also highlights in red any objects in the target account that conflict with bundle objects, allowing the user to choose whether to add and rename each bundle object, leaving the target account object intact, or to overwrite the existing target account object with the bundle object.

In contrast, the update process for bundles installed into production from sandbox allows selective updates to production objects. When you start the update process for this type of bundle, the system evaluates whether objects have changed in production or sandbox since the last bundle update, and makes intelligent decisions about whether the update should overwrite each production object. This limitation of updates to a subset of the production objects can improve performance and can prevent unintentional overwrites of production object changes.

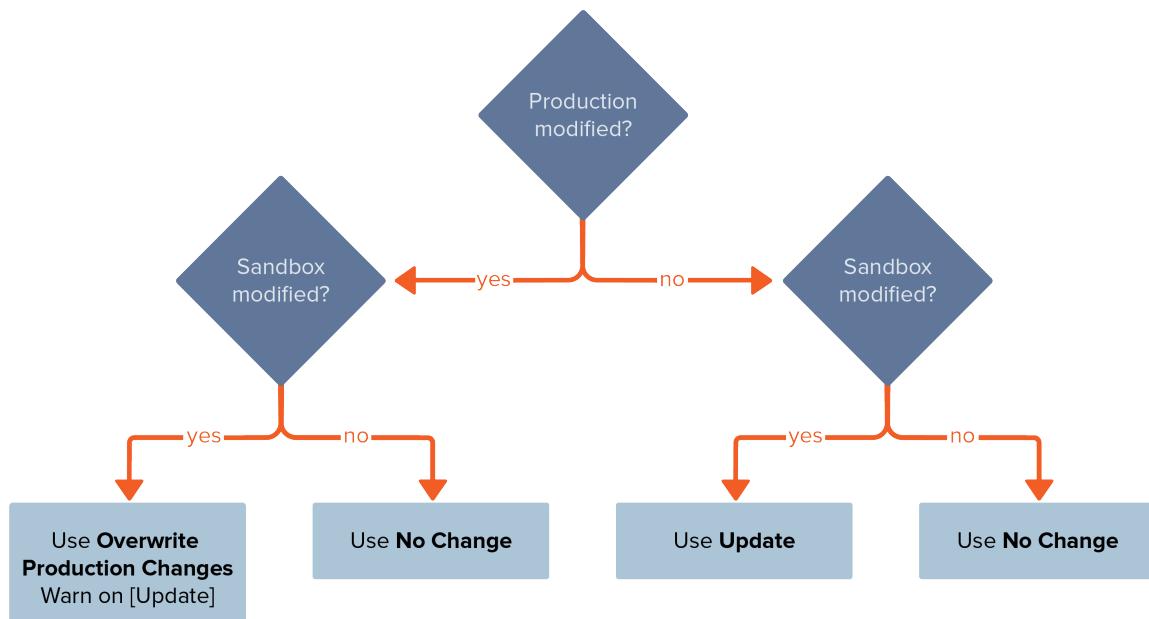
Note: The process for updating bundles installed into a sandbox account from a development account is the same as the update process for other bundles. The normal bundle update process updates all bundle objects in the target account, overwriting target account objects with source bundle objects. For bundle updates from a development account to a sandbox account, the Preview Bundle Update page always shows the Update action for all objects, even for objects that have not changed in the source account since the last bundle update.

The following table shows how the update process for bundles installed into production from sandbox works for different cases:

Production object changed since last bundle update?	Sandbox object changed since last bundle update?	Bundle Update Action	Action Text on Preview Bundle Update Page
No	No	No change to the production object.	No Change
No	Yes	Production object overwritten by sandbox object.	Update
No	Yes	Production object and data deleted. Occurs on any custom field when the type in the source is changed so that it is incompatible with the type of data for the custom field in the production account.	Delete then Re-add
Yes	Yes	Production object overwritten by sandbox object (Note that this may overwrite production object changes.)	Overwrite Production Objects (with warning)
Does not yet exist in production	Yes (added to bundle)	Sandbox object is added to production.	Add

—	Yes (removed from bundle)	Production object is removed.	Delete (with warning)
---	---------------------------	-------------------------------	-----------------------

The following diagram provides another illustration of the update process for bundles installed into production from sandbox. This diagram shows how the update handles objects from the sandbox bundle that also exist in production:



The Preview Bundle Update page for bundles installed into production from sandbox clearly displays the actions to be taken during the update. This page allows users to understand exactly what happens to each production object during the update. Each object displays action text indicating what the update will do to the production object. The preview page highlights in red the Overwrite Production Changes and Delete actions, and provides hover text warnings for each of these actions. Also, when the user clicks the Update Bundle button on this page, a list of the production objects that will be deleted, or that will have changes overwritten, is displayed. These alerts allow you to review these objects in the production account before the update. You can verify that overwriting the changes will not be a problem, or keep a record of the changes so the objects can be manually fixed after the update. You also can decide not to proceed with the update.

Preview Bundle Update

Name: Student Courses **Current Version:** 1.0 **New Version:** 1.1

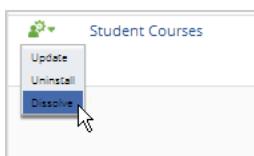
BUNDLE CONTENTS	SCRIPT ID	ACTION	PREFERENCE	LOCK ON INSTALL
Custom Fields				
Entity Fields				
Training ID	custentity_training_id	⚠ Delete then Re-add		
Student Courses Search	custentity_saved_search_ref	No Change		
Training Dept	custentity_training_dept	No Change		
Custom Lists/Records				
Lists				
Level of Difficulty	customlist_level_difficulty	No Change		
Employee Classes	customlist_employee_classes	No Change		
Student Rating	customlist_list_student_rating	Update	Preserve Data ▾	
Instructor Rating	customlist_instructor_rating	Add		
Records				
Student Courses	customrecord_student_courses	⚠ Overwrite Production Changes	Preserve Data ▾	
Forms				
Entry Forms				
Customer with Classes		No Change		
Saved Searches				
Student Courses				
Student Courses Search	customsearch134	⚠ Delete		

Dissolving Bundles Created in Sandbox

The Dissolve action is available for bundles created in a sandbox account that have been installed in a linked production account, to eliminate bundle definitions that are no longer needed in the production account. The Dissolve action removes the bundle definition from the production account, so the bundle is no longer listed on the Installed Bundles page for that account. When a bundle created in sandbox is dissolved, the customization objects installed in production as part of the bundle remain in the production account and do not lose any functionality. They are no longer considered part of a bundle.

The Dissolve action is available for any bundle installed into a production account from a linked sandbox account (meaning a sandbox account that belongs to the same company as the production account). So this action can be used for other bundles in addition to orphaned bundles (bundles with a definition but no objects in a production account). For example, when the development cycle of a bundle is complete, you can dissolve a bundle in a production account where it has been installed, if you want the bundle objects to be treated as native objects rather than bundled objects in the production account.

Account administrators can find the Dissolve action at Customization > SuiteBundler > Search & Install Bundles > List, as an option in the Action dropdown list.



When a user selects the Dissolve action, a preview page displays. This page lists the objects in the bundle whose definition will be dissolved and allows the user to either cancel or proceed with the action. Note

that the objects are listed for information purposes; they are not dissolved, the bundle definition is dissolved.

Information
This action will remove the bundle definition from your account, and the bundle will no longer be listed on the Installed Bundles page. All bundled objects listed below will remain in the account but will no longer be associated with the bundle.

Preview Bundle Dissolve Details	
<input type="button" value="Cancel"/>	<input type="button" value="Dissolve"/>
Name: Student Courses	
NAME	ID
Custom Fields	
Entity Fields	
Training Dept	custentity_training_dept
Student Courses Search	custentity_saved_search_ref
Training ID	custentity_training_id
Custom Lists/Records	
Lists	
Employee Classes (12 records)	customlist_employee_classes
Level of Difficulty (4 records)	customlist_level_difficulty
Student Rating (4 records)	customlist_list_student_rating
Records	
Student Courses	customrecord_student_courses
Forms	
Entry Forms	
Customer with Classes	25

⚠ Important: You can only dissolve a bundle that was installed into a production account from a linked sandbox account. You will receive an error if you attempt to dissolve a bundle installed into production from a location other than your own sandbox. The Dissolve action is not available for bundles installed in sandbox accounts.

Avoiding Bundle Installation Errors due to Duplicate External IDs

The following steps provide an example of how an issue with duplicate external IDs may cause an unexpected error during bundle installation from a sandbox account to a production account:

1. Create a custom record with the script ID recorda in a production account, then create a custom record with the same script ID in a sandbox account manually or through a sandbox refresh.

```

1 Production:
2   Custom Record - recorda
3     instance1
4     instance2
5     instance3
6
7 Sandbox:
8   Custom Record - recorda

```

2. Assign external IDs to the custom record values in the production account using SuiteScript or web services, for example ext1, ext2, ext3.

```

1 Production:
2     Custom Record - recorda
3         instance1 ext1
4         instance2 ext2
5         instance3 ext3
6
7 Sandbox:
8     Custom Record - recorda

```

- In the sandbox account, assign external IDs to the custom record values using SuiteScript or web services.

In this example, the custom record values in the sandbox account have different script IDs than the custom record values in the production account, but they have the same external IDs ext1, ext2, and ext3.

```

1 Production:
2     Custom Record - recorda
3         instance1 ext1
4         instance2 ext2
5         instance3 ext3
6
7 Sandbox:
8     Custom Record - recorda
9         instance4 ext1
10        instance5 ext2
11        instance6 ext3

```

- Bundle the custom record in the sandbox account and install the bundle to the production account.

During bundle installation, there is a script ID conflict because custom record recorda has the same script ID in the production account and the sandbox account.

On the Preview Bundle Install page, select Replace Existing Object option to replace custom record recorda in the production account with custom record recorda from the sandbox account.

In this case, the custom record values are not matched between the production account and the sandbox account because the custom record values have different script IDs in the production account and the sandbox account.

Instead, the custom record values instance4, instance5 and instance6 are added to the production account. The addition of these record instances causes a conflict of duplicate external IDs among custom record values in the production account that causes the bundle install to fail with an unexpected error.

```

1 Production:
2     Custom Record - recorda
3         instance1 ext1
4         instance2 ext2
5         instance3 ext3
6
7         instance4 ext1
8         instance5 ext2
9         instance6 ext3
10
11 Sandbox:
12     Custom Record - recorda
13         instance4 ext1
14         instance5 ext2
15         instance6 ext3

```

- To fix the bundle installation error in the production account, you must match the custom record values between the production account and the sandbox account by external IDs, and rename the

script IDs of sandbox account custom record values using the script IDs of production account custom record values that have the same external IDs.

In this example, the custom record values in the production account are renamed from instance4, instance5, and instance6 to instance1, instance2, and instance3, allowing the bundle installation to proceed and match the custom record values between the sandbox account and the production account correctly.

```

1 Production:
2     Custom Record - recorda
3         instance1 ext1
4         instance2 ext2
5         instance3 ext3
6
7 Sandbox:
8     Custom Record - recorda
9         instance1 ext1
10        instance2 ext2
11        instance3 ext3

```

To avoid unexpected errors during a bundle installation from a sandbox account to a production account due to duplicate external IDs:

1. Log in to your sandbox account.
2. Go to Setup > Company > Enable Features and enable the SuiteCloud Development Framework feature on the SuiteCloud tab.
3. Go to Home > Set Preferences and enable Show ID Fields on Sublists.
4. Open the custom record.
5. Click Manage Values.
6. Go through each custom record value and click Edit, then click Change ID.
7. Rename the script IDs for sandbox account custom record values using the script IDs of production account custom record values that have the same external IDs.
8. Update the bundle. During bundle update, custom record values are matched between the sandbox account and the production account so the bundle installation proceeds correctly.

Note: This same fix is also applicable for custom segments that may have duplicate external IDs.

SuiteBundler Pages

This section provides links to topics about the pages used to create and install bundles with SuiteBundler. These topics are available as context-sensitive help when you click Help links on SuiteBundler pages.

- [Bundle Availability, Install Bundle Pages](#)
- [Bundle List Pages](#)
- [Bundle Action Pages](#)
- [Bundle Preview Pages](#)

Bundle Availability, Install Bundle Pages

- For details about the **Bundle Availability** page, see [Bundle Availability](#).
- For details about the **Bundle Install** page, see [Installing a Bundle](#).

Bundle List Pages

NetSuite provides the following list pages for bundle developers and installers. These lists provide details and an action menu to access bundle-related tasks for each bundle.

- The **Saved Bundles** page lists all bundles that you have created in your account. See [Reviewing the Saved Bundles List](#).
- The **Managed Bundles** page lists all bundles that you have created in your account and defined to be managed. See [Managed Bundles Overview](#).
- The **Installed Bundles** page lists all bundles that you have installed into your account. See [Reviewing the Installed Bundles List](#).

Bundle Action Pages

NetSuite provides the following pages where bundle developers can review status and take action on bundles in target accounts:

- The **Bundle Install Base** page lists details about accounts that have a bundle installed. See [Viewing Your Bundle Install Base](#).
- The **Push Bundle to Accounts** page lists the accounts where you can push a bundle to be installed (accounts where you have administrator access). See [Pushing a Bundle to Other Accounts](#).
- The **Copy Bundle to Accounts** page lists the accounts where you can copy a bundle. See [Copying a Bundle to Other Accounts](#).
- The **Copy Bundle Status** page lists details about bundles you have copied. See [Reviewing Copy Bundle Status](#).

Bundle Preview Pages

NetSuite provides the following pages where bundle installers can review bundle contents and set bundle object preferences before installation or update:

- For details about the **Preview Bundle Install** page, see [Reviewing the Preview Bundle Install Page](#).
- For details about the **Preview Bundle Update** page, see [Reviewing the Preview Bundle Update Page](#).

Custom Transaction Types in Bundles

If appropriate, you can create bundles that include custom transaction types. For details, see the following topics:

- [Pre-Bundling Checklist](#)
- [Including a Custom Transaction Type in a Bundle](#)
- [Other Recommended Objects](#)
- [Post-Installation Steps](#)
- [Removing Custom Transaction Types](#)
- [Accounting Preferences for Custom Transactions Applied to a Bundle](#)

For more information about custom transactions, see the help topic [Custom Transactions](#).



Note: SuiteBundler is still supported, but it will not be updated with any new features.

To take advantage of new features for packaging and distributing customizations, you can use the Copy to Account and SuiteCloud Development (SDF) features instead of SuiteBundler.

Copy to Account is an administrator tool that you can use to copy custom objects between your accounts. The tool can copy one custom object at a time, including dependencies and data. For more information, see the help topic [Copy to Account Overview](#).

SuiteCloud Development Framework is a development framework that you can use to create SuiteApps from an integrated development environment (IDE) on your local computer. For more information, see the help topic [SuiteCloud Development Framework Overview](#).

Pre-Bundling Checklist

Before bundling your custom transaction type, you may want to review the type. In particular, check the type's settings for each of the following:

- [Transaction Style](#)
- [Statuses](#)
- [Class, Department, and Location](#)
- [Custom Fields](#)

Transaction Style

Make sure that you have chosen the most appropriate list style for your custom transaction type. For example, you may have created a transaction type for which the entire GL impact should be determined

by a Custom GL Lines plug-in implementation. For these types, the list style should be Header Only. For more details on list styles, see the help topic [Custom Transaction Styles Overview](#).

Statuses

If your transaction type includes statuses, check that the statuses are set up correctly. For example, you may want to make sure that the posting flag has been set as needed for each status. After the bundle is installed, the Statuses sublist cannot be modified in the target account. This limitation exists even if you do not select the Lock On Install option for the transaction type. You can always make changes to the type's statuses in the source account and update the bundle. However, users in the target account are never able to change existing statuses or add new statuses. For more information about statuses, see the help topic [Statuses for a Custom Transaction Type](#).

Class, Department, and Location

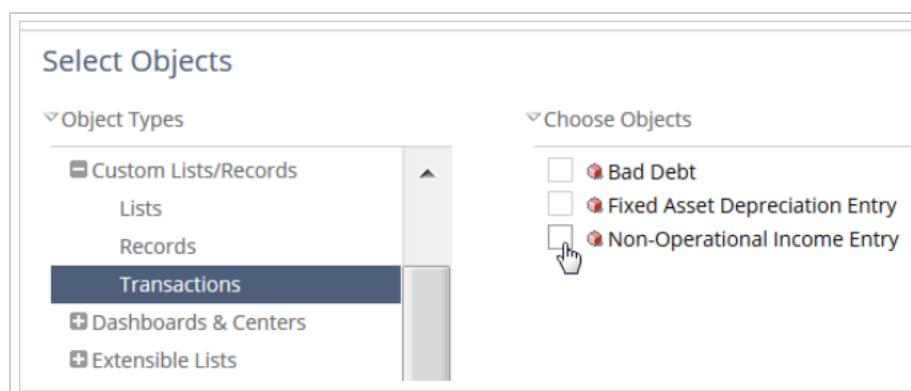
Check that your custom transaction type has the required settings for the class, department, and location fields. For help with these fields, see the help topic [Custom Transaction Type Classification Fields](#).

Custom Fields

Check the Custom Fields subtab of the custom transaction type. This subtab lists all custom body fields and column fields that are linked to the transaction type. If any of these fields should be omitted from the bundled version of the type, remove them before creating the bundle. To remove a field, click the name of the field to open it for editing. On the **Applies To** subtab, cancel the selection of the custom transaction type. For more details about custom fields, see the help topic [Custom Fields in Custom Transaction Types](#).

Including a Custom Transaction Type in a Bundle

You can add any custom transaction type to a bundle. In the Bundle Builder's **Object Types** list, select **Custom Lists/Records > Transactions**. In response, the system updates the **Choose Objects** list to include all of your custom transaction types. You can then select the appropriate types to include.



For more information about custom transactions, see the help topic [Custom Transactions](#).



Note: SuiteBundler is still supported, but it will not be updated with any new features.

To take advantage of new features for packaging and distributing customizations, you can use the Copy to Account and SuiteCloud Development (SDF) features instead of SuiteBundler.

Copy to Account is an administrator tool that you can use to copy custom objects between your accounts. The tool can copy one custom object at a time, including dependencies and data. For more information, see the help topic [Copy to Account Overview](#).

SuiteCloud Development Framework is a development framework that you can use to create SuiteApps from an integrated development environment (IDE) on your local computer. For more information, see the help topic [SuiteCloud Development Framework Overview](#).

For general details on using the Bundle Builder, see [SuiteApp Creation and Distribution](#).



Note: If a saved search is filtered by a custom transaction type value, the custom transaction type is included in the bundle automatically.

Other Recommended Objects

In some cases, you may want to consider including the following objects with your custom transaction type:

- **A Custom GL Lines plug-in implementation** – If your bundle will include a Header Only style custom transaction type, consider adding a Custom GL Lines plug-in implementation to the bundle. An instance of a Header Only transaction type has no GL impact unless the type is associated with a Custom GL Lines plug-in implementation. For more information, see the help topic [Custom GL Lines Plug-in Overview](#).
- **A workflow** – If your transaction type includes statuses, consider adding a workflow to your bundle. You can use a workflow to set the status flag and take actions based on status values. For details on why you should use a workflow, see the help topic [Custom Transaction Type Statuses Overview](#). For information about creating a workflow, see the help topic [SuiteFlow Overview](#).

For more information about custom transactions, see the help topic [Custom Transactions](#).

Post-Installation Steps

Certain tasks cannot be managed by the bundle. These tasks must be completed by an administrator after the bundle has been installed. They include the following:

- If your bundle includes a Basic, Sales, or Purchase style transaction type, an administrator in the target account must define the offset account for the type. For more information, see the help topic [Account Field Setup for Custom Transaction Types](#).
- If the bundle includes a Custom GL Lines plug-in implementation, an administrator in the target account must configure the plug-in implementation. For details on this process, see the help topic [Configuring the Custom GL Lines Plug-in Implementation](#).

- By default, only account administrators have permission to work with instances of any custom transaction type. If other roles require permission, an administrator must configure it. For details, see the help topic [Permissions for Custom Transaction Instances](#).

Additionally, an administrator may want to review and customize the transaction type's numbering logic. For details, see the help topic [Numbering for a Custom Transaction Type](#).

Removing Custom Transaction Types

When a custom transaction type is installed as part of a bundle, it can also be removed, even if instances of that transaction type exist. You can remove the type by uninstalling the bundle. Additionally, a bundle update can remove the type.

This behavior differs from how the system treats transaction types that were created in the UI. Transaction types created in the UI cannot be deleted if an instance of the type exists.

When a bundle operation is used to delete a transaction type, the system takes the following actions with any existing instances of that transaction type:

- The transactions are migrated to a new transaction type called Deprecated Custom Transaction.
- The transactions are locked to editing, although they can still be voided.
- If the transactions are in open periods, they can be deleted.
- Body fields are added to the transaction showing:
 - The name of the custom transaction type that was deleted.
 - The status of the transaction at the time the transaction type was deleted.
- Associated printing templates are deleted.
- Custom fields are discarded unless they are related to a custom segment that has GL impact.

To see a list of transactions that were instances of deleted custom transaction types, use the Transaction search type. Set the Type field to Deprecated Transaction Type.

For more information about custom transactions, see the help topic [Custom Transactions](#).

Accounting Preferences for Custom Transactions Applied to a Bundle

Partners can create bundles for customers that include custom transactions. Setup accounts defined in custom transactions cannot be transferred, except for Default Accounts Receivable and Default Accounts Payable accounts. Customers are responsible for specifying the accounts to use after the bundle installation.

If the bundle is not locked, you can specify an account or a default account to use in the custom transaction type account settings. For more information, see the help topic [Entering the Account for Sales and Purchase Transaction Types](#).

If the bundle is locked, you select the account or default account on the Accounting Preferences page. If the Default Account is Default Accounts Receivable or Default Accounts Payable, then neither the Account nor Default Account fields display.

To select the account:

1. Go to Setup > Accounting > Account Preferences and click the **Items/Transactions** sublist.
2. Under the Accounts heading, locate the Account or Default Account field for the custom transaction.

For a Basic custom transaction type, an Account field is displayed. For Sales and Purchase custom transaction types, the field can be either Account or Default Account, depending on what is required for the custom transaction.

3. Select the required account or default account.
4. Click **Save**.

For more information about custom transactions, see the help topic [Custom Transactions](#).

Adding a Custom Segment to a Bundle

You can create bundles that include custom segments. When creating the bundle, custom segments are listed under Custom Lists/Records on the Select Objects page of the Bundle Builder. For more information about SuiteBundler, see [SuiteBundler Overview](#) and [Creating a Bundle with the Bundle Builder](#).



Note: SuiteBundler is still supported, but it will not be updated with any new features.

To take advantage of new features for packaging and distributing customizations, you can use the Copy to Account and SuiteCloud Development (SDF) features instead of SuiteBundler.

Copy to Account is an administrator tool that you can use to copy custom objects between your accounts. The tool can copy one custom object at a time, including dependencies and data. For more information, see the help topic [Copy to Account Overview](#).

SuiteCloud Development Framework is a development framework that you can use to create SuiteApps from an integrated development environment (IDE) on your local computer. For more information, see the help topic [SuiteCloud Development Framework Overview](#).

For the limitations of custom segments in bundles, see the following:

- [Creating and Installing Bundles with Segments](#)
- [Updating Bundles with Segments](#)
- [Uninstalling Bundles with Segments](#)
- [Correcting a Damaged Custom Segment](#)
- [Updates Required for Some Bundle Objects that Reference Custom Segments](#)

For more information about custom segments, see the help topic [Custom Segments](#).

Creating and Installing Bundles with Segments

When a bundle is installed, all selected objects are copied as well as their settings, such as GL impact and where the segment will be used. Whenever a custom segment that is part of a parent/child hierarchy is included in a bundle, all segments in the hierarchy are included in the bundle automatically.

For more information about custom segments, see the help topic [Custom Segments](#).



Note: The Display Type field in the custom segment configuration is not supported by SuiteBundler. The Display Type field is set to Normal on any custom segments that are included in a bundle.

Locked Custom Segments in Bundles

If a custom segment is locked in the Bundler, you cannot change it from the user interface in the target account.

If permissions are assigned for a locked custom segment and a standard role, those permission levels are included with the bundle. For locked custom segments, no one can edit the custom segment definition and only an administrator and the roles specified can manage custom segment values.

To add or change the roles that are allowed to manage custom segment values, you can assign the value management permission to a custom role for specific custom segments.

To assign the value management permission to a custom role:

1. Go to Setup > Users/Roles > Manage Roles.
2. Click **Edit** next to the custom role that you have created.



Note: You cannot assign access to locked custom segments for built-in NetSuite roles.

3. On the Permissions > Setup subtab, select **Custom Segments** from the **Permission** list.
4. In the **Level** field, select the level of access that this role requires for custom segments.
5. On the Permissions > Custom Record subtab, select the custom segment from the **Record** list.
6. In the **Level** field, select the level of access that this role requires for custom segment values.
7. Click **Save**.

On the Permissions subtab of the Custom Segment configuration, the added custom role and value management access level are shown. No record access or search/reporting access levels are set for the custom role, so the default settings are applied.

Updating Bundles with Segments

When an installed bundle is updated in a target account, the custom segment definitions in the source bundle overwrite the settings in the account, with the following exceptions:

- Segment use settings are merged
- Display order is not changed
- GL Impact flags are not changed
- Segment values are preserved or merged.

For any custom segment that includes data, a preference can be set that is applied during bundle updates. This preference indicates whether: the target account custom segment data should be preserved, or data from the bundled custom segment and target account custom segment should be merged, during the update. The default is to preserve data.

A list is available for custom segments on the Set Preferences page of the Bundle Builder. The list includes **Preserve Data** and **Merge Data** options. When the **Preserve Data** option is selected, the bundle update does not make any changes to target account custom segment values. When the **Merge Data** option is selected, the bundle update merges custom segment values, without deleting any values from the target account custom segment. The avoidance of target account custom segment value deletion is due to the possibility that a custom segment can have GL impact. In addition, no option is provided to replace data for a custom segment, also to avoid unintentional changes to GL impact.

For more information about setting bundle preferences, see [Step 4 Set Preferences](#) and [Bundle Object Preferences](#).

If the custom segment type in the source account is changed, the type is updated on the custom segment in the target account. Be aware of the effect the change will have:

- If the custom segment type is changed from list/record to multi-select, the segment is removed from any transaction sublists. Transaction column applications are not valid for multi-select lists.
- If the custom segment type is changed from multi-select to list/record, data can be lost.

If the Allow GL Custom Segment Deletion preference is enabled, an update to a bundle can delete a custom segment. If the Allow GL Custom Segment Deletion preference is not enabled, an error appears when you attempt to update a bundle containing a deleted custom segment.

You should avoid deleting custom segments using the bundle update. The best practice is to inactivate any custom segments that are no longer needed.

For more information about custom segments, see the help topic [Custom Segments](#).

Uninstalling Bundles with Segments

If the Allow GL Custom Segment Deletion setting is enabled, bundles containing custom segments can be uninstalled. If the Allow GL Custom Segment Deletion setting is not enabled, an error appears when you attempt to uninstall a bundle containing a custom segment.

For more information about uninstalling bundles, see [Uninstalling a Bundle](#).

For more information about custom segments, see the help topic [Custom Segments](#).

Correcting a Damaged Custom Segment

A custom segment can sometimes become corrupted if any of its required components are missing. A completely defined custom segment must contain one custom record component, and one each of the following five custom field component types: Body, Column, Entity, Event, and Item. For information about the components of a custom segment, see the help topic [Creating a Custom Segment](#)

If any components are missing from the custom segment, you receive a segment corrupted error on the Custom Segment configuration page. However, you can continue to use the custom segment configuration page with limited operations available. As of 2020.1 if you installed a custom segment from a bundle and it is damaged, you can reinstall the bundle to correct the error. If you created a custom segment manually and it is damaged, delete the custom segment definition and recreate it.

When you edit the definition of a corrupted custom segment, only the **Cancel** button is available. Depending on your permissions, the **Delete** button may also be available.

If a form or applied record accesses a corrupted custom segment, you can save or view the record without the custom segment value .

For more information about custom segments, see the help topic [Custom Segments](#).