Service Cloud Workbook

Giving Your Customers Multi-Channel Support

30 min Tutorials

- Salesforce Console
- Computer-Telephony Integration
- Email-to-Case
- Web Chat
- Knowledge Base
- Web Communities

Table of Contents

About the Service Cloud Workbook....1

Tutorial #1: Setting Up a Help Desk....4

Step 1: Customize Highlights Panels....4
Step 2: Create a Salesforce Console App....5
Step 3: Choose How Lists Display....6
Step 4: Customize Push Notifications....7
Step 5: Grant Users Console Access....8
Summary....8
Tell Me More!....9

Tutorial #2: Turning Customer Emails into Cases Automatically....10

Step 1: Enable On-Demand Email-to-Case....10
Step 2: Define and Verify Routing Addresses....11
Step 3: Test and Add Email Addresses to a Website....12
Summary....12
Tell Me More!....13

Tutorial #3: Providing Self-Service on the Web with Communities....14

Step 1: Create a Community....14
Step 2: Brand Your Community....15
Step 3: Add Chatter Answers for Self-Service....16
Step 4: Create a Zone for Your Community....17
Step 5: Customize Community Tabs....19
Step 6: Publish Your Community....19
Step 7: View Cases from the Community in the Help Desk....19
Summary....20
Tell Me More!....20

Tutorial #4: Integrating Phone Calls....21

Step 1: Create an Open CTI SoftPhone....21
Step 2: Create a Call Center....23
Step 3: Add Users to a Call Center....24
Step 4: View the SoftPhone in the Help Desk....24
Summary....26
Tell Me More!....26

Tutorial #5: Chat Live with Customers on the Web....27

Step 1: Enable Live Agent....27
Step 2: Configure Live Agent Users....27
Step 3: Create Agent Skills and Assign Them to Users....29
Step 4: Set Up Live Agent Configurations....30
Step 5: Create Chat Deployments....32
Step 6: Create Chat Buttons....33

Step 7: Create Automated Chat Invitations....34
Step 8: Add Live Agent to the Help Desk....36
Step 9: View Live Agent in the Help Desk....37
Summary....38
Tell Me More!....38

Tutorial #6: Storing and Retrieving Knowledge....40

Step 1: Create Article Types....40
Step 2: Enable Salesforce Knowledge and Select Settings....41
Step 3: Create Data Category Groups....42
Step 4: Set Data Category Group Visibility....43
Step 5: Grant Users Access to Salesforce Knowledge....44
Step 6: Turn on the Knowledge Sidebar in the Help Desk....44
Step 7: View the Knowledge Sidebar in the Help Desk....45
Summary....46
Tell Me More!....46

Tutorial #7: Streamlining the Look and Feel of Cases....47

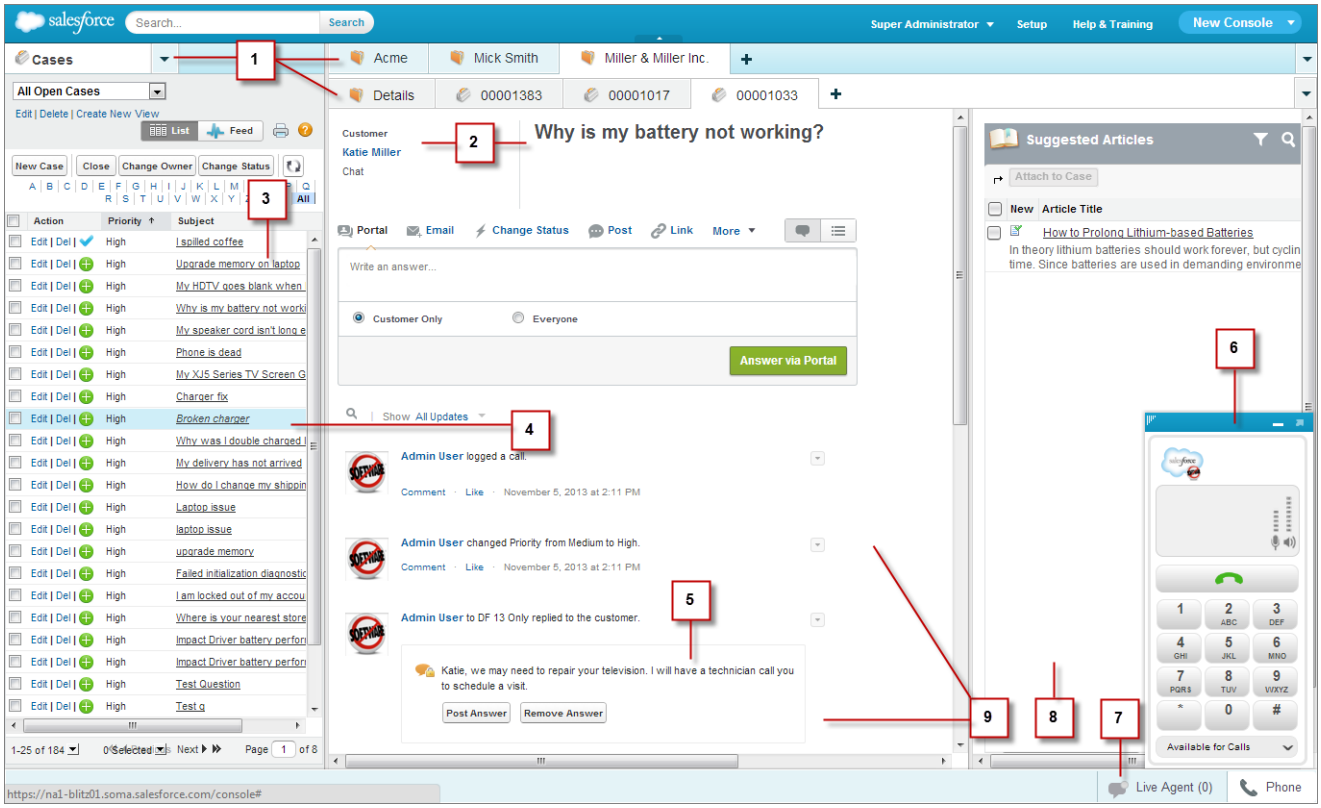
Step 1: Customize What Agents See in the Publisher....48
Step 2: Customize the Case Detail View....49
Step 3: Customize the Feed View....50
Step 4: View a Case in the Help Desk....51
Summary....52
Tell Me More!....52

Final Summary....53

About the Service Cloud Workbook

Gone are the days when support agents just needed phones to help customers. Now, there's email, websites, online chats, social media—and who knows what's next? It might seem like finding and deploying all the tools agents need to make customers happy is as difficult as climbing Mount Everest. But it's not. With the Service Cloud, you can set up entire call centers in a few hours. Plus, with minimal training, agents can use the Service Cloud to quickly answer customers' questions, whether by phone, text chat—you name it!

The *Service Cloud Workbook* shows you how your agents can monitor and respond to all of your customer channels from one screen. Through a series of tutorials, we'll show you how to click a few buttons to set up an awesome help desk for your service team. Then, we'll add your customers' preferred channels, and integrate those channels with the help desk. When you finish the tutorials, you'll have customized a user interface that helps your agents decrease clicks and context-switching to decrease customer-response times. And, you'll have set up a few features that your customers will love because they'll be able to contact you by their favorite channel. By the end of the exercises, your help desk, or what we call a Salesforce console, will look something like this:



1. Tabs for easy navigation
2. Highlights panel to spot important items quickly
3. Customized lists to display to-do items
4. Push notifications to see when items are updated by others
5. Customer inquiries from emails and the Web
6. Call-control tools to receive and make phone calls
7. Text-chat to work with customers in real-time
8. Knowledge sidebar to find or suggest content from a knowledge base
9. Feed view for a streamlined way to interact with customer issues.

Intended Audience

This workbook is for administrators who want to give their customers more contact channels, and give agents new ways to receive and respond to customers' issues. It assumes administrators are already familiar with basic case management functionality, such as queues, workflow, escalation, and auto-response rules, and instead focuses on setting up a call center that centers around a help desk, otherwise known as a Salesforce console. This workbook doesn't cover security features, customizing Salesforce, or managing Salesforce users. Please see the online help for more information about those topics.

Before You Begin

This workbook is designed for administrators with Enterprise, Unlimited, or Performance Editions that have purchased the following user and feature licenses:

Licenses Needed	
To give users access to a Salesforce console:	Service Cloud User
To give customers access to Salesforce Communities:	Create and Customize Communities Customize Application
To give users access to Live Agent:	Live Agent User
To give users access to Salesforce Knowledge:	Knowledge User

Permissions You Need

This workbook is designed to be used with any organization that has Service Cloud features enabled. You'll need the following permissions to perform the exercises in this workbook:

User Permissions Needed	
To set up a Salesforce console:	"Customize Application"
To set up On-Demand Email-to-Case:	"Modify All Data" AND "Customize Application"
To set up Salesforce Communities:	"Customize Application"
To set up a Call Center with Open CTI:	"Manage Call Centers" AND "Customize Application"
To set up Live Agent:	"Customize Application" AND "Manage Users"
To set up Salesforce Knowledge:	"Customize Application" AND "Manage Salesforce Knowledge" AND "Manage Data Categories" AND

User Permissions Needed	
	"Manage Users"
To set up Case Feed:	"Customize Application" AND "Manage Users"

How is the Workbook Organized?

The tutorials in this workbook build on one another and are designed to be done in order.

Tutorial	Difficulty
Tutorial #1: Setting Up a Help Desk	Beginner
Tutorial #2: Turning Customer Emails into Cases Automatically	Beginner
Tutorial #3: Providing Self-Service on the Web with Communities	Intermediate
Tutorial #4: Integrating Phone Calls	Advanced
Tutorial #5: Chat Live with Customers on the Web	Advanced
Tutorial #6: Storing and Retrieving Knowledge	Advanced
Tutorial #7: Streamlining the Look and Feel of Cases	Beginner
Final Summary	

Tutorial #1: Setting Up a Help Desk

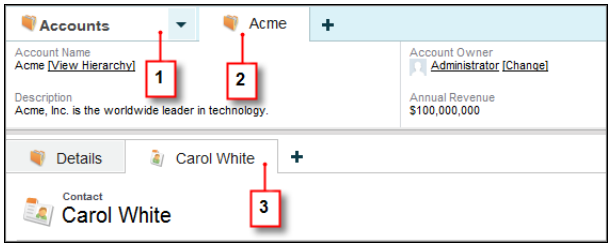
Level: Beginner

Before we get started, let's talk about help desks. Suppose your executives want to set up an application that helps your service team manage and answer customers' questions, but they don't want to invest in pricey on–premise systems that take weeks or months to implement. You could be a hero by suggesting this simple solution: let's set up a Salesforce console. A Salesforce console is designed for fast-paced environments, such as support, telesales, and telemarketing centers. Unlike standard Salesforce pages, all items clicked in a console display as tabs. Tabs let agents quickly:

- Find, update, and create cases or other records
- Review and update cases or other records from lists
- Display all the customer, product, and service information needed on one screen without losing context

There are three types of tabs in a console:

1. Navigation tab—lets users select an object like a case and view its lists.
2. Primary tab—displays the main item to work on, such as a case.
3. Subtab—displays items related to the primary tab, such as a case's account or contact.



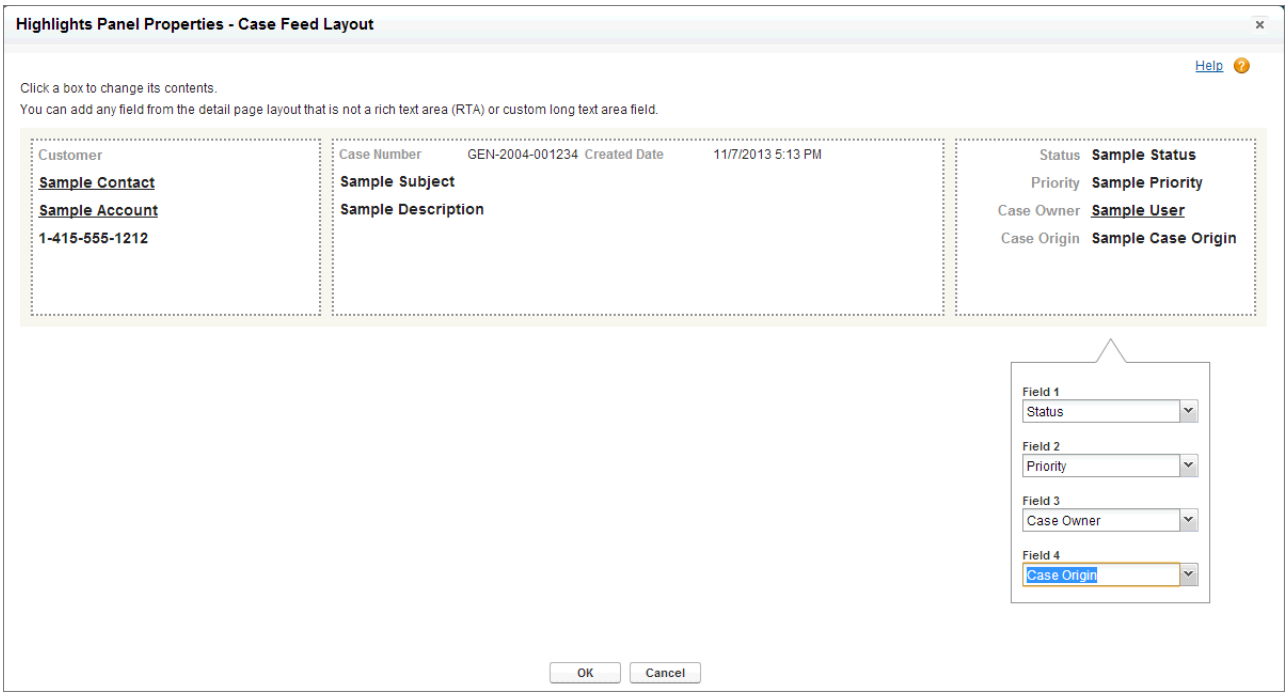
In this tutorial, we'll set up a console with basic functionality. When we're finished, you'll have customized the Salesforce user interface so that agents can view key info at a glance, work off of lists, and automatically see when the records they're working on have been changed by others. The end result will provide agents with an easy way to view and resolve cases, whether they come in from emails, the Web, or other channels.

Step 1: Customize Highlights Panels

Before we create a Salesforce console, we're going to choose the fields it will highlight. This will prevent your agents from being spooked by mysteriously highlighted fields later on. So let's start by customizing highlights panels. The highlights panel is a table of up to four columns at the top of every primary tab in a console—it lets agents view key information at a glance. Each of its columns has fields that you can customize to suit your business needs. For example, if you know that your agents look at `Priority` and `Status` fields most, you can add those fields to the case highlights panel so that agents can spot them easily.

1. From Setup, click **Customize** > **Cases** > **Page Layouts**.
2. In Page Layouts for Case Feed Users, click next to the Case Feed Layout and select **Edit detail view**. You'll learn about Case Feed later in Tutorial #7: Streamlining the Look and Feel of Cases.
3. Click **Edit** next to the Case Feed Layout.
4. Click .

5. Even though there are several fields already included in the highlights panel, let's add `Case Origin` to the last column so that your agents will see where each case comes from.



6. Click **OK**.
7. Click Layout Properties at the top of the page layout.
8. Select `Show in the Console` next to Highlights Panel.
9. Click **OK**, then **Save**.

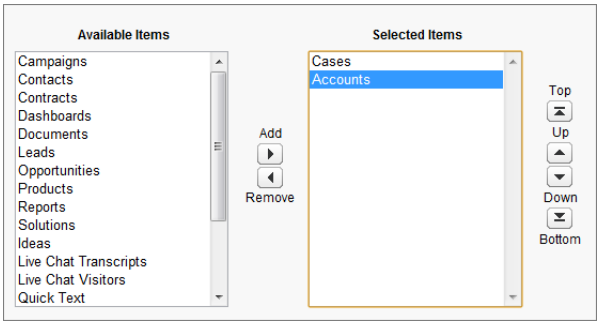
Next, we'll create a console app to display highlights panels. But before we do that, ask yourself which fields your agents need most on records. Now that you know how to customize a highlights panel, you can choose any page layout and add the most-needed fields on top so that agents don't waste time hunting for information they need.

Step 2: Create a Salesforce Console App

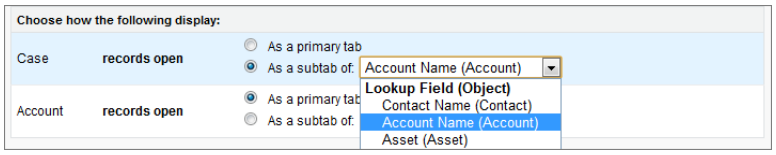
In the previous step, we did some prep work for a console—we chose the fields it will highlight for agents. In this step, we'll create a Salesforce console app so that agents can select it from the Force.com app menu, located in the upper right of every Salesforce page. This makes a console easy for anyone to find!

1. From Setup, click **Create** > **Apps**.
2. Click **New**.
3. Select `Console` and click **Next**.

4. Type a label for the app, such as `New Console`. This label will be the app's name in the Force.com app menu. If you want, you can choose a custom logo for the app, but let's skip this step.
5. Click **Next** .
6. Select the items to include in the navigation tab, which is a tab where agents can select objects and choose lists. Since your agents probably spend a lot of time reviewing case and account lists, let's add Cases and Accounts to the navigation tab.



7. Click **Next**.
8. Choose how items display in the console when they're not selected from a primary tab or subtabs. For example, when agents select cases from lists or search results, you can choose whether cases display as primary tabs or as subtabs on parent records. To decrease context-switching, let's choose that case records open as a subtab of their accounts so that support agents can view customers' inquiries along with customers' account information.



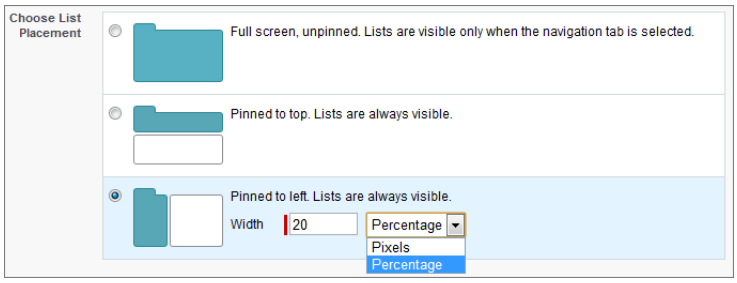
9. Click **Next**.
10. Check the `Visible` box next to any user profiles that will access the console. For example, if you have a user profile for agents, make the console visible to that profile.
11. Click **Save**.

We've created a console app and configured its tabs, and later we'll give agents access to your console so that they find it on any page in Salesforce. As an administrator, you have the flexibility to create many console apps, so think about whether your service team could use multiple consoles to streamline various business requirements. For example, if you have different support tiers that use different records, you could create a specific console for each tier.

Step 3: Choose How Lists Display

Let's choose how lists display in the Salesforce console app we created. Lists help agents work off of a specific set of records, such as cases with a `Priority` of High. We can choose a couple of ways to display lists in a console, but let's assume your agents have told you that they're most productive when lists are pinned to the left of the screen. We'll set that up for them in this step.

1. From Setup, click **Create** > **Apps**.
2. Click **Edit** next to our console app.
3. In Choose List Placement, select `Pinned to left. Lists are always visible.`



4. Specify a default size for lists. Let's type `20` in `Width` and choose `Percentage`.

   We recommend that you keep the width or height of lists to no more than 40% the size of a browser, otherwise your lists might look silly.

5. Click **Save**.

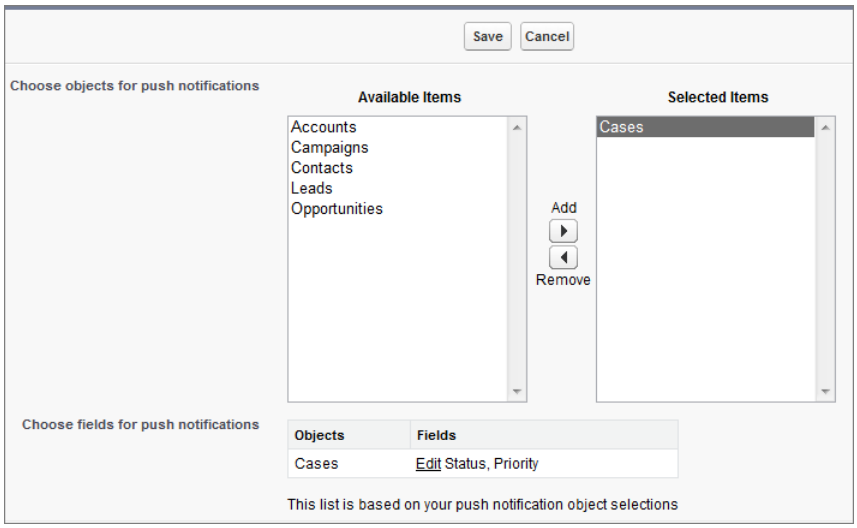
Wonderful, we're done with lists! Now, the lists in your console will only take up 20% the width of a browser. Keep in mind that your agents can resize any list by clicking on its margin and dragging it to the size they want. But if your agents get tired of the default list size, you can change it at any time.

Step 4: Customize Push Notifications

Push notifications are visual indicators on lists and detail pages in a console that show when a record or field has changed during an agent's session. For example, if two support agents are working on the same case and one agent changes the `Priority`, a push notification displays to the other agent so he or she notices the change and doesn't duplicate the effort. In this step, we'll choose when push notifications occur on cases, and we'll choose which case fields trigger push notifications. When you've finished this step, your agents should send you a thank you card because they'll rarely see any outdated cases.

1. From Setup, click **Create** > **Apps**.
2. Click **Edit** next to your console app.
3. In Choose How Lists Refresh, you can select when push notifications occur. Let's select `Refresh List Rows` so that lists automatically update when there are any changes to fields.
4. In Choose How Detail Pages Refresh, you can select when push notifications occur. Let's select `Flag` so a message appears when a record changes.

5. Click **Select objects and fields for notifications**.
6. Click **Edit**.
7. Select the objects for triggering push notifications. You probably want your agents to know when cases change, so let's move Cases from Available Items to Selected Items.
8. Under Fields, click **Edit** to select the case fields for triggering push notifications. Let's add `Priority` and `Status`, and then click **OK**.



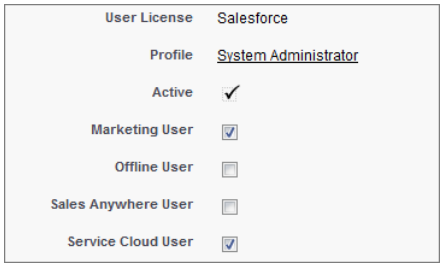
9. Click **Save**.

Wow, with just a few clicks, we've set up push notifications. Now, when agents start using your console, they'll see the most current stuff in lists and know when cases are being worked on by others. Imagine all the time you just saved your service team!

Step 5: Grant Users Console Access

Guess what? We're done setting up your console, and now we can assign users access to it. In Step 2 of this tutorial, we checked `Visible` next to user profiles for which our console will be available. In this step, we'll assign the Service Cloud User feature license to agents with those profiles so that they can start using your console.

1. From Setup, click **Manage Users**.
2. Click **Edit** next to an agent's name.
3. Select `Service Cloud User`.



4. Click **Save**.

Repeat the steps above for each agent qualified for console access, and remember to grant yourself access to the console so that you can use it too!

Summary

Congratulations! By following a few steps, you set up the beginnings of a powerful help desk application for a call center.

In this tutorial, we showed you how to build a Salesforce console for support agents—and it probably took you less time than you imagined. In the next tutorial, we'll show you how you can turn customer emails into cases that agents can respond to from a console.

Tell Me More!

We've shown you the basics of how to set up a Salesforce console, but you can take a console to the next level by learning about the following features in the Salesforce online help:

- Custom console components—customize, extend, or integrate the footer, sidebars, or highlights panels, of a console using Visualforce. Visualforce uses a tag-based markup language to give developers a more powerful way to build applications and customize the Salesforce user interface.
- Salesforce console integration toolkit—write code that integrates the console with third-party systems or specifies custom functionality.
- Keyboard shortcuts—let agents perform actions more efficiently in a console by pressing a combination of keys instead of using mice. For example, instead of typing case details and then using a mouse to click **Save**, you can create a shortcut so that agents can type case details and press CTRL+S.
- Console whitelist—allow agents to access domains outside of Salesforce from within a console. For example, you can add `www.example.com`, which can be a third-party system, so that agents can access it without leaving the console.
- Interaction logs—let agents write notes on records that appear on primary tabs.

Tutorial #2: Turning Customer Emails into Cases Automatically

Level: Beginner

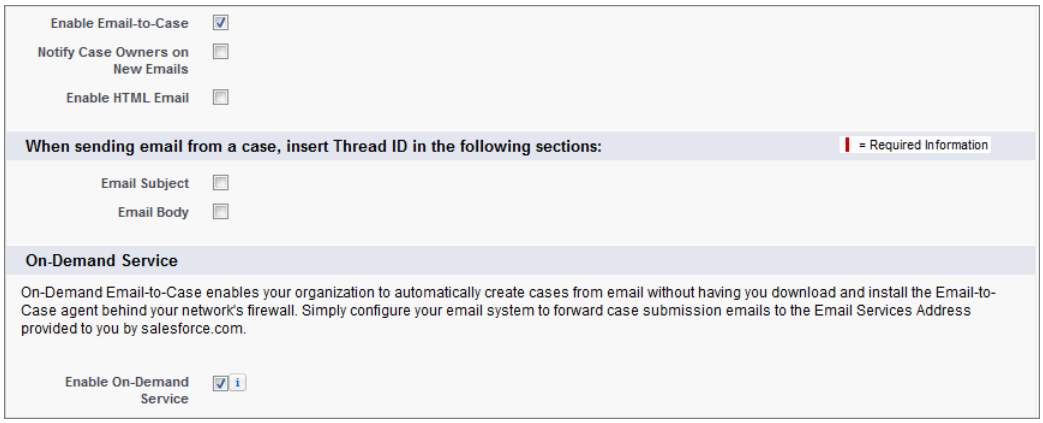
In the previous tutorial, we set up a Salesforce console so that your agents can review and respond to customer cases with lightning speed. Now, we'll start adding some channels to your console so that customers can contact your agents by their favorite method.

Suppose your customers want to ask questions or contact your service team by email. In this tutorial, we'll show you how you can use On-Demand Email-to-Case to turn a simple email address into a popular support channel. When we're finished, emails sent to a specific address will automatically convert to cases in Salesforce, which your agents can monitor from the console.

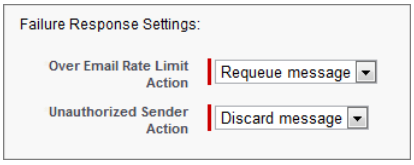
Step 1: Enable On-Demand Email-to-Case

Let's start by turning on On-Demand Email-to-Case and configuring a few of its settings. Note that this step assumes you've set the `Default Case Owner` and `Automated Case Owner` in Salesforce—these settings just make sure that each case is automatically assigned an owner and not lost in a technological void. If necessary, you can review these settings from Setup by clicking **Customize** > **Cases** > **Support Settings**.

1. From Setup, click **Customize** > **Cases** > **Email-to-Case**. If you see a page that introduces Email-to-Case, click **Continue**.
2. Click **Edit**.
3. Check `Enable Email-to-Case` and `Enable On-Demand Service`.



4. Choose what Salesforce does with emails that go beyond the daily processing limit, as well as what happens to emails from senders who are blocked from your organization. Let's select `Requeue message` so that any emails that surpass the daily limit are processed the following day, and let's select `Discard message` from unauthorized senders. After all, we don't want limits to prevent your agents from hearing from legitimate customers, but we don't want your agents to hear from the bad people you've blocked.



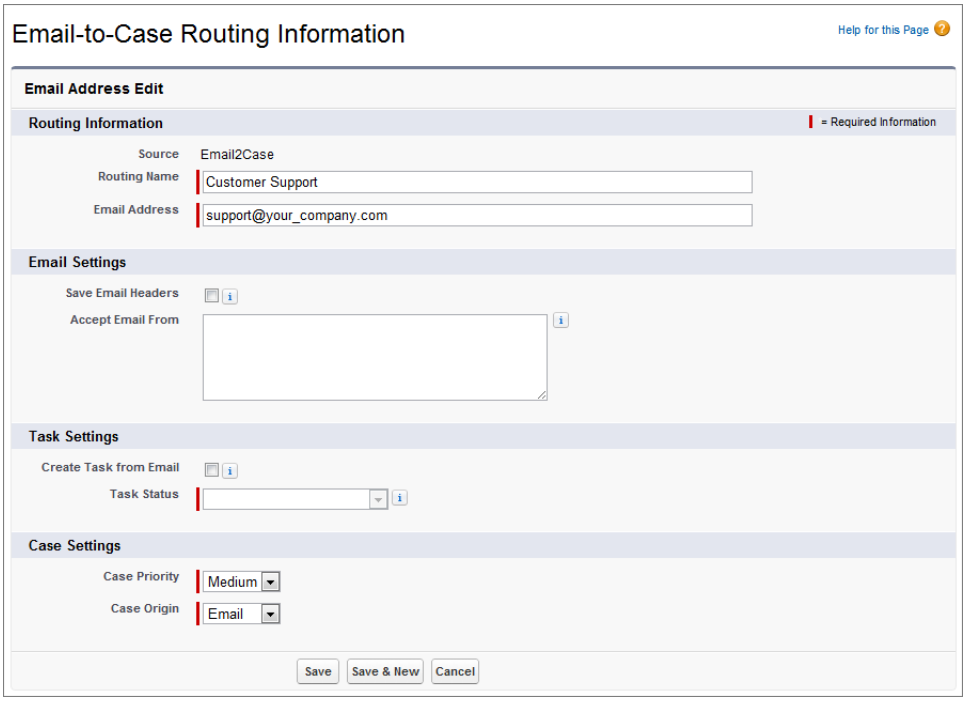
5. Click **Save**.

Wasn't that quick? We just turned on On-Demand Email-to-Case. Now, let's define some email addresses.

Step 2: Define and Verify Routing Addresses

We're ready to define an email address that will route any emails it receives to Salesforce and convert them to cases. Talk about instant and easy case gathering!

1. From Setup, click **Customize** > **Cases** > **Email-to-Case**.
2. In Routing Addresses, choose `Email2Case` and click **New**.
3. Type a `Routing Name` and `Email Address`. Let's name our routing address `Customer Support` and use an address authorized by your company as our email address. We're also going to set the `Priority` on cases created from emails to `Medium` and the `Case Origin` as `Email`.

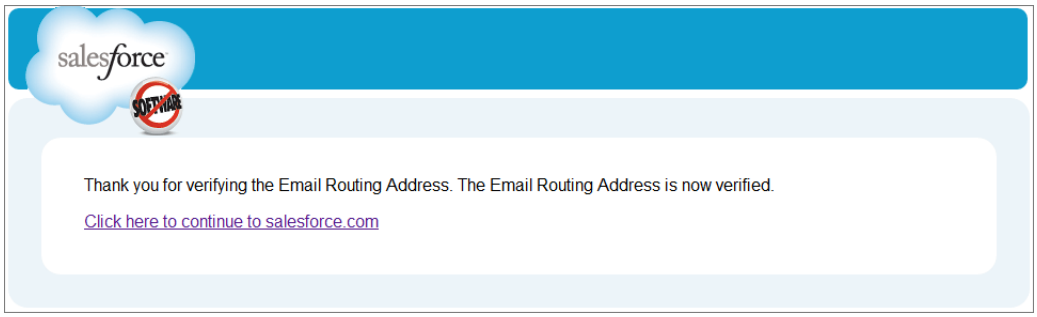


4. Click **Save**.

   A verification email is sent to the routing address you provided.

5. Click the link in the verification email.

   A confirmation page opens in your Web browser.



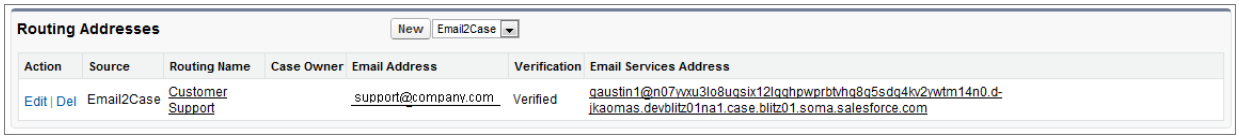
6. Click the link in the confirmation page to continue to Salesforce.

We've nearly finished adding one email address, but you can add many email addresses to support all kinds of things. For example, if your service team supports lots of products, you can create a separate email address for each product and use assignment rules to route the generated cases to the right agents.

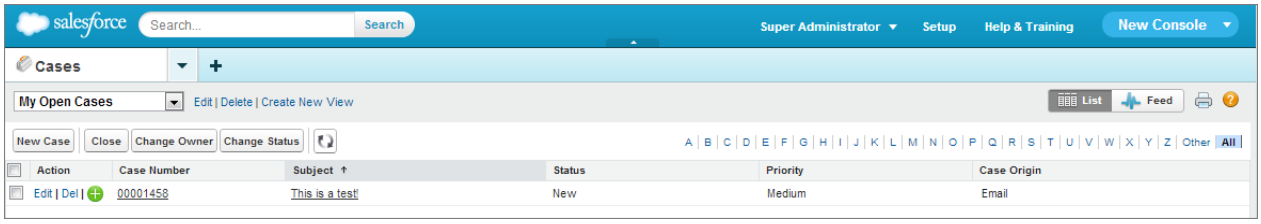
Step 3: Test and Add Email Addresses to a Website

We're nearly at the finish line for setting up On-Demand Email-to-Case. In this step, we're going to test that our routing address works, and then post it on a website so that your agents can receive cases from emails.

1. From Setup, click **Customize** > **Cases** > **Email-to-Case**.
2. In Routing Addresses, locate the verified Email Services Address.



3. Copy the verified email services address.
4. Paste the address into the `To` field of an email, and type some text into the email `Subject`, such as `This is a test`. Or, you could have some fun and type `Administrators are numero uno!`
5. Send the email.
6. From the Force.com app menu, select the console we created in Tutorial #1: Setting Up a Help Desk.
7. Select Cases from the navigation tab.
8. Verify that the email you sent has been converted to a case. Note that you might have to search for the case or update list criteria to find it.



9. Since you verified that emails sent to the email services address are converted to cases, you can send the address to your company's Web developer to add to your site.

Get ready to celebrate—we're done setting up On-Demand Email-to-Case! Wasn't that easy? One thing you can do is suggest to your Web developer that he or she create a custom alias for the email services address so that it appears branded for your company, and not Salesforce.

Summary

Woohoo! You've just added email as a support channel for your customers, and now your agents can use the Salesforce console to track, review, and respond to email inquiries. In the next tutorial, we'll show you how you can provide a Web channel to your customers.

Tell Me More!

We've shown you the basics of how to set up On-Demand Email-to-Case, but you can kick email support up a notch by learning about the following features in the Salesforce online help:

- Emails related list—add the Emails related list to case page layouts so that any cases generated by Email-to-Case include all emails and email threads between your customers and support agents.
- Email templates—customize and brand email templates so that your support agents can send consistent messaging to your customers.
- Email-to-Case agent—download and install the Email-to-Case agent to keep all email traffic within your network's firewall.
- Salesforce for Outlook—create cases in Salesforce from emails in Microsoft® Outlook®.

Tutorial #3: Providing Self-Service on the Web with Communities

Level: Intermediate

So far we've created a help desk for your service team, and we've created an email channel so that your customers can contact your agents by email. Now, we're going to set up a Web channel, or what we call a community, so that your customers can contact your agents from a website. Salesforce Communities give your customers another option to reach your team and each other.

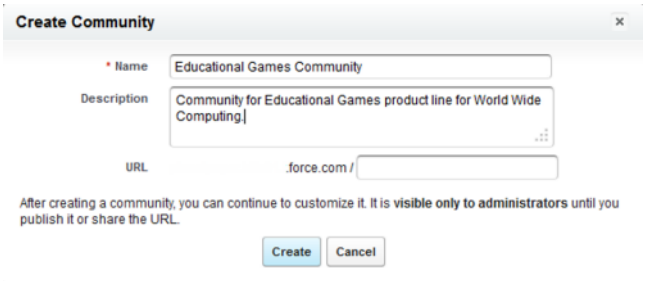
Suppose your customers want to find answers to their questions without contacting a support agent. Let's also suppose your management wants to reduce costs by finding ways customers can help themselves before contacting your service team. In this tutorial, we'll show you how you can do both. We'll setup a community with Chatter Answers so customers can find answers to their questions. Your customers can also post a private question to your agents who can turn the question into a case if there isn't an answer already. We'll also brand your community so that your customers know they are on a trusted site. Finally, we'll create a zone within the community where your customers can find information on specific product areas, collaborate with other users and experts, and more!

There are a lot of ways to set up communities. You can add awesome features to enhance customer engagement, such as Ideas, so that your customers can vote for new products or services. You can even add Salesforce Knowledge to your communities so that your customers can find and comment on articles from your knowledge base. But in this tutorial, we'll just get you started with a basic community that uses Chatter Answers. Let's begin!

Step 1: Create a Community

Let's start by creating a community. A Salesforce Community is a website where your customers and support agents can communicate publicly or privately. Think of it as an online forum where people help each other and get things done. Creating a community is easy.

1. From Setup, click **Customize** > **Communities** > **Settings**.
2. Select **Enable Communities**.
3. Type a unique domain name for your community, and click **Check Availability**. Since your company name is recognizable to your customers, let's enter it and assume it's available.
4. Click **Save**.
5. Click **OK** on the confirmation message to enable your community.
6. From Setup, click **Customize** > **Communities** > **Manage Communities**, then click **New Community**.
7. Name your community. Let's use a short name so your customers can remember it.
8. Accept the unique community URL, or type your own. This URL is added to the domain you created earlier for your community.



9. Click **Create**.

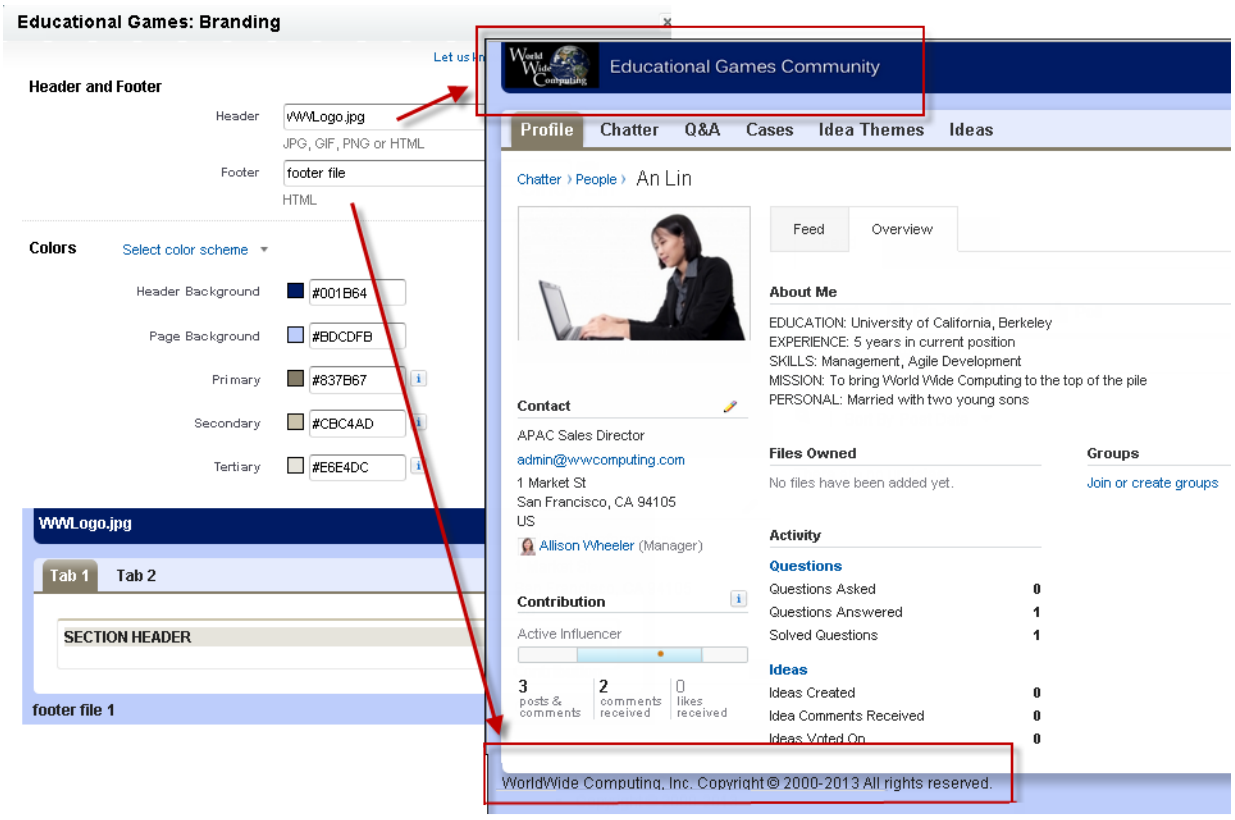
You've done it—you've created a community! Next, we'll configure how your community looks.

Step 2: Brand Your Community

Now, let's customize the look and feel of your community. You can make your community look like an extension of your official website so that your customers know that they're in a trusted area where they can post questions and get help. You can intricately customize your community's user interface, but we'll do it an easy way. Before we start, let's assume that we've uploaded these files to the Documents tab in Salesforce:

- A logo for the community landing page
- An email header file for messages sent to community members
- An email footer file for messages sent to community members

1. From Setup, click **Customize** > **Communities** > **Manage Communities**.
2. Click **Edit** next to the name of your community.
3. Click **Branding**.
4. In `Header` or `Footer`, click to choose an image.
5. Click **Select color scheme** and choose a predefined color scheme. If you wanted something fancier, you could click the textbox next to the page section fields to select specific colors.



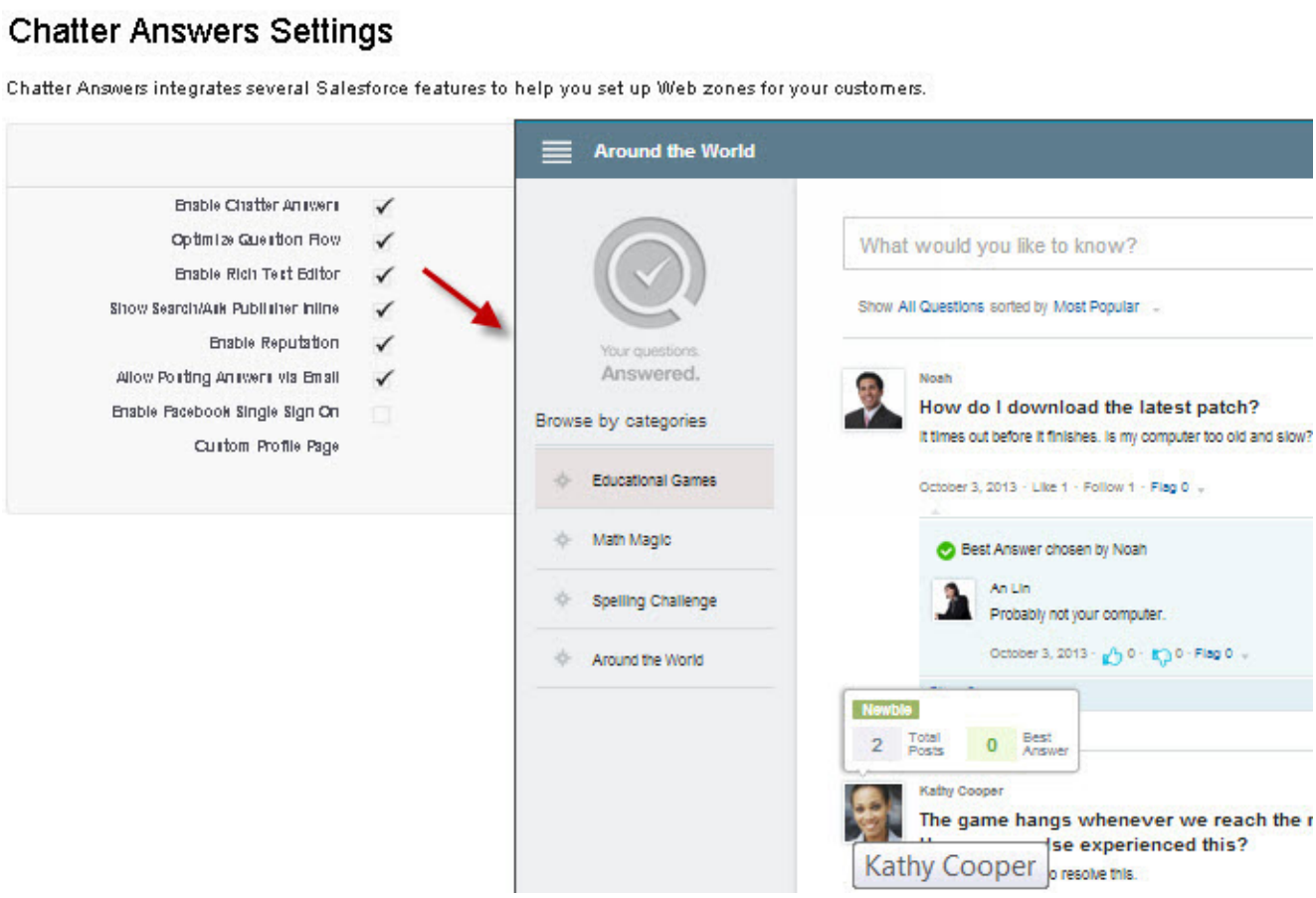
6. Click **Save**.

As you can see, the customization options are wide open. You can control the look and feel to match your branding and customer experience.

Step 3: Add Chatter Answers for Self-Service

In the previous steps you created and branded your Web community. Now, let's add Chatter Answers to your community so that your customers can help each other by posting questions and answers. Keep in mind that some of the options below might be turned on already due to your license type. Lucky you!

1. From Setup, click **Customize** > **Chatter Answers** > **Settings**.
2. Click **Edit**.
3. Click `Enable Chatter Answers`.
4. Let's select the following options to help out your customers:
   - `Optimize Question Flow` so that your customers can easily search for an answer before posting a question.
   - `Enable Rich Text Editor` to let community members format text and upload images when posting questions and answers.
   - Select `Show Search/Ask Publisher Inline` to embed the search/ask user interface inline instead of in a popup window.
   - `Enable Reputation` to let community members earn points and ratings that display on their profile pictures.
   - `Allow Posting Answers via Email` to let members post answers by replying to email notifications.



5. Click **Save**.

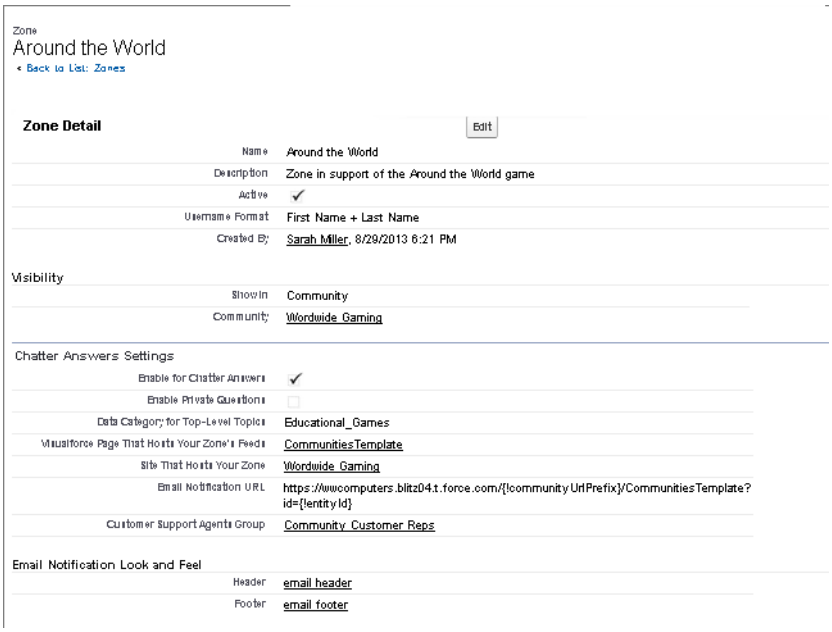
Okay, now that we've enabled Chatter Answers, let's move on to creating a zone!

Step 4: Create a Zone for Your Community

So far, we've created your Web community, branded it, and turned it into a channel where customers can help themselves and others. Now, we're going to create zones. Zones might sound funny, but they organize questions posted to your community into logical groups with their own focus. For example, you could create communities for each of your product lines and separate zones for each product. Before we create a zone, let's assume you've created a Salesforce user group with support agents. We'll add that group of agents to the zone after we create it. We'll also assume that you have some data categories set up. This makes the steps below even easier, especially since we're going to skip a lot of optional settings.

1. From Setup, click **Customize** > **Chatter Answers** > **Zones**.
2. Click **New**. By default, a zone already exists, but it's for your internal users. We want to focus on your customers.
3. Type a name for your zone.
4. Click `Active` to display the zone in the community.
5. Choose a `Username Format` to determine how community members' names appear throughout the zone in questions and answers.
6. In `Show in`, select Community and then your community's name.
7. Click `Enable for Chatter Answers`.
8. Let's click `Enable Private Questions` so that your customers can post their questions privately to your support agents. Your agents will see these questions as cases in Salesforce.
9. Select a data category in `Data Category for Top-Level Topics`.
10. In `Customer Support Agents Group`, select the Salesforce user group that includes the support agents for your community.
11. In `Header` or `Footer`, click to choose files that include any branding for email notifications sent from the zone. We assumed that you already uploaded these publicly accessible files to the Documents tab in Step 2: Brand Your Community
12. Click **Save**.

Now that your community has a zone, let's give your community some tabs.

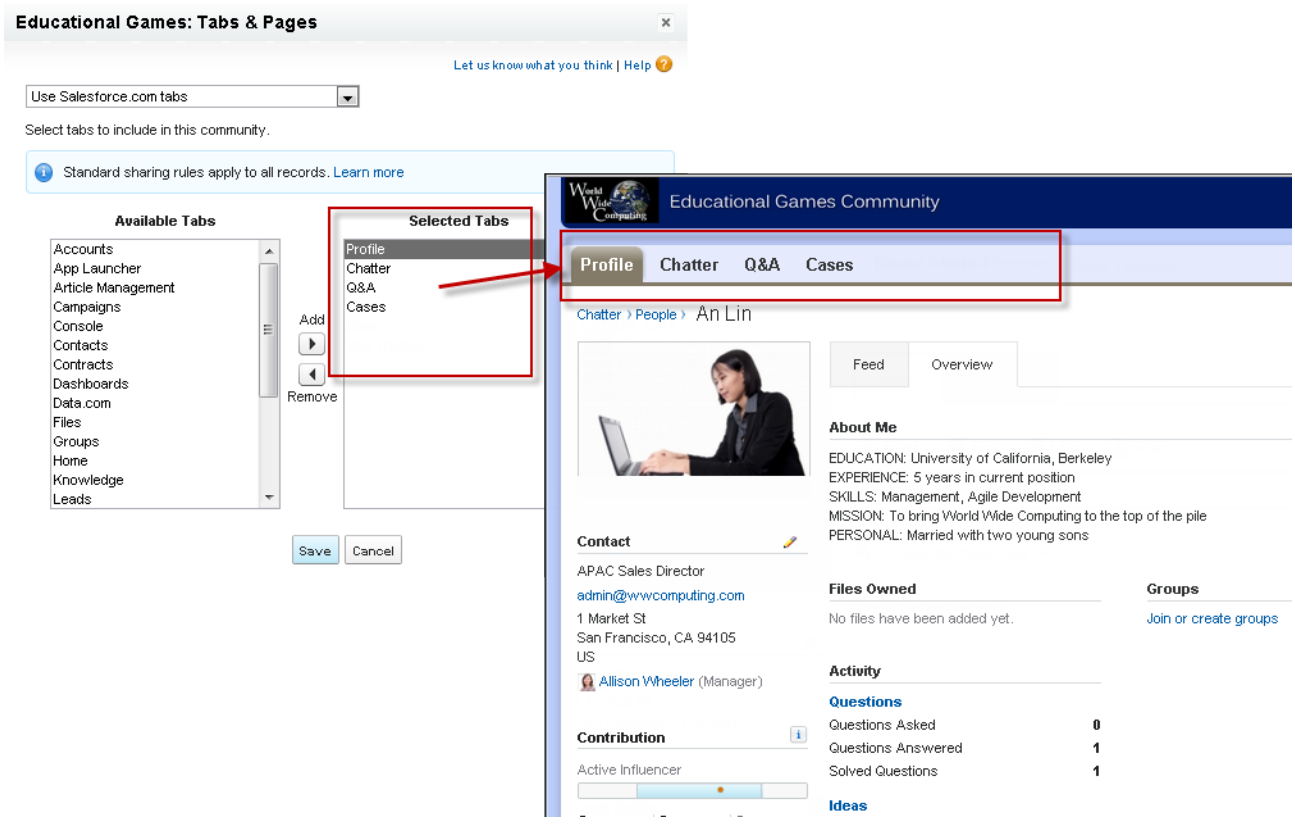


Step 5: Customize Community Tabs

Let's choose which tabs to display in your community. Note that the tabs community members can see is based on user profiles. The tabs that we're going to display will provide customers with what they need for self-service on the Web.

1. Click **Customize** > **Communities** > **Manage Communities**, then click **Edit** next to your community name.
2. Click **Tabs & Pages**.
3. Select `Use Salesforce.com tabs`—we're not going to create Site.com pages.
4. Add the following to Selected Tabs:

   - Profiles—lets members view all of their open questions in one place, add a picture to their profile, and edit their privacy settings.
   - Chatter—lets members view posts and comments in Chatter.
   - Q&A—lets members ask questions and receive answers from the community.
   - Cases—lets members work with support agents on cases.



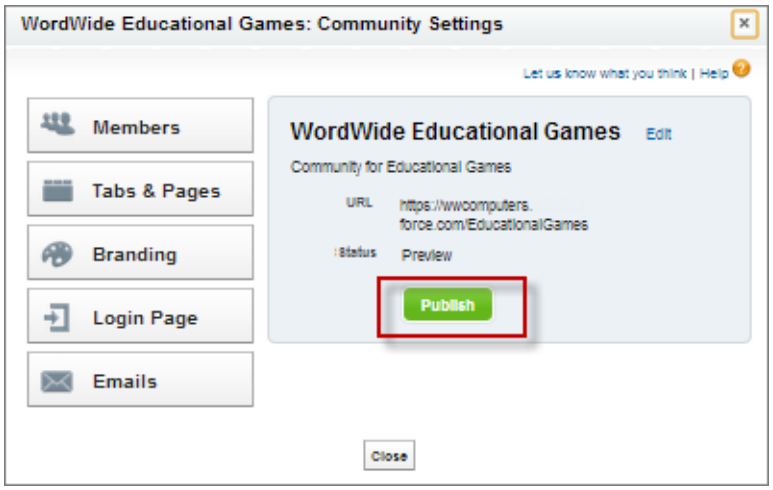
5. Click **Save**.

We've added the content to display in your community. By adding the Cases, Profile, and Q&A tabs, we've created a resource where your customers can help themselves and each other, thereby reducing support calls and increasing customer satisfaction. Remember that you can add all kinds of tabs and features to your community. For example, if you set up Ideas , you can add its tab to your community so members can vote on product or service ideas or make suggestions.

Step 6: Publish Your Community

We're almost done! We've created a community, branded it, given it Chatter Answers, "zoned" it, and determined its tabs. Now, let's publish the community so your customers can access it.

1. From Setup, click **Customize** > **Communities** > **Manage Communities** and click **Edit** next to your community.
2. Click `Publish`.

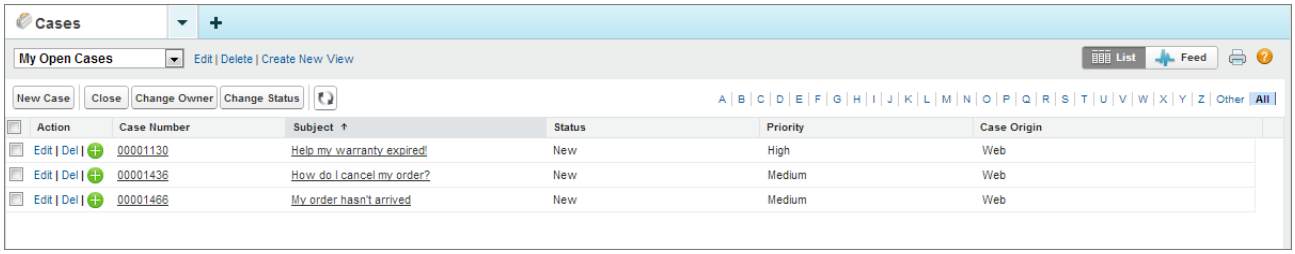


Congratulations! You've just made a self-service Web channel available to your customers. Even though you've published your community, you can continue to customize it while it's live, or you can take it offline to make changes.

Step 7: View Cases from the Community in the Help Desk

In the previous step, we published your community. Let's say that a few of your customers couldn't find answers to their questions in the community, so they logged some cases. In this step, we're going to see how your agents can view cases created over the Web in the console we created in Tutorial #1: Setting Up a Help Desk. To keep things simple, let's assume that you've created a workflow rule that routes cases created on in your community to you, and sets `Case Origin` to Web.

1. Click **New Console** from the Force.com app menu.
2. Click **Cases** on the navigation tab.
3. Notice that the My Open Cases view lists `Web` as the `Case Origin` for your cases—these are cases that came in from your community.



Now, your agents can easily find and work on cases that come in from the Web, without leaving the console.

Summary

Amazing! In just a few steps, you've set up a Web community so that you can provide self-service to your customers, and your support agents can use the console to identify and respond to questions from the Web.

We've covered the basics of communities, and now you know how to create them for your customers.

Tell Me More!

We've shown you the basics of how to set up a community, but you can take your community to new heights by learning about the following features in the Salesforce online help:

- Multiple communities—create more than one community for your business partners, customers, employees, you name it.
- Self-registration for new community members—let customers self-register for your community.
- Replies to questions and comments with email—let your community members reply directly from email when they receive email notifications related to a question. The replies appear as an answer in the community.
- Salesforce Knowledge—add a knowledge base to your community so that your customers and partners can find the knowledge articles they need.
- Entitlement Management—add entitlements and service contracts to your community so that customers and agents can verify service agreements and terms of service.
- Ideas—give your community members an online suggestion box where they can post, vote, and comment on ideas.
- Idea Themes—create contests for community members to vote on.
- Workflow—set up workflow rules or case assignment rules so that any cases created by community members are routed to the right agents or support queue.
- Web-to-Case—if you don't want to use the case-deflection tools of a community, you can create a form to post on your company's website where customers can log cases directly into Salesforce.

Tutorial #4: Integrating Phone Calls

Level: Advanced

Let's take a look at what we've accomplished so far. Not only does your service team have a unified help desk application, but now your customers can log cases by email, search for solutions through a Web community, and generate cases directly from the Web. But what about phone calls? Now, we're going to set up a call channel, or what we call computer-telephony integration (CTI), so that your customers can contact an agent from a phone.

Let's say your customers want to phone a support agent for immediate help. In this tutorial, we'll show you how you can add a SoftPhone, otherwise known as a call-control tool, to the footer of the Salesforce console so that agents can answer phone calls and update customer info in Salesforce while speaking with customers. But to keep things short and snappy, we're going to create a SoftPhone with very limited capabilities. By the end of this tutorial you'll have created a SoftPhone that lets agents make phone numbers clickable in Salesforce, and screen pop a contact record. Later, we'll point you to some resources that will help build more robust call tools.

Some background on CTI: Salesforce and its partners offer many options to choose from, including Open CTI, which gives you the benefits of cloud architecture and less maintenance. But Open CTI requires developers to use JavaScript to embed API calls. This might sound scary, but our partners do most of the work for you. Typically, an Open CTI installation is as simple as:

1. You install a CTI package created by a Salesforce partner from the AppExchange: Salesforce's online marketplace for apps.
2. When you install the package, it creates a call center for your organization: a third-party CTI system that integrates with Salesforce.
3. You add users to the call center so that they can make and receive calls with a SoftPhone in Salesforce.

That's the easy way to integrate phone calls with a help desk; and we recommend the easy way for non-developers. But if you want to experiment with some code and "dial in" to the details of creating a SoftPhone with Open CTI, Visualforce, and JavaScript, then this tutorial is for you. To learn more about Call Center and Open CTI before we begin, check out "Call Center Overview" and "Salesforce Open CTI Overview" in the Salesforce online help. Okay, let's get started!

Step 1: Create an Open CTI SoftPhone

Let's put on our developer hat. We're going to write some code and we're going to write it in Visualforce. Why? Because Visualforce is Salesforce's tag-based markup language, and it comes with a rich component library that lets you quickly build Salesforce pages without having to do a lot of extra work, such as host pages on an external server. You can write code for Open CTI in other languages, and host pages related to Open CTI on servers that have little to do with Salesforce; but we want to keep things simple. In the spirit of simplicity, we're going to assume that you're a developer and have already enabled `Development Mode` for yourself so that you can create Visualforce pages. To learn more about Visualforce, check out "Visualforce Overview" in the Salesforce online help.

1. Create a Visualforce page by clicking **Setup** > **Develop** > **Pages** > **New**.
2. Type a label and name for the page, such as `softphone`.
3. Copy/paste the code below into your page. This code determines the SoftPhone's functions and look and feel. To guide you through what the code does, it includes comments, which begin with `/*` or `//`.

```
<apex:page showHeader="false">
<apex:includeScript value="/support/api/29.0/interaction.js/>
<style>
    body {background-color:white;}
</style>
<script type="text/javascript">
    var isClickToDialEnabled = false; /* Variable that keeps track of click-to-dial
```

```
state.
                                        * If true, click-to-dial is enabled, false
otherwise.
                                        */
    // Callback used with enableClickToDial API method.
    var enableClickToDialCallback = function (response) {
        isClickToDialEnabled = true;
    };

    // Callback used with disableClickToDial API method.
    var disableClickToDialCallback = function (response) {
        isClickToDialEnabled = false;
    };

    /* This function toggles the click-to-dial feature.
     * It enables the feature if it was disabled or
     * disables it if it was enabled.
     */
     function toggleClickToDial() {
        if (isClickToDialEnabled) {
           // This function allows phone elements to be clickable on a Salesforce page.

            sforce.interaction.cti.disableClickToDial(disableClickToDialCallback);
        } else {
            // Enable click-to-dial.
            sforce.interaction.cti.enableClickToDial(enableClickToDialCallback);
        }
    }

    // Callback for screenPop API method.
    var screenPopCallback = function (response) {
        if (response.result) {
            alert('Screen pop was set successfully.');
        } else {
            alert('Screen pop failed.' + result.error);
        }
    };

    /* This function will screenPop to a contact in your organization.
     * When working on this sample code, please replace the contact Id with
     * a valid one from your organization.
     */
    function screenPop() {
       var objectIdRelUrl = '/003D000000PS4iL'; // Replace the Id with a valid one from
 your organization.
        sforce.interaction.screenPop(objectIdRelUrl, true, screenPopCallback);
    }
    // Callback for onClickToDial API method.
    var onClickListener = function (response) {
        if (response.result) {
            alert("User clicked on a phone number. The data returned as JSON format is
: " + response.result );
        }
    };

    // Registers a callback function that will execute when a user clicks on a phone
link.
    sforce.interaction.cti.onClickToDial(onClickListener);
</script>
    <button onclick='toggleClickToDial();'>Toggle Click-to-Dial</button>
    <button onclick='screenPop();'>Screen Pop</button>
</apex:page>
```

4. Click **Save**.

We can almost hear the ringtones—we've finished the first step towards providing your customers with a phone channel. Keep in mind that we created a SoftPhone with very limited capabilities. By copying/pasting some code, we built a SoftPhone that

lets agents make phone numbers clickable in Salesforce, and click a button to screen pop a contact record. If you want to add more bells and whistles to your SoftPhone, visit the *Open CTI Developer's Guide* for some code samples and ideas.

Step 2: Create a Call Center

In the previous step, you created a SoftPhone, but it's useless until you hook it up to a call center definition file. A call center definition file just tells Salesforce what your call system can do. Typically, the file is a few lines of code in an XML file; partners create the file for any call centers created with Open CTI on the AppExchange. But since we're creating a phone channel from scratch, let's assume you've already created a call center definition file named `callcenter.xml` and saved it to your desktop. Here's what it looks like:

```
<callCenter>
    <section sortOrder="0" name="reqGeneralInfo" label="General Information">
     <item sortOrder="0" name="reqInternalName" label="InternalName">DemoOpenCTI</item>
     <item sortOrder="1" name="reqDisplayName" label="Display Name">Demo Call Center Open
 CTI</item>
     <item sortOrder="2" name="reqAdapterUrl" label="CTI Adapter
URL">https://local/softphone</item>
     <item sortOrder="3" name="reqUseApi" label="Use CTI API">true</item>
     <item sortOrder="4" name="reqSoftphoneHeight" label="Softphone Height">300</item>
     <item sortOrder="5" name="reqSoftphoneWidth" label="Softphone Width">500</item>
    </section>
</callCenter>
```

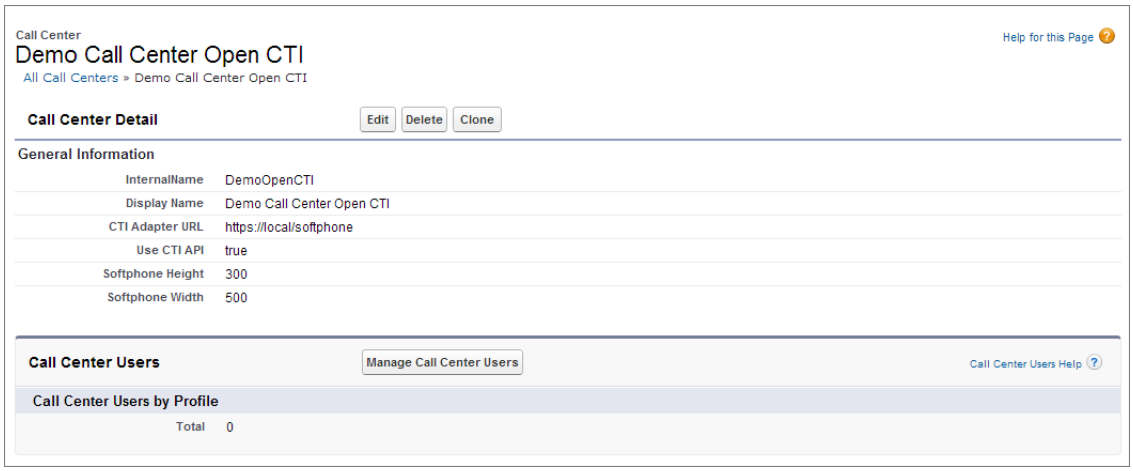
In this step, we'll import the call center definition file into Salesforce so that our SoftPhone can "talk" with Salesforce.

1. From Setup, click **Customize** > **Call Center** > **Call Centers**.

   If an Introducing Salesforce CRM Call Center page displays, click **Continue**.

2. Click **Import**.
3. Click **Choose File** to navigate to the call center definition file on your desktop.
4. Click **Open** to enter the path in the `Call Center Definition File` field.
5. Click **Import**.

As you can see, we've created your call center and you can now view its settings.

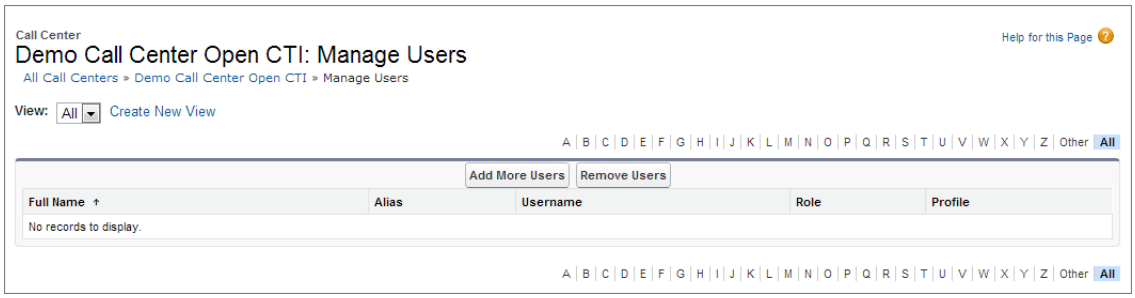


Next, we'll assign agents to your call center so that they can use it.

Step 3: Add Users to a Call Center

Before your service team can access a SoftPhone, we must assign them to the call center we created. This will take less time than dialing your cell-phone provider, and it won't be anywhere near as painful.

1. From Setup, click **Customize** > **Call Center** > **Call Centers**.
2. Click the name of the call center we created.
3. In the Call Center Users related list, click **Manage Call Center Users**.
4. Click **Add More Users**.



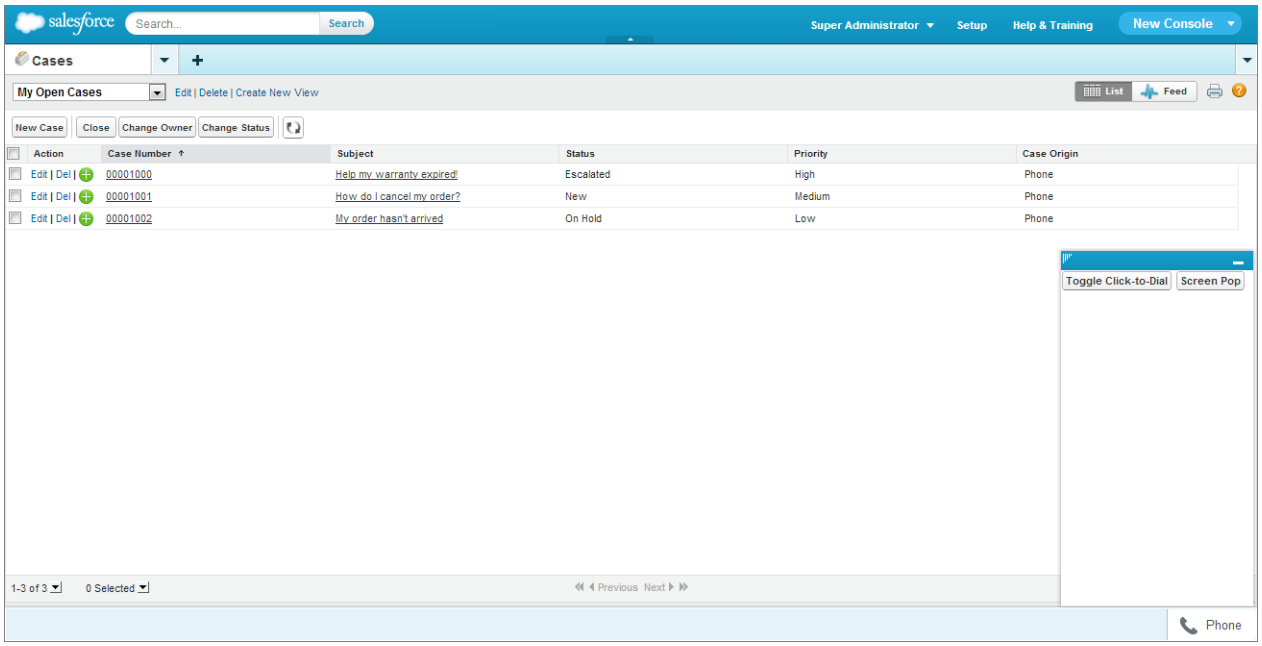
5. Specify search criteria to find agents who should be assigned to the call center. Since you're an administrator who wants to use a SoftPhone, let's enter the criteria of `Profile Contains Administrator`.
6. Click **Find**.
7. Check the box next to your name, and click **Add to Call Center**.

Bingo! We've assigned you and some of your lucky agents to a call center. Now, you and your service team can use a SoftPhone from the Salesforce console we created in the first tutorial. By completing three simple steps, you've added a call channel to your help desk, which lets your agents multi-task over multiple channels.

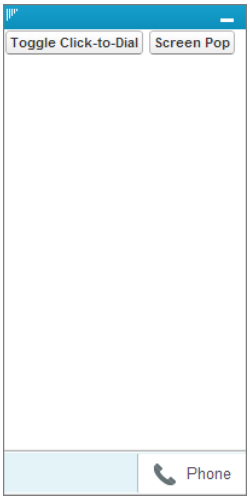
Step 4: View the SoftPhone in the Help Desk

So far we've created a SoftPhone, defined a call center, and assigned your service team to the call center. In this step, we'll use the SoftPhone to make calls by clicking phone numbers in the console we created in Tutorial #1: Setting Up a Help Desk. Are you ready to click-to-dial?

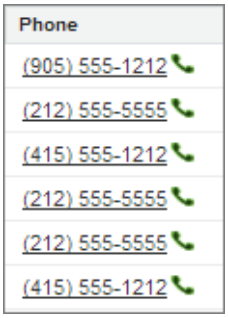
1. Go to the console we created by clicking **New Console** from the Force.com app menu.
2. Click **Phone** in the footer of the console.



3. Click **Toggle Click-to-Dial**.



4. Go to any record or list view with phone numbers. Notice that the phone numbers are links that you can click to place a call.



There you have it—we've set up a call system that integrates with your console. Now, your service team can make and receive calls while helping other customers on email and Web channels. Your agents' productivity is now officially in the zone! Take a moment to high-five someone.

Summary

You've just completed another exercise that gives your agents the power of phenomenal multi-tasking. Now your service team can handle email, Web, and phone inquiries from a single screen. If we managed your company, we'd give you a raise or at least refurnish your cubicle.

We just showed you how to set up computer-telephony integration (CTI) so that you can make your customers happy by providing them with immediate phone support. In the next tutorial, we'll make your customers even happier by giving them text-only live chat with your agents.

Tell Me More!

We've shown you the basics of how to set up computer-telephony integration (CTI) that works with the console, but you can take CTI to greater heights by learning about the following features in the Salesforce online help:

- Call center directories—customize directories that let users search for phone numbers throughout your organization.
- Salesforce console integration toolkit—write code to integrate call systems with Salesforce or leverage call data for other uses.

If you're a developer, take a look at the options available through Open CTI and Salesforce Console APIs, located in our developer's guides:

- *Open CTI Developer's Guide*
- *Salesforce Console Integration Toolkit Developer's Guide*

Tutorial #5: Chat Live with Customers on the Web

Level: Advanced

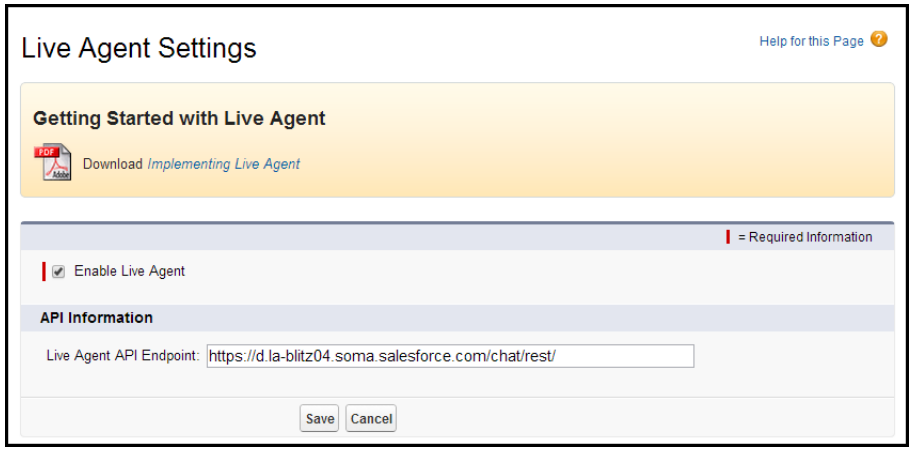
Service is about giving your customers choices. As an administrator, you want to provide those choices, but without burdening your agents. So far, you've seen how the console makes it easy for agents to handle multiple channels simultaneously. In this tutorial, we'll add another channel: chat. Lots of customers want the satisfaction of instant communication, without having to send an email, log a case on the Web, or phone someone they don't know. With text chat—or what we call Live Agent—you can give your customers another choice in how to contact your service team.

In this tutorial, we're going to set up a Web-based chat, and add it to the footer of the console so that your agents can offer immediate help.

Step 1: Enable Live Agent

Let's start by enabling Live Agent.

1. From Setup, click **Customize** > **Live Agent** > **Settings**.
2. Select `Enable Live Agent`.



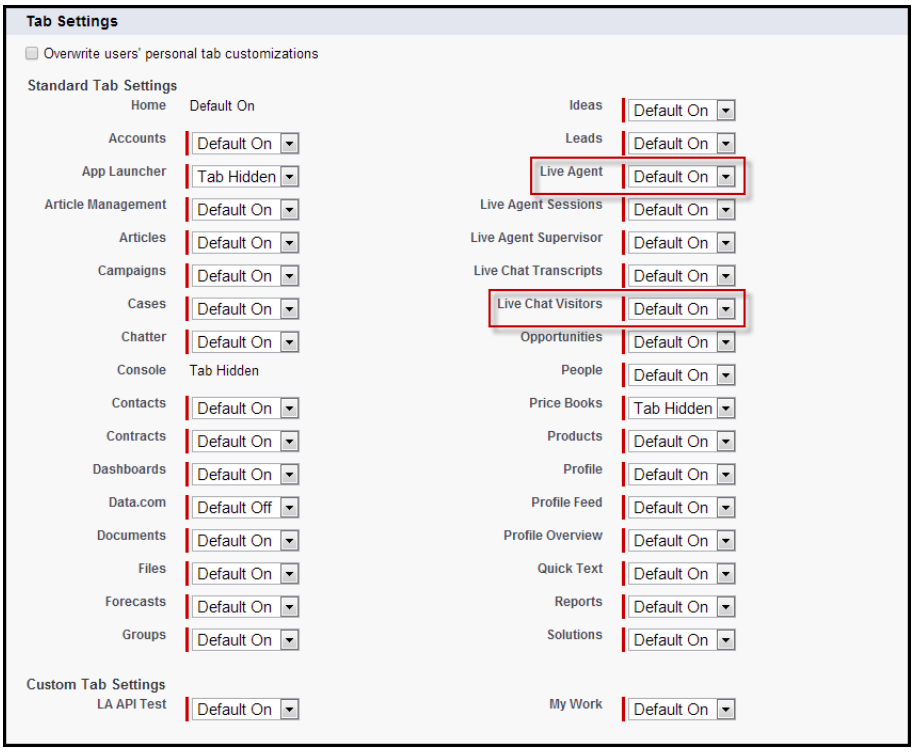
3. Click **Save**.

Now, we're ready to configure your service team to use Live Agent.

Step 2: Configure Live Agent Users

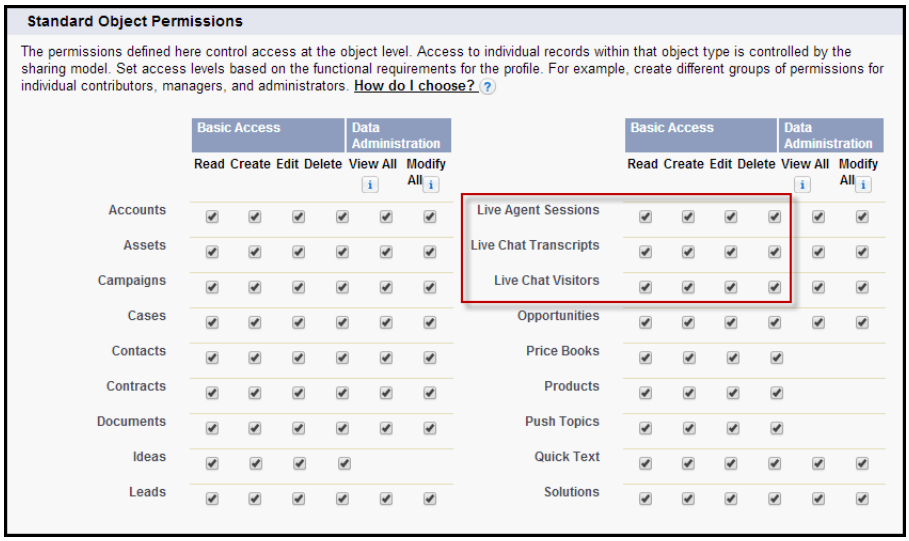
In this step, we'll give your agents access to Live Agent. That means we'll update a few permissions on agents' profiles and assign agents the Live Agent User feature license. In this highly customizable world, you can give agents permissions to all kinds of Live Agent functionality, but to keep things simple, let's give your agents access to basic features, such as live chat visitors, which lets your agents receive and view chat requests from customers visiting your company's website.

1. From Setup, click **Manage Users** > **Profiles**.
2. Click **Edit** next to an agent's profile. For this tutorial, let's edit your profile.
3. In Tab Settings, choose Default On for `Live Agent` and `Live Chat Visitors`.

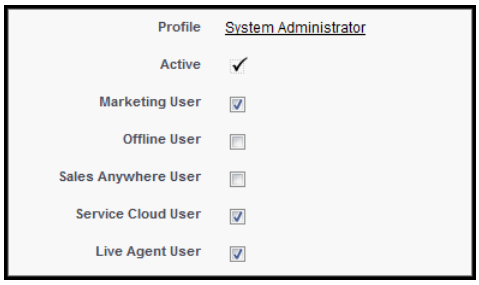


4. In Standard Object Permissions, choose the "Read," Create," "Edit," and "Delete" permissions for the following objects:

   - `Live Agent Sessions`
   - `Live Chat Transcripts`
   - `Live Chat Visitors`



5. Click **Save**.
6. From Setup, click **Manage Users** > **Users**.
7. Click **Edit** next to an agent's name. For this tutorial, let's edit your name.
8. Select **Live Agent User**.



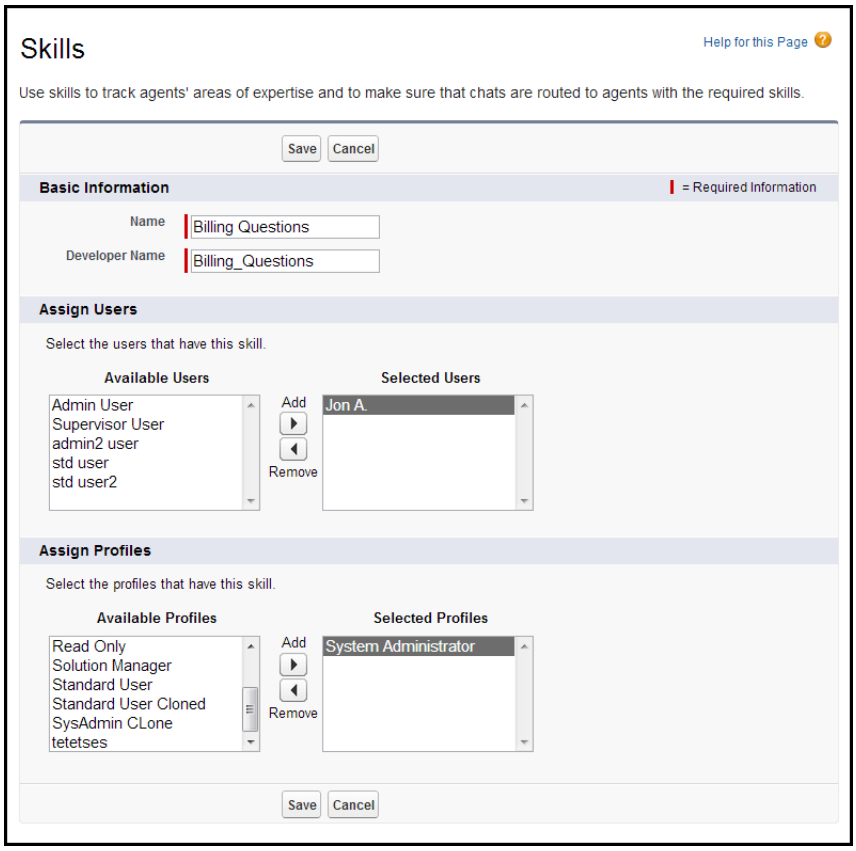
9. Click **Save**.

We've just given you and your agents access to Live Agent. Soon, the online chatting will begin, and the customer satisfaction won't end.

Step 3: Create Agent Skills and Assign Them to Users

Now, we're going to wow you by setting up a super useful Live Agent feaure—skills! Skills identify experts in specific areas and route chats to agents who are qualified to answer certain questions. For example, if you have an agent who is a genius at answering billing questions, you can create a skill named Billing Questions and route all chats related to billing to that agent. As you can probably tell, skills help your service team work more efficiently by making sure the right chat requests automatically go to the right agents. In this, step, we're going to create and assign skills.

1. From Setup, click **Customize** > **Live Agent** > **Skills**.
2. Click **New**.
3. Type a name for the skill. Let's type `Billing Questions`.
4. In Assign Users, select the agents you want to associate with the skill.
5. In Assign Profiles, select the agent profiles you want to associate with the skill.



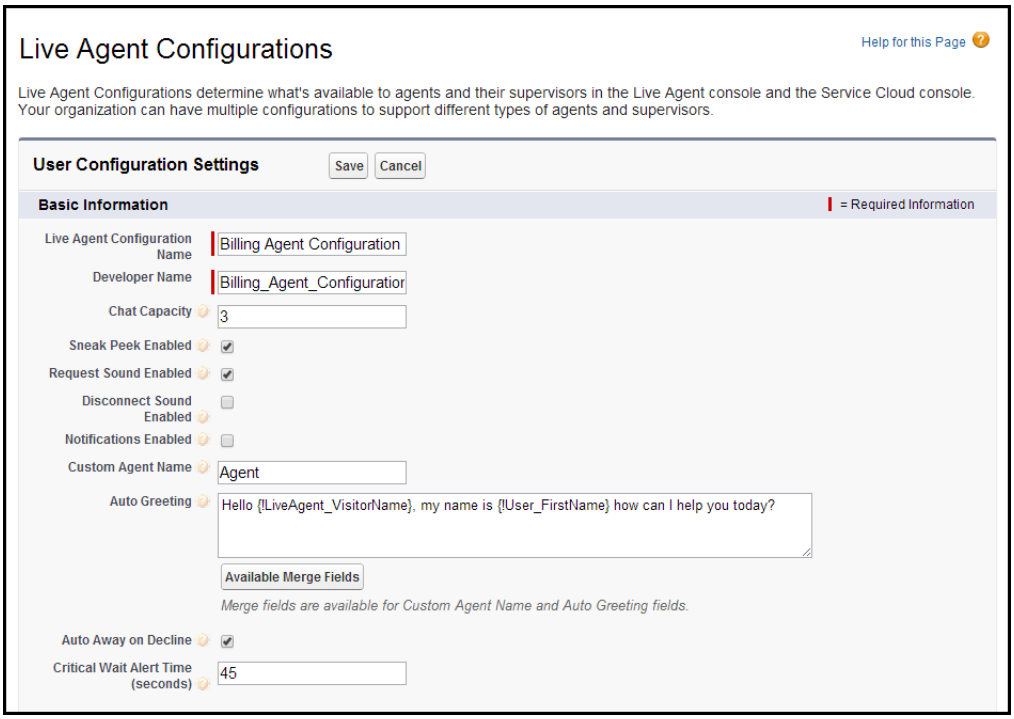
6. Click **Save**.

Awesome! Now that we've set up our users and their skills, let's configure how Live Agent will work for these users in the console.

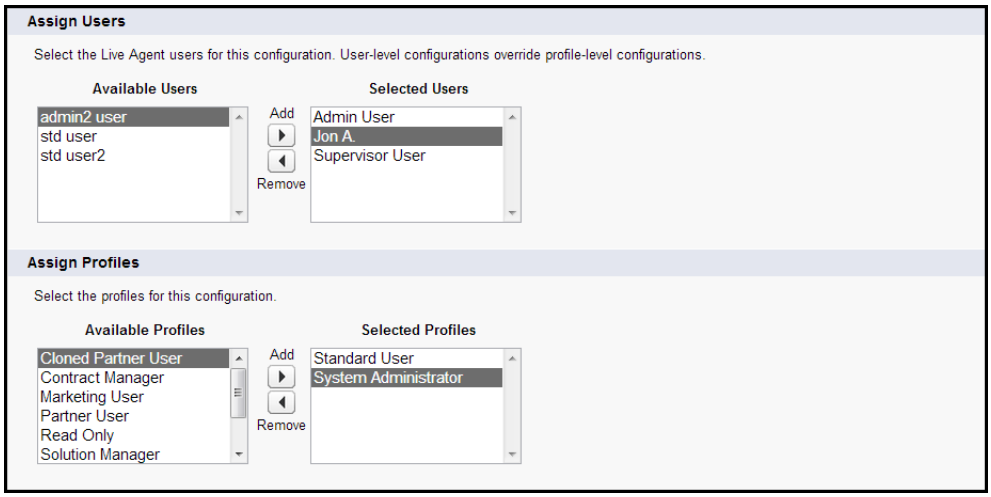
Step 4: Set Up Live Agent Configurations

A Live Agent configuration determines how chat works in the console. Configurations control everything from the notifications agents hear when they receive a new chat to the level of management supervisors can provide agents during a chat session. In this step, we'll set these configurations.

1. From Setup, click **Customize** > **Live Agent** > **Live Agent Configurations**.
2. Click **New**.
3. Name the configuration. Let's type `Billing Agent Configuration`.
4. Enable the various options you want your agent to have when they chat with customers. In this example, let's select the following:
   - `Enable Sneak Peek`—lets agents view customer messages before they send them.
   - `Request Sound Enabled`—plays a sound to alert the agent that they've been assigned a new chat request.
5. Type the name you want to appear to customers when your agents chat with them. In this example, let's keep things simple and type `Agent`.
6. Enter a greeting message. This message will be sent to customers when an agent starts a chat.
7. Let's select `Auto Away on Decline` to automatically change agents' status to Away when they decline chat requests.
8. In `Critical Wait Alert Time`, enter the number of seconds an agent has to respond to a customer's last message. If the agent doesn't respond within that time, they'll receive an alert. For our example, let's set this to `45` seconds.



9. In Assign Users, select the agents you want to associate with the configuration.
10. In Assign Profiles, select the agent profiles you want to associate with the configuration.



11. In the Supervisor Settings area, select the following functions for your supervisors:

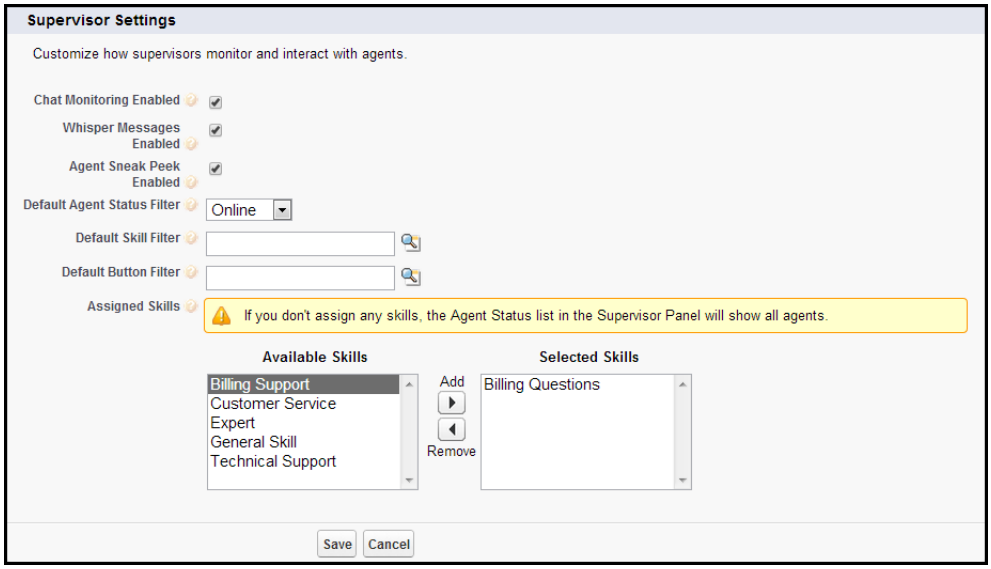
- Chat monitoring—lets supervisors monitor agents' chats with customers in real time.
- Whisper messages—lets supervisors send private messages to agents during chats.
- Agent sneak peek—lets supervisors view agent messages before they send them.

12. Select the default criteria for filtering the Agent Status list in the Supervisor Panel. The Supervisor Panel acts as a command center for supervisors that allows them to monitor their agents' activities as they chat with customers.

- Agent status
- Default skill
- Default button
- Agent skills

Filters make it easier for your supervisors to monitor their agents' activities as they chat with customers. In this example, let's set the following preferences:

a. Set the Default Agent Status Filter to `Online`.
b. In the Assigned Skills list, select the `Billing Questions` skill we created earlier.



13. Click **Save**.

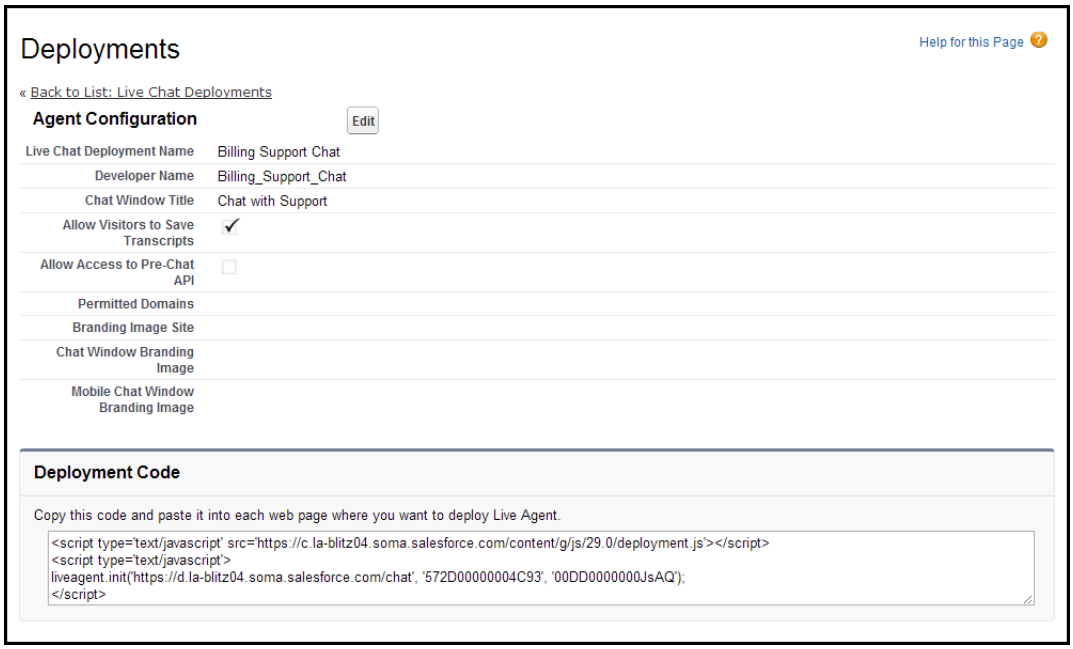
We've finished setting up Live Agent for your service team. Now, we're ready to present Live Agent to the world.

Step 5: Create Chat Deployments

A deployment is a place on your company's website that's enabled for Live Agent. In this step, we'll generate the code to include on your website to make Live Agent visible. After all, why use a chat application if your customers can't find it?

1. From Setup, click **Customize** > **Live Agent** > **Deployments**.
2. Click **New**.
3. Type a name for the deployment. Let's type `Billing Support Chat`.
4. Type a title for the Chat window. This is the title that visitors see. Let's type `Chat with Support`.
5. Select `Allow Visitors to Save Transcripts`.
6. Note that you can brand your chat window to control the look and feel of your customers' chat experience. In this tutorial, let's skip this and click **Save**.

   Salesforce generates the deployment code.



7. Copy/paste the automatically generated code and send it to your company's Web developer to add to your site.

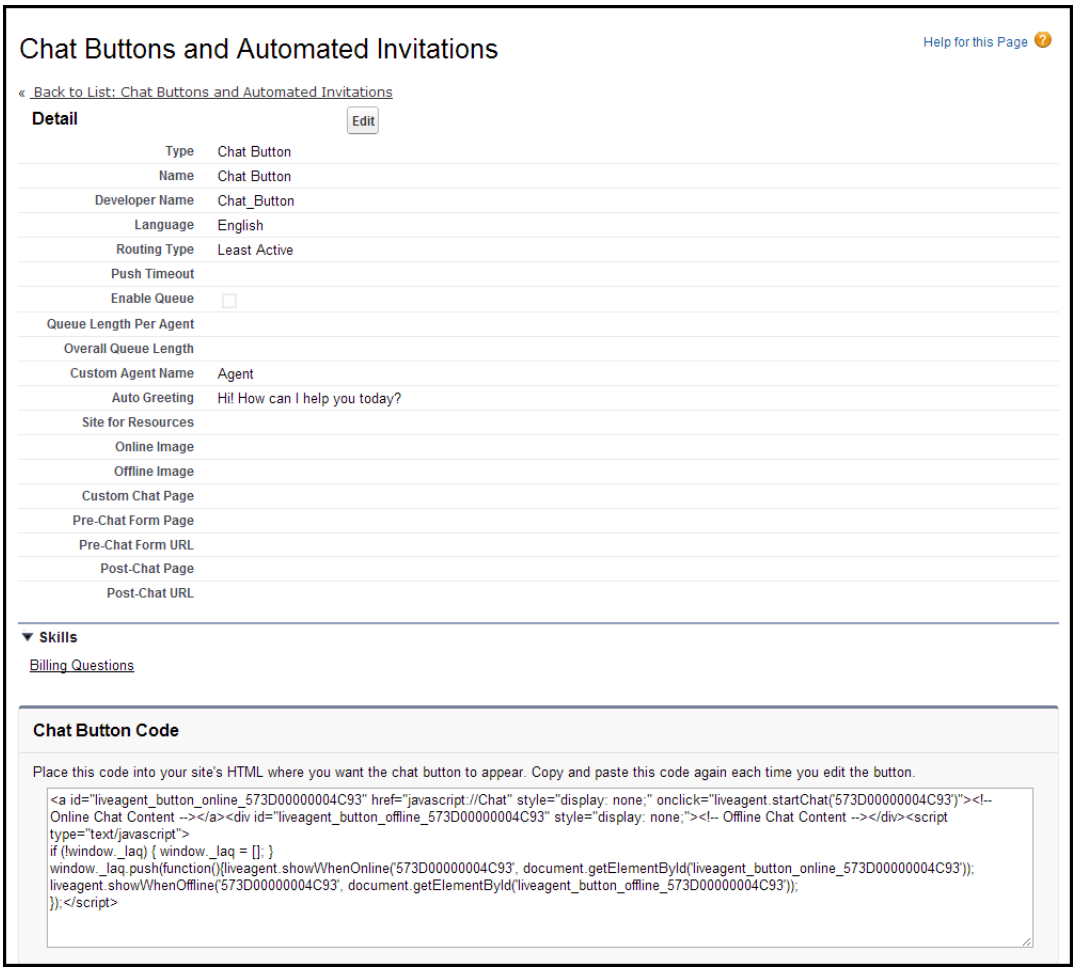
Now we have code that we can add to our website, but we still need to create the custom buttons that customers click to initiate chat sessions. We'll do that next.

Step 6: Create Chat Buttons

A chat button is used with a Live Agent deployment to initiate chat sessions. In this step, we'll create a chat button to include on your website so that customers can chat with your agents.

1. From Setup, click **Customize** > **Live Agent** > **Chat Buttons & Invitations**.
2. Click **New**.
3. Select Chat Button from the `Type` drop-down list.
4. Type a name for the button. Let's type `Chat Button`.
5. Select how chats initiated from this button are routed to support agents. Let's choose `Least Active`, which means that incoming chats are routed to the agent who has the fewest active chats.
6. Select a skill to associate with the button by moving it from the `Available Skills` list to the `Selected Skills` list. Let's select the `Billing Questions` skill we created earlier.
7. Select `English` as the language for text in the chat window.
8. Note that you can brand your chat button to control the look and feel of your customers' chat experience—maybe add an image of some pleasant clouds to the button. In this tutorial, we'll skip this and click **Save**.

   Salesforce generates the chat button code.



9. Copy/paste the automatically generated button code and send it to your company's Web developer to add to your site.

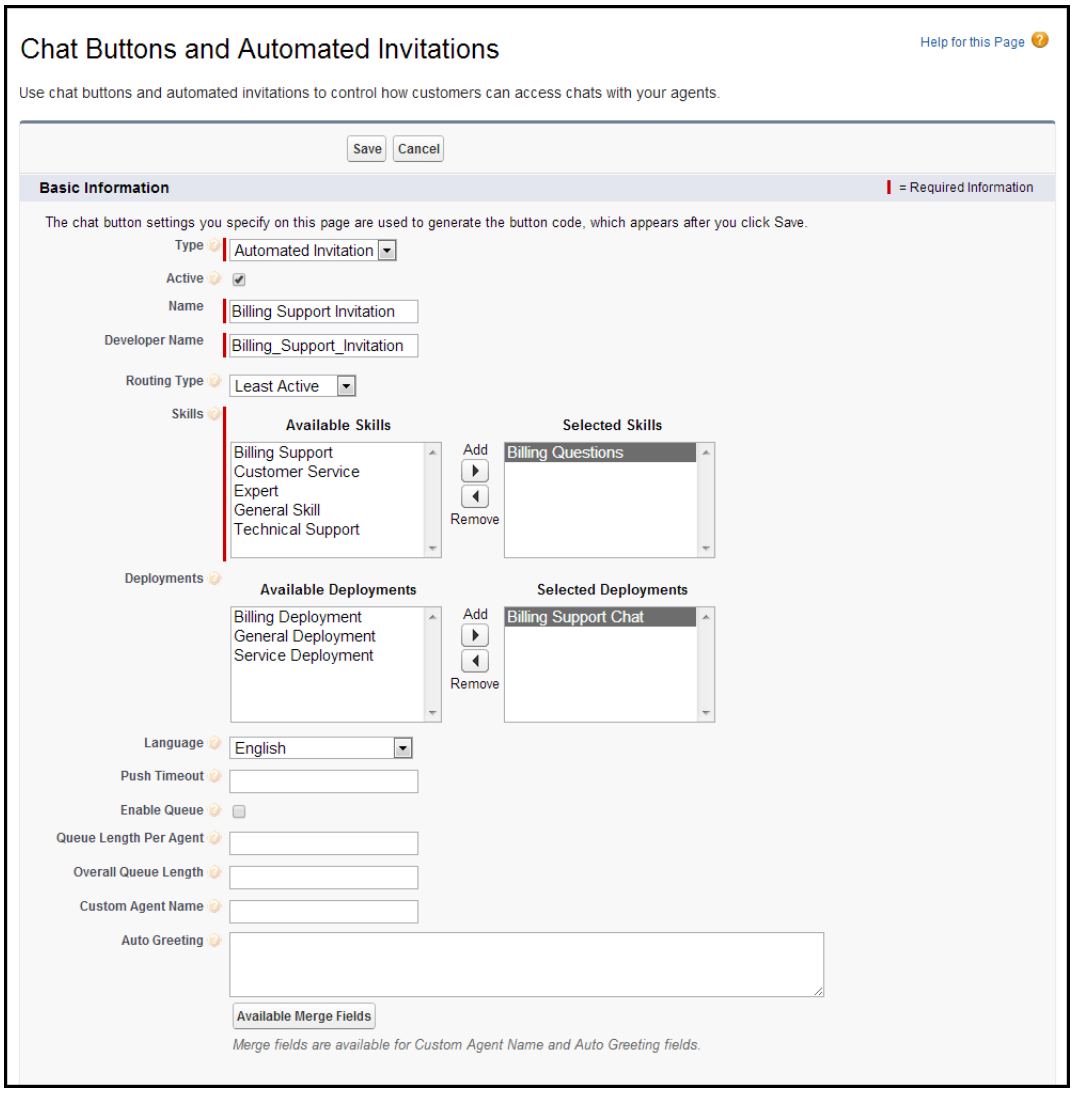
Let's take what we just learned about chat buttons and set up another useful feature for your customers: automated chat invitations.

Step 7: Create Automated Chat Invitations

Having a chat button on your website is great, but automated chat invitations get your customers' attention. An automated chat invitation is an animated popup window that invites a customer to chat with an agent. You can set automated invitations to pop up when certain things happen, such as whether a customer remains on a Web page for more than a few minutes. Invitations are similar to chat buttons, so you can associate them with specific skills to ensure that chats are routed to the right person. Let's set one up.

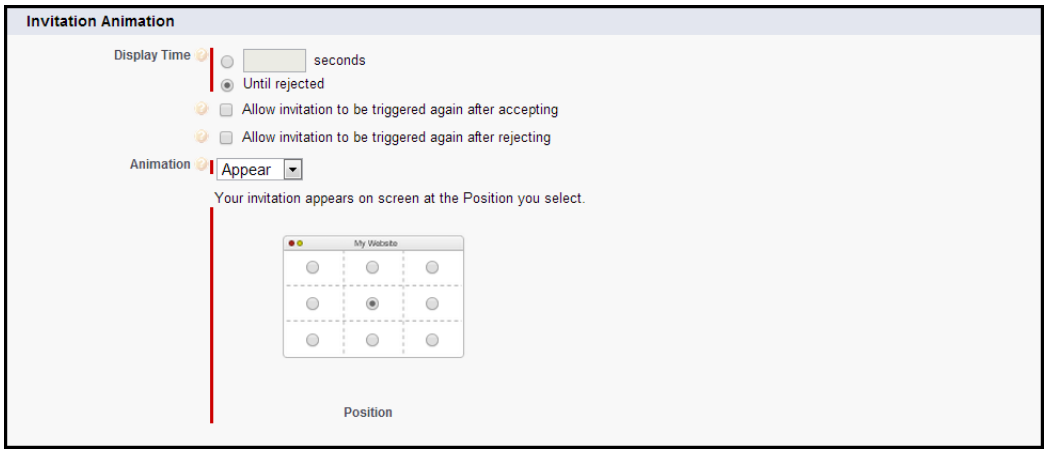
1. From Setup, click **Customize** > **Live Agent** > **Chat Buttons & Invitations**.
2. Click **New**.
3. Select Automated Invitation from the `Type` drop-down list.
4. Select `Active` so your invitation will appear to customers right away.
5. Type a name for the invitation. Let's type `Billing Support Invitation`.
6. Select how chats initiated from this invitation are routed to support agents. Let's choose `Least Active`, which means that incoming chats are routed to the agent with the required skill who has the fewest active chats.
7. Select a skill to associate with the button by moving it from the `Available Skills` list to the `Selected Skills` list. Let's select the `Billing Questions` skill we created earlier.

8. Select a deployment to associate with the invitation. Let's choose the `Billing Support Chat` deployment we created earlier.
9. Select `English` as the language for text in the chat window.



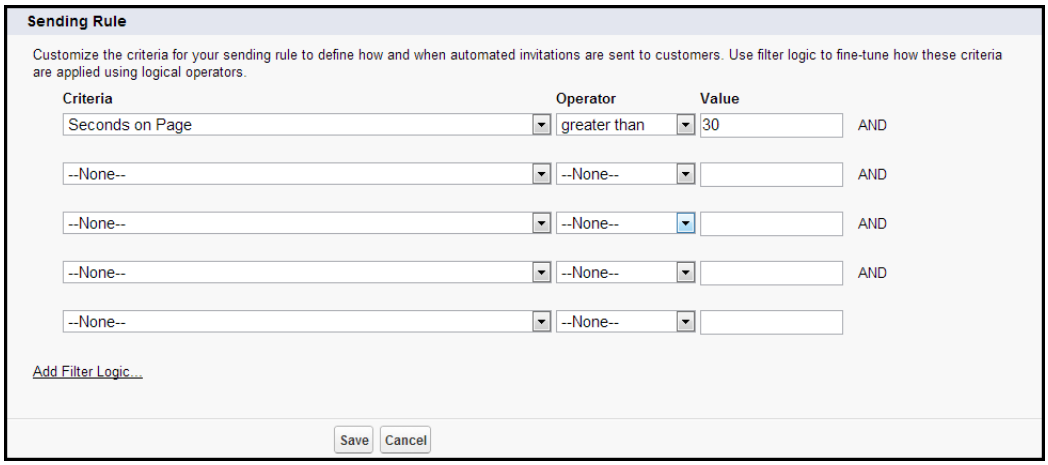
Now, let's skip down to the Invitation Animation section.

10. Select how long you want your invitation to display to customers. In this tutorial, let's select `Until Rejected`. That means the invitation will display to customers until they close it.
11. Choose the animation type. Let's choose Appear.
12. Select the position of the invitation. The invitation will appear on the screen at the position you choose.



You can brand your invitation to control the look and feel of your customers' chat experience, just like chat buttons. But in this tutorial, we'll skip this and jump down to the Sending Rule section.

13. Set up sending rules for your invitation. Sending rules determine when the invitation appears on the screen. For our example, let's define a rule that pops up the invitation if a user has spent more than 30 seconds on the Web page.



14. Click **Save**.

    Salesforce generates the invitation code.

15. Just as you did with your chat button code, copy/paste the generated invitation code and send it to your company's Web developer to put it on your site.

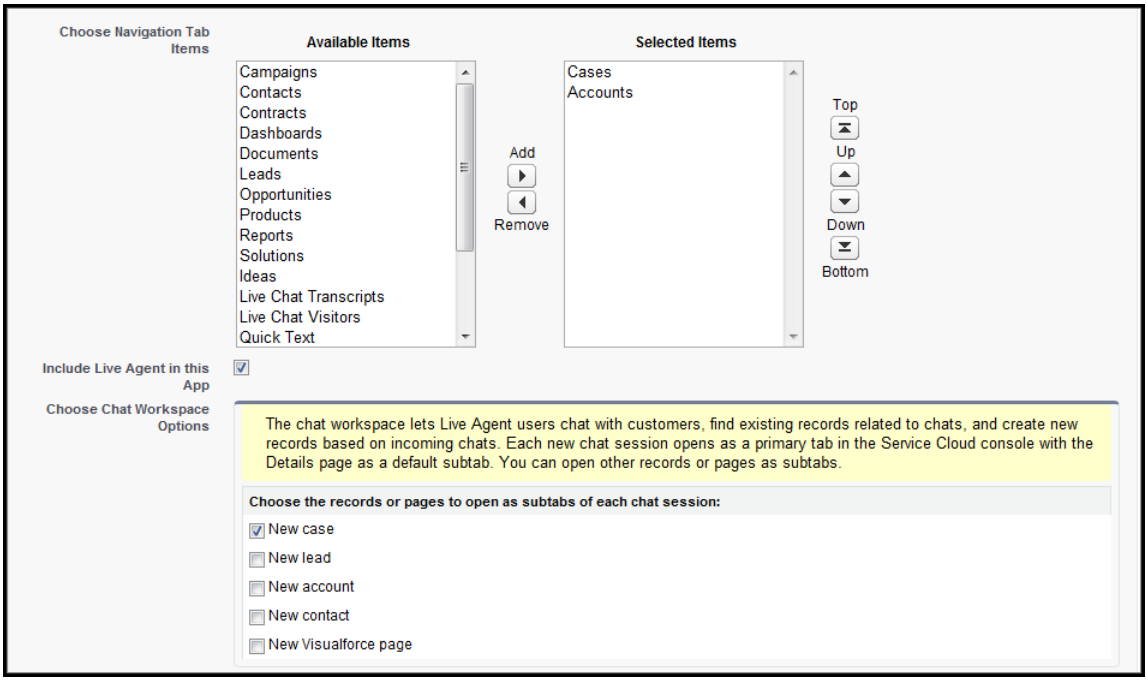
You know what? We've just finished setting up Live Agent. In no time at all, you've added another channel that can deliver service and smiles. Now let's integrate this channel with the console.

Step 8: Add Live Agent to the Help Desk

Let's add Live Agent to the console so that your agents can work with incoming chats from customers. When we're done, your agents will be able to answer questions from emails, the Web, phone calls, and chats—all at the same time. Your agents will be so productive that you're sure to be nominated as Employee of the Month.

1. From Setup, click **Create** > **Apps**.
2. Click **Edit** next to New Console, which is the console we created earlier.
3. Select `Include Live Agent in this App`.

4. Choose the records or pages you want to open as subtabs of chat sessions in the chat workspace. Let's choose `New case` so that your support agents have the option to create a case during each chat session. What a way to increase efficiency!



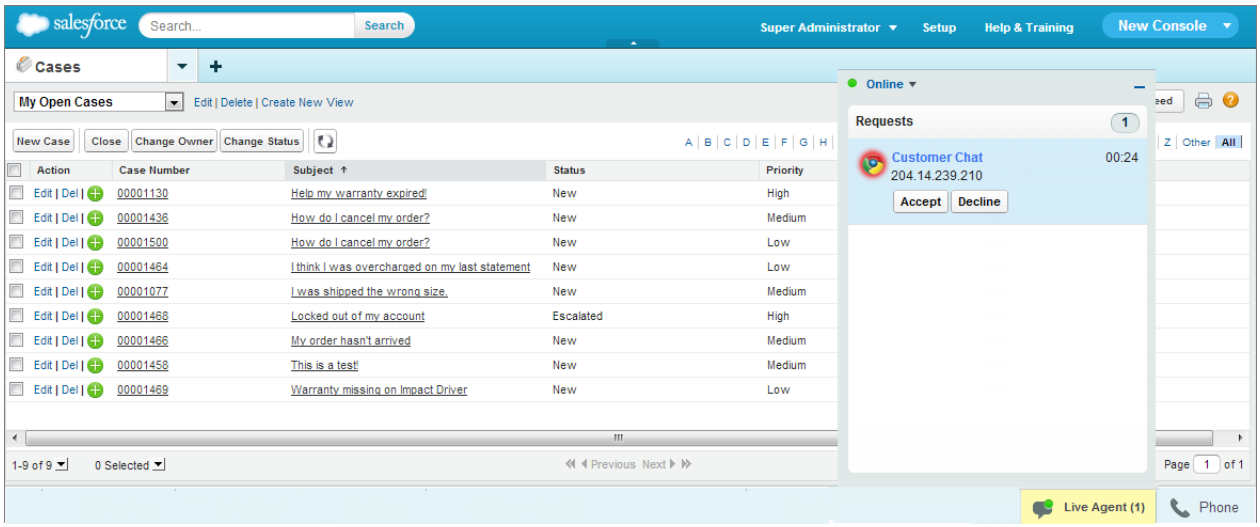
5. Click **Save**.

That's it! We've added Live Agent to the console and your agents are free to chat with customers. Note that if you wanted to standardize chat greetings or other chat messages for your agents, you can create what we call Quick Text. Check out the Salesforce online help for details.

Step 9: View Live Agent in the Help Desk

We've come a long way in no time at all: we've turned on Live Agent, given agents access to it, created a chat deployment for your site, as well as a chat button, so that customers can initiate chats. Now we'll show you how agents can reply to chat requests in the console we created in Tutorial #1: Setting Up a Help Desk.

1. Go to the console we created by clicking New Console from the Force.com app menu.
2. In the chat monitor in the console, choose `Online`. The chat monitor automatically shows a list of incoming requests.



3. Click **Accept** on a chat request.
4. Type your message in the chat text field and click **Send** or press ENTER on your keyboard.
5. Click **End Chat** to end the chat session.

As you can see, it's pretty easy for an agent to chat with customers from the console. Also, agents can transfer chats to other agents, create cases from chats, and turn chats into knowledge base articles. Plus, as an administrator, you can monitor chats to make sure your agents are providing stellar service. Live Agent in the Salesforce console makes life easier for agents, administrators, and your customers!

Summary

Woohoo! You just scored a big win for your company, agents, and your customers. You've added a chat-channel for your customers, and now your agents can use the Salesforce console to respond to chats in real time. Next, we'll show you how to set up a knowledge base so that your service team can easily create, manage, and retrieve content they need while working in the console.

Tell Me More!

We've shown you the basics of how to set up chat with the Salesforce console, but you can take Live Agent to new places by learning about the following features in the Salesforce online help:

- Multiple deployments—Create more than one Live Agent deployment to support multiple websites or multiple products or services.
- Quick text—Customize standard messages that agents can include in their chats.
- Web Communities—Add Live Agent to your Web channel so that your customers have one location for self-service, logging cases, and initiating chats with your service team.
- Brand chat windows—Style your chat windows to promote your brand.
- Customize routing rules—Customize how incoming chat requests are routed to your agents so that customers can get the help they need quickly.
- Live Agent Sessions—Add the Live Agent Sessions tab so that your team can review information about your agents' and customers' chat activities.
- Live Chat Transcripts—Add the Live Chat Transcripts tab so that chat session content is saved and accessible to agents and managers.

- Live Chat Visitors—Add the Live Chat Visitors tab so that your team can review information about visitors that have participated in chat sessions.
- Salesforce Knowledge—Integrate your knowledge base into chat sessions so that agents can add information from knowledge articles directly into chat sessions with customers.

If you're a developer, take a look at the options available through Live Agent's APIs, located in our developer's guides:

- *Live Agent Developer's Guide*
- *Live Agent REST API Developer's Guide*
- *Salesforce Console Integration Toolkit Developer's Guide*

Tutorial #6: Storing and Retrieving Knowledge

Level: Advanced

So far, you've added umpteen new channels to your Salesforce console, but one piece is missing: knowledge. With a shared knowledge base, agents can find and reuse known solutions and access the most up-to-date information about your products or services. In this tutorial, we'll show you a quick way to set up Salesforce Knowledge—you can add all kinds of bells and whistles later—and we'll show you how to integrate it with your console. You can also make Salesforce Knowledge available to your customers so that they can answer their own questions, and at the end of this exercise we'll point you to some resources that will help you do just that.

Suppose your company wants a centralized, reliable source of product or service information. Let's also suppose your management wants to reduce the time it takes to find information by categorizing it. In this tutorial, we'll show you how you can do both. We'll set up Salesforce Knowledge with basic features so that your agents can quickly find information that they can use to help customers—and they'll find this information while they're working on cases with the console.

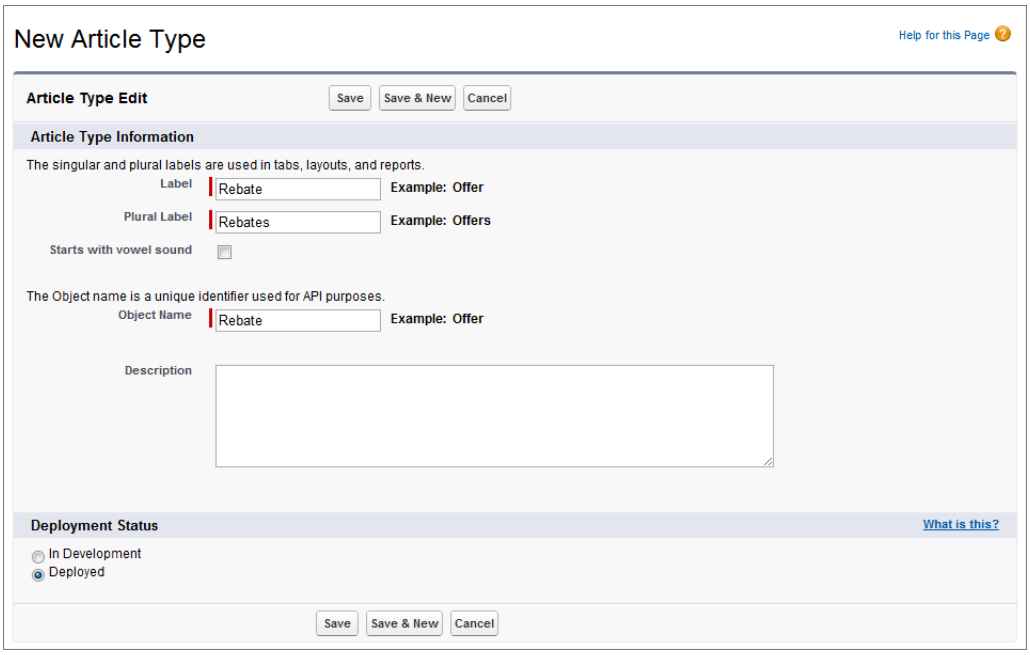
Step 1: Create Article Types

In this highly customizable universe, you can create a knowledge base that lets some agents write drafts of articles while others rewrite, translate, and approve articles. You can even fine tune your knowledge base so that it automatically routes articles to the right people for review. But in this exercise, we're going to stick to the basics and just get you started with Salesforce Knowledge. The first step to setting up Salesforce Knowledge is to create article types. Article types, such as FAQ's or tutorials, provide the format and structure that control how a knowledge base article displays to an audience.

1. From Setup, click **Customize** > **Knowledge** > **Article Types**.
2. Click **New Article Type**.
3. Type a `Label` and `Plural Label` for the article type. Let's label your article type `Rebate`.

   Notice that `Object Name` is automatically updated with the `Label`. If we were developers, we could use the `Object Name` for knowledge integrations to Salesforce.

4. Click `Deployed` so that users can choose your article type when creating knowledge articles.



5. Click **Save**.

We've just created an article type. You can create new article types at any time to extend your knowledge base and present new types of information to different people. Now, let's turn on Salesforce Knowledge and choose some settings.

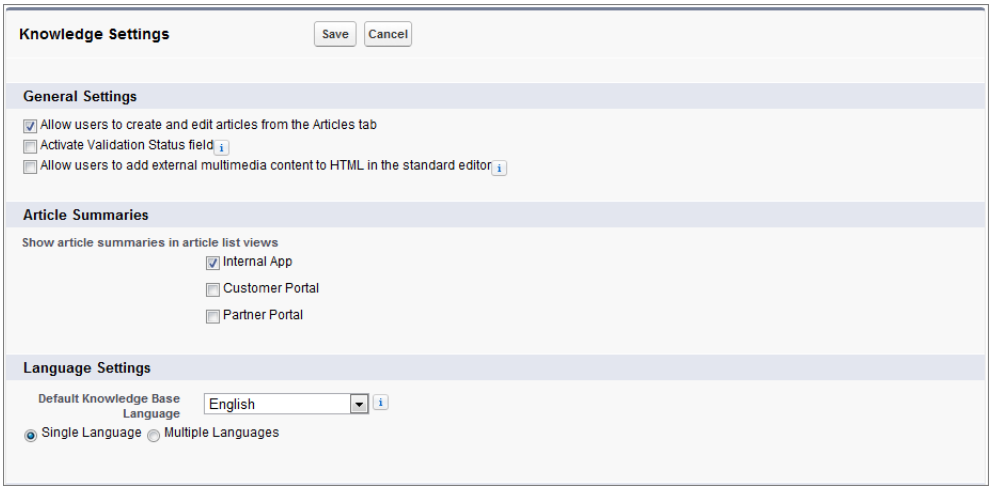
Step 2: Enable Salesforce Knowledge and Select Settings

In this step, we'll turn on Salesforce Knowledge and choose some settings. This will get you and your team closer to organizing all of your company's information so that it can be used to help your customers—and everyone else at your company.

1. From Setup, click **Customize** > **Knowledge** > **Settings**.

   If a message notifies you that enabling Salesforce Knowledge is irreversible and displays the Articles tab to all standard profiles, you can choose to continue with the implementation by checking `Yes` and clicking **Enable Salesforce Knowledge**.

2. Click **Edit**.
3. In General Settings, click `Allow users to create and edit articles from the Articles tab`.
4. In Article Summaries, click `Internal App`. This just means that we're going to display knowledge summaries to your agents in Salesforce. Notice that you can display knowledge summaries in portals too.
5. In Language Settings, we'll select English as our `Default Knowledge Base Language`, and `Single Language`. Keep in mind that you can't remove a language once it's added.



6. Click **Save**.

We've just turned on Salesforce Knowledge, but your knowledge base is pretty empty. Next, we'll start to fill up your knowledge base with data category groups so that agents creating or managing articles can categorize them to help others quickly find info.

Step 3: Create Data Category Groups

Let's create category groups—containers for the individual data categories used to organize your knowledge articles. For example, a Contracts category group might contain Fixed Price, Cost Reimbursement, and Indefinite Delivery categories. Let's add some organization to your company's information so that your agents can find what they need to make your customers happy.

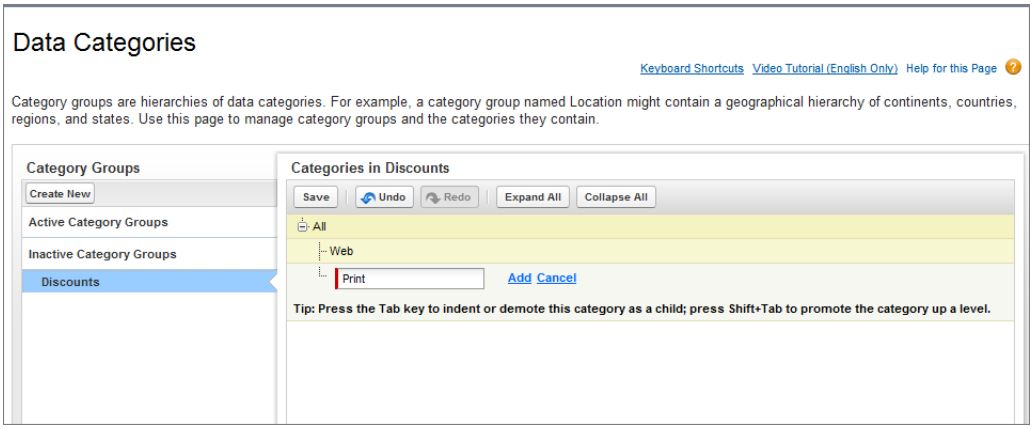
1. From Setup, click **Customize** > **Data Categories** > **Data Category Setup**.
2. Type a `Group Name`. Let's type `Discounts`.

   Notice that `Group Unique Name` is automatically updated with the `Group Name`. If we were developers, we could use the `Group Unique Name` for knowledge integrations to Salesforce.

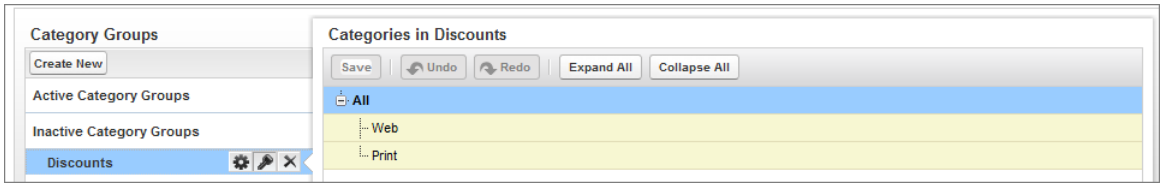
3. Click **Save**.

   Notice that a top-level category in the group named All is automatically created for you.

4. Type a category for our group. Let's type `Web`.
5. Click **Add**.
6. Type another category for our group, such as `Print`, and click **Add**.



7. Click **Save**.
8. Hover your mouse pointer over the category group name and click to activate it. This makes the group available to your service team.



Congratulations, you've taken another step towards organizing your company's product and service information! Your agents are about to find everything they need to help customers in seconds. Now that we've created and activated your data category group, we're ready to choose which agents see it.

Step 4: Set Data Category Group Visibility

In this step, we'll choose which agents can see the data category groups we created. This helps determines the articles agents can access so that they don't waste time seeing articles that have nothing to do with solving your customers' issues. As an administrator, you have the flexibility to base data category visibility by roles, profiles, or permission sets. To keep things easy, we're going to base data category visibility by profiles.

1. From Setup, click **Manage Users** > **Profiles**.
2. Click on a profile name. Let's choose your profile.
3. In Category Group Visibility Settings, click **Edit** next to the data category group we created earlier.

4. Select a visibility setting. Let's select `All Categories` so that agents can see all categories in the group we created.

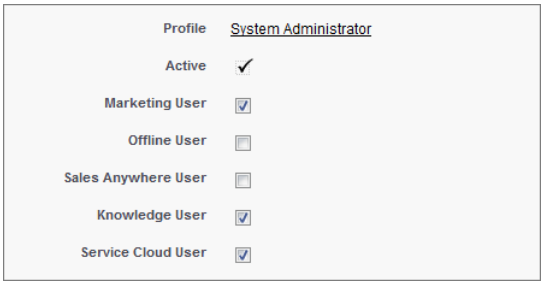
5. Click `Save`.

We've just streamlined your company's information for your agents. By choosing which data category groups your agents can see, we've narrowed down volumes of information to just relevant info that will help your service team.

Step 5: Grant Users Access to Salesforce Knowledge

Take a step back and look at all that you've accomplished so far in this exercise. You've created an article type for a knowledge base, turned on and configured a knowledge base, organized your knowledge base by categories, and managed information overload by determining who should see what in your knowledge base. Now that you're done setting up Salesforce Knowledge, the next step is to assign agents the Salesforce Knowledge feature license so that they can access data categories and articles that are visible to them.

1. From Setup, click **Manage Users** > **Users**.
2. Click **Edit** next to a user's name. Let's do this next to your name.
3. Select the `Knowledge User` checkbox.



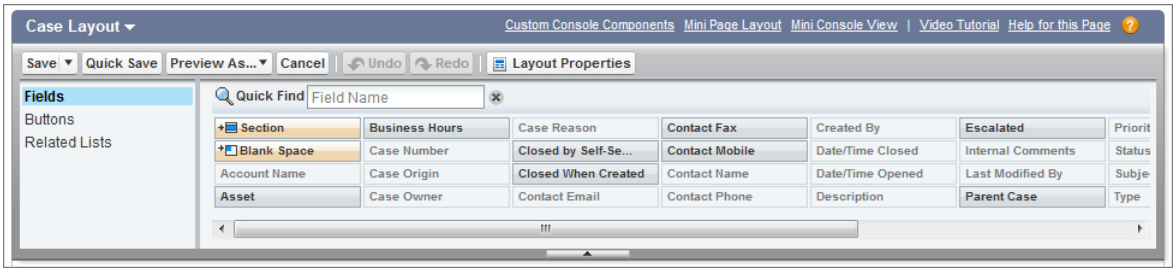
4. Click **Save**.

We've just given your agents access to Salesforce Knowledge. Was that easy or what?

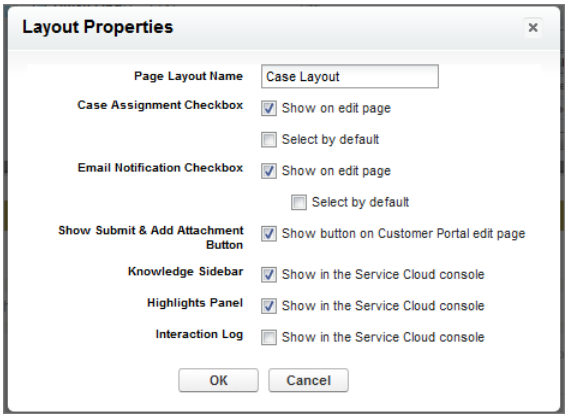
Step 6: Turn on the Knowledge Sidebar in the Help Desk

Since we've set up Salesforce Knowledge, we're ready to turn on the Knowledge sidebar in your console. This will only take a few minutes. The sidebar automatically displays articles that might solve any cases agents work on in the console—it magically passes along any useful information to help your agents close cases. The magic is that articles appear based on any matching words typed in a case's `Subject`. Your agents don't have to spend time searching your knowledge base because it's searched for them while they type.

1. From Setup, click **Customize** > **Cases** > **Page Layouts**.
2. Click **Edit** next to a page layout. We recommend that you don't edit the Close Case Layout because a case is solved by the time an agent views that page.
3. Click **Layout Properties**.



4. Check `Knowledge Sidebar`, and click **OK**.



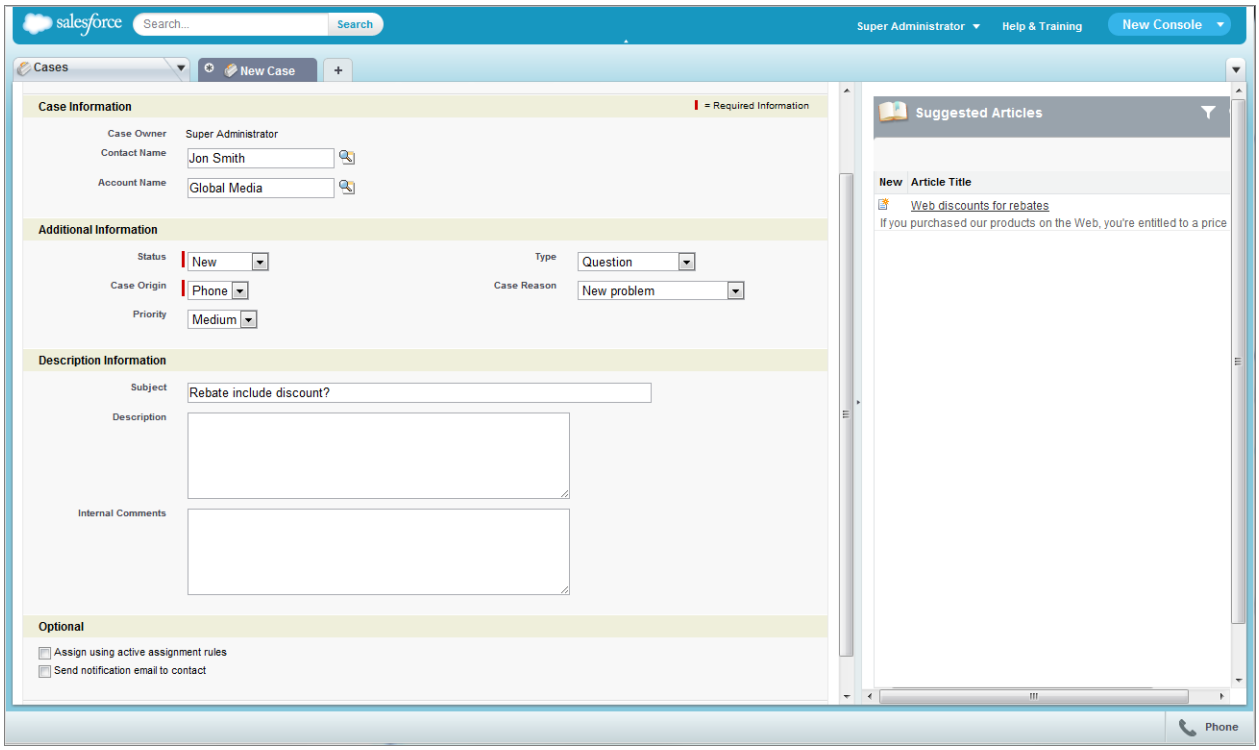
5. Click **Save**.

We've integrated Salesforce Knowledge with your Salesforce console. Next, we'll see how the integration looks to agents.

Step 7: View the Knowledge Sidebar in the Help Desk

We're almost done! In the previous steps, we set up Salesforce Knowledge and configured it to automatically suggest articles to agents using the console. Now, we'll show you this time-saving feature in action. As agents, we'll work on a case with the console we created in Tutorial #1: Setting Up a Help Desk, and we'll see how articles magically appear to help us solve the case. Note that this step assumes that you've created an article with the article type and data categories we configured earlier, and that you published the article to your knowledge base. To learn how to publish articles, check out "Publishing Articles" in the Salesforce online help.

1. Go to the console we created by clicking New Console from the Force.com app menu.
2. Click Cases from the navigation tab.
3. Click **New Case**.
4. Type case details into the `Subject`. Let's type `Rebate include discount?`
5. Notice that the Knowledge sidebar automatically displays an article that includes the words we entered in the `Subject`.



You've just witnessed incredible productivity. Now whenever an agent selects a case in your Salesforce console, your knowledge base will immediately display potential answers. Congratulations, your service team now has a tool to help them close cases faster!

Summary

You've just created a basic knowledge base for your company, and now your agents can automatically see relevant articles to help them solve any customer case, regardless of its channel. Because reliable information is the backbone of service, we showed you how to set up Salesforce Knowledge so that your team can find, reuse, and organize info that will help your customers. With just a few clicks, we integrated your company's knowledge base with your console so that your agents can close cases faster. To finish the setup, or to create a more powerful knowledge base that routes articles to qualified reviewers for approval, check out "Granting Permissions for Salesforce Knowledge Users" and "Salesforce Knowledge Workflow Rules and Approval Processes" in the Salesforce online help.

Tell Me More!

We've shown you the basics of how to set up a knowledge base that works with the Salesforce console, but you can take Salesforce Knowledge to the next level by learning about the following features in the Salesforce online help:

- Multiple portals—add Salesforce Knowledge to your Customer Portals or Partner Portals so that your customers and business partners can find the info they need.
- Workflow and approval processes—create processes to help your support team manage article creation and publication.
- Multiple languages—translate and publish articles in more than one language.
- Import articles—import articles from existing knowledge bases into Salesforce Knowledge.
- Web communities—make articles visible to customers using your Web channel.
- Synonyms—create a synonyms group to let knowledge users search for articles using synonyms or keywords.

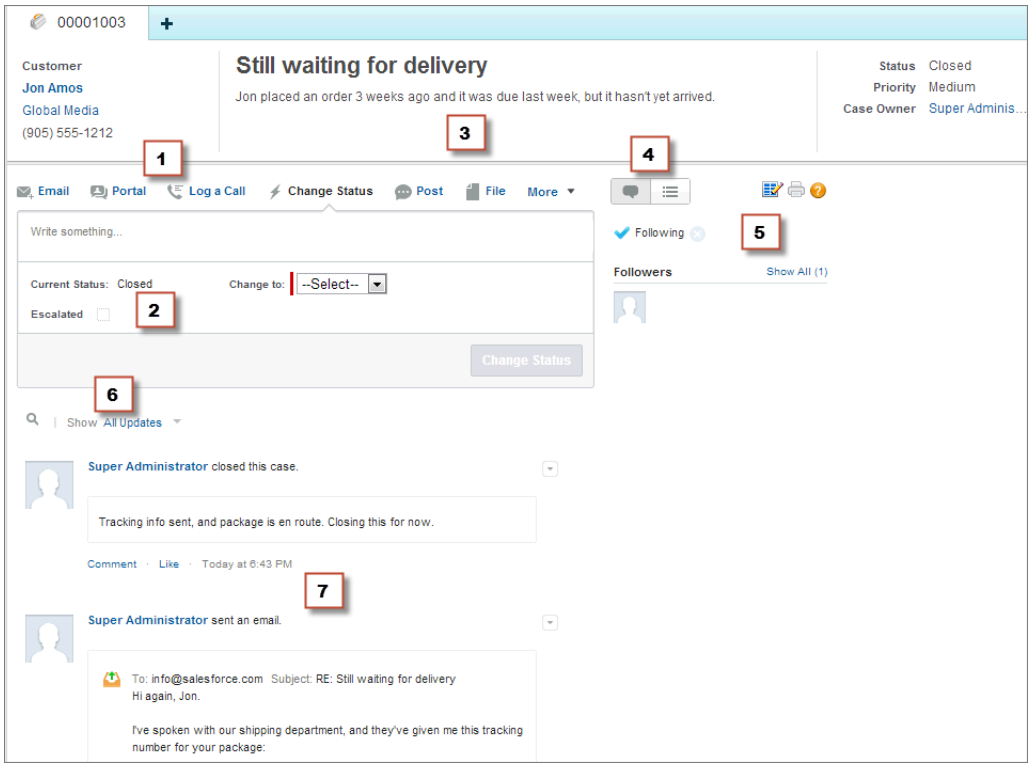
Tutorial #7: Streamlining the Look and Feel of Cases

Level: Beginner

Last but not least, why not give your agents a simpler, more interactive way of managing and viewing cases? After all, they're probably working with dozens—if not hundreds—of cases each day, and they could use a break. Case Feed makes working with cases easier by offering two separate views:

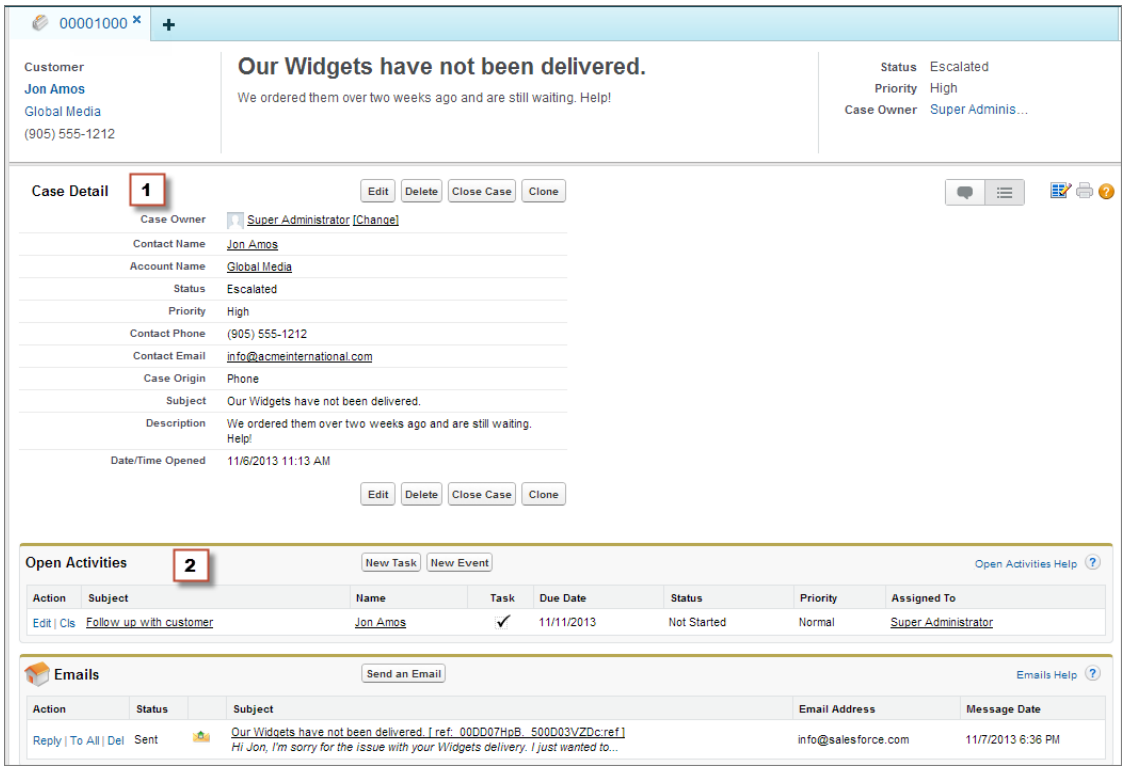
1. Feed view—agents can quickly see a case's history, follow the case to get updates, and use actions in the publisher to do things like email customers, type notes about a phone call, or change the case's status.
2. Detail view—agents can see complete case details and work with related lists.

Here's what the feed view includes.



1. The publisher, which includes actions for the tasks agents do most often when working with cases
2. Customizable fields on each action so you can include the info that's most relevant to your business
3. The highlights panel, with the most important case details
4. A button for switching between the feed view and the detail view
5. Tools and components such as a **Follow** button and followers list
6. Feed filters, which let agents find specific info in the feed quickly and easily
7. The feed itself, with a chronological list of activity on the case

And here's a peek at the detail view.



1. Comprehensive details about the case
2. Related lists on the case

The benefit of Case Feed is that it's flexible enough to support agents however they want to work. For example, one agent might want to use the Email action in the publisher to quickly send a message to a customer. Another agent handling a more complex case that has required several phone calls and emails might browse the case history in the feed to see what happened, and then use the Emails related list on the detail page to see all of the messages sent to the customer by other agents.

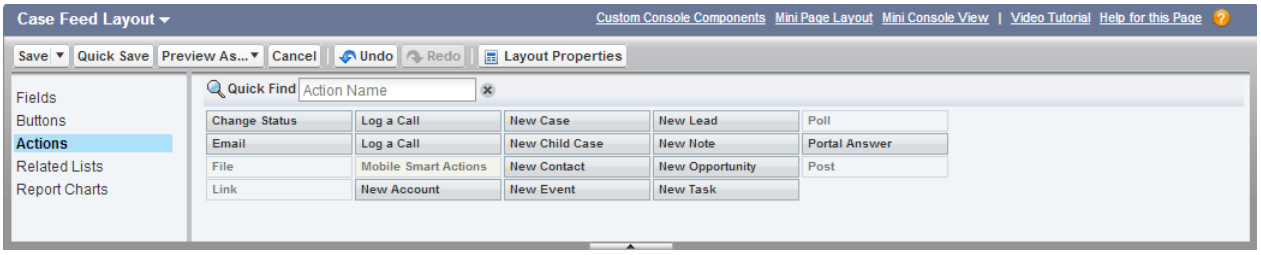
In this tutorial, we'll set up Case Feed in just a few steps. If you purchased Salesforce *before* Winter '14, this tutorial isn't for you, because setting up Case Feed requires a little more effort for older organizations. Check out "Setting Up Case Feed" in the Salesforce help for the nuts and bolts.

Step 1: Customize What Agents See in the Publisher

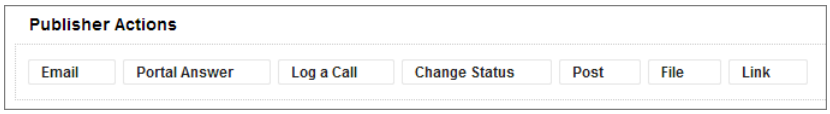
Ready for the easiest thing you'll do all day? Let's choose what your agents will see in the Case Feed publisher, feed view, and detail view. We'll start by tackling the publisher—that is, the tool at the top of the feed that includes the actions agents use to work with a case, such as Email and Log a Call actions.

1. From Setup, click **Customize** > **Cases** > **Page Layouts**.
2. In Page Layouts for Case Feed Users, click next to the Case Feed Layout and select **Edit detail view**.
3. The page layout editor appears. Let's start by stocking the publisher with actions that will be useful to support agents. In the Publisher Actions section, click **override the global publisher layout**.

4. In the palette at the top of the page, click **Actions**. A list of available actions appears. The actions that are already included in the publisher are greyed out.



5. Drag Mobile Smart Actions and Poll from the Publisher Actions section and drop them on the palette. This removes actions that support agents probably won't need on cases.
6. Drag these agent-friendly actions from the palette and drop them onto the publisher:
   - Email—lets agents write and send messages to customers.
   - Portal Answer—lets agents post replies to your Web community.
   - Log a Call—lets agents jot down notes about calls and other customer interactions. Pick the first Log a call action (yes, there are two) because it's designed specifically for Case Feed.
   - Change Status—lets agents close, reopen, or escalate cases.
7. Drag the actions in the publisher layout into this order since agents will use them most often:



8. Click **Save**, and presto!

We've finished getting the publisher spiffed up. Now let's work on the detail view.

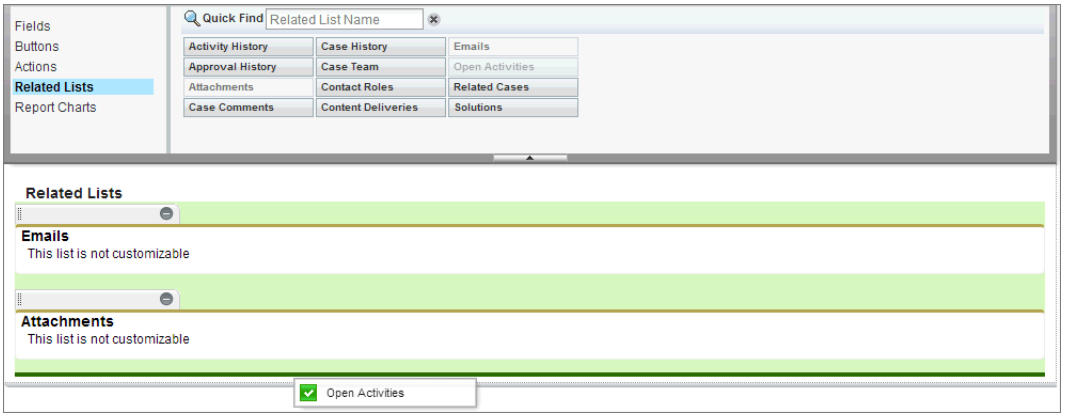
Step 2: Customize the Case Detail View

The detail view in Case Feed shows agents a case's details and related lists. You can add buttons and links, but let's keep things simple and just add one field and a few related lists for your agents.

1. From Setup, click **Customize** > **Cases** > **Page Layouts**.
2. In Page Layouts for Case Feed Users, click next to the Case Feed Layout and select **Edit detail view**.
3. Drag `Date/Time Opened` from the Fields section to the Case Details section so that your agents can quickly see how long any case has been around.

4. Click **Related Lists** in the palette, and drag the following to the Related Lists section:
   - Attachments

- Emails
- Open Activities



We're adding only a handful of lists to keep the layout uncluttered. One of the benefits of Case Feed is that a lot of what agents see in the feed, such as an activity on a case, replaces many of the standard related lists, which can be clunky to work with.

5. Click **Save**.
6. Click **Yes** in the dialog box that asks if you'd like to overwrite users' related list customizations. This makes sure that all of your agents see the same related lists on cases.

Then give yourself a pat on the back. You only have one more layout to update!

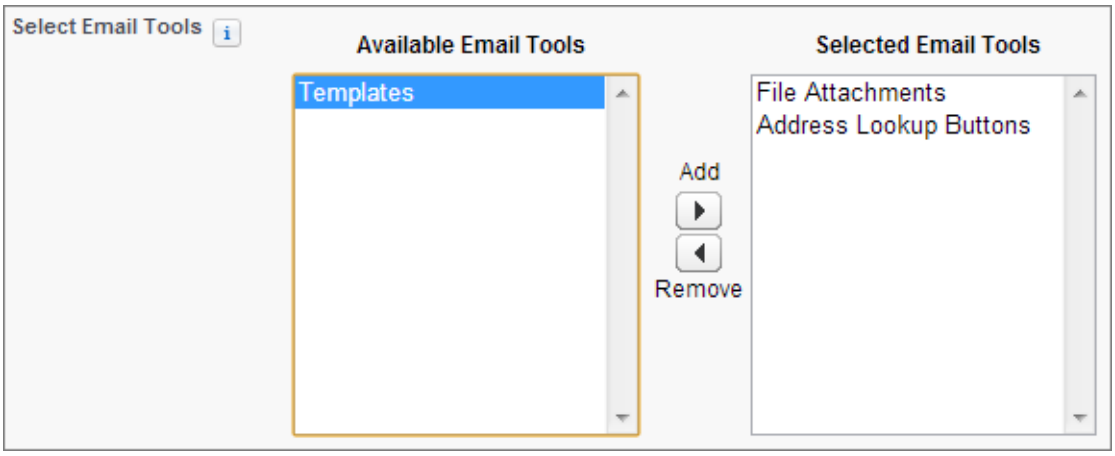
Step 3: Customize the Feed View

In the previous steps, you chose what your support agents will see in each case's feed and detail views. Now, we'll make a few small tweaks to the feed view so that your agents can close cases faster. Are you ready?

1. From Setup, click **Customize** > **Cases** > **Page Layouts**.
2. In Page Layouts for Case Feed Users, click next to the Case Feed Layout and select **Edit feed view**.

   Now we have the option to add more fields to the Log a Call, Change Status, and Email actions. Because we don't want to slow down your agents when they're writing call notes, updating a case status, or sending email, let's leave the fields on these actions alone. We're also going to leave the pre-set email header fields intact—no need to remove standard email fields.

3. From the Email action, remove **Templates** from the Selected Email Tools list. We're hiding the templates tool since we haven't created any email templates. Remember, the email action has fields, header fields, and tools; we're changing only the tools.



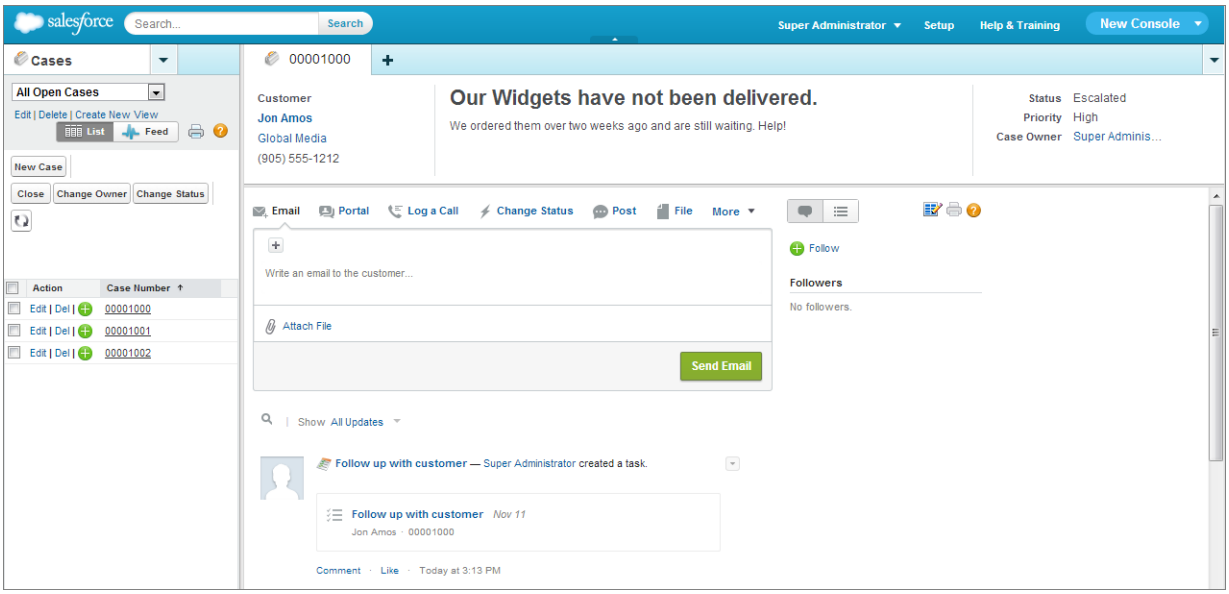
4. In the Case Feed tools section, remove **Article** from the Selected list We don't need the Article tool because we turned on the Knowledge sidebar for the console in Tutorial #6: Storing and Retrieving Knowledge.
5. Next to Filters Appear, select `As a drop-down list in the center column`. This will make feed filters appear directly above the feed in a neatly collapsed list.
6. Click **Save**.

Guess what? We're done! Let's check out our handiwork.

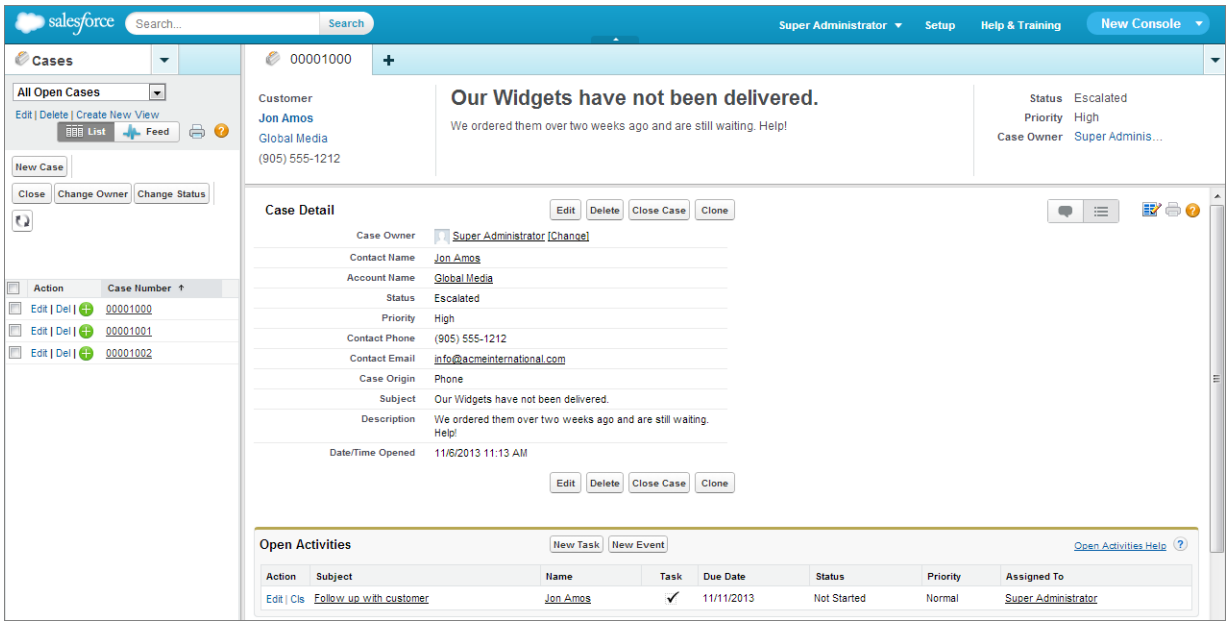
Step 4: View a Case in the Help Desk

Now let's take a look at Case Feed in the console to see how it will make your agents' lives easier.

1. Go to the console we created by clicking New Console from the Force.com app menu.
2. Click Cases from the navigation tab.
3. Click a case to view it.
4. Notice that the feed view for the case appears.



5. Click to switch to the detail view.



Nicely done! Your support agents should treat you to dinner for streamlining how they work with customer cases.

Summary

You made it! We customized the Case Feed publisher with actions that'll make your agents' jobs easier. You also tweaked the case detail view so it's easy to quickly find info on any case, and changed the feed view so it includes only the options and tools you need. In a few simple steps, we created a look-and-feel for cases that's clean and comprehensive at the same time.

Tell Me More!

You've just set up Case Feed to make your agents happier and more productive, but you can do a lot more with this simple tool. Check out the Salesforce online help for info on these features:

- Quick text—create pre-written messages that agents can email to customers.
- Email templates—create text, HTML, or Visualforce email templates to help agents save time and increase consistency.
- Email drafts—let agents save drafts of messages before they send them to customers.
- Approval actions for email—create approval actions and processes for peace of mind or to train new agents.
- Portal email notifications—send customers an alert email when agents respond to their posts on your Web communities.
- Publisher actions—add more actions to the publisher to let agents do more in the feed.
- Custom buttons and links—include custom functionality in the feed or detail view.
- Custom Case Feed components—create Visualforce pages to add customized tools to your feed view layouts.

Comfortable with Visualforce? Take a look at the *Customizing Case Feed with Visualforce* for even more ways of making Case Feed your own.

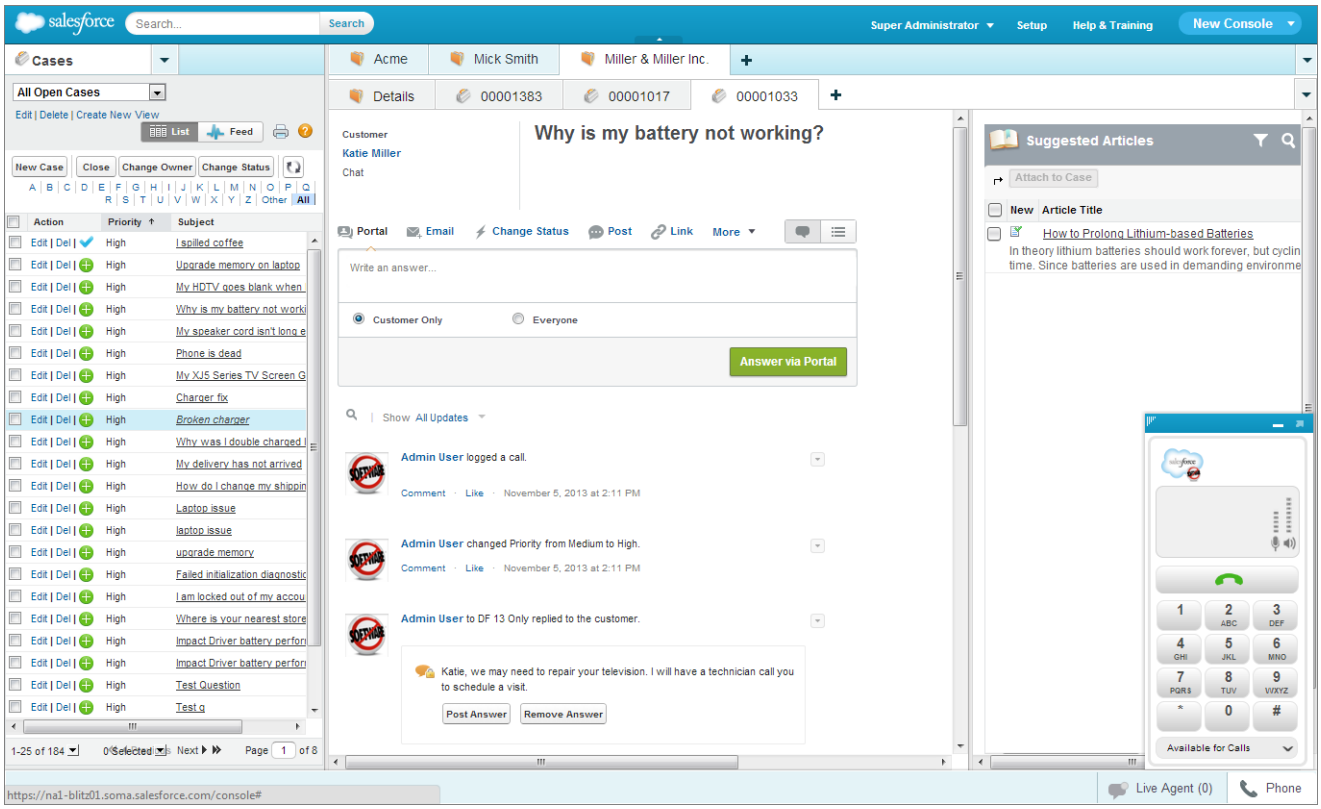
Final Summary

You should be smiling from ear-to-ear because you've just finished setting up a help desk! You might not have thought it was possible for one person to set up a call center in a few hours, but you just did. Plus, you've opened new channels to your customers while increasing your agents' productivity—Win-win.

We've done a lot since the beginning of this book. We've covered how to set up and customize a help desk for your agents. We've also covered how to integrate your help desk with Web, email, chat, and phone channels. Towards the end, we created a knowledge base so that your agents can access a reliable source of information to help your customers, and we streamlined how cases look to your agents so that they can interact with them more efficiently. Ultimately, we set up a complete, scalable call center that can support a team of agents anywhere in the world!

The call center we created is just the beginning. Through the *Tell Me More!* sections in this book, we've shown you additional, powerful features. Imagine what you can do if you really spend some time exploring the possibilities. We encourage you to learn more about these features because, with one single click, they might just solve some of your complex business problems, help your customers, or lower costs.

The tutorials we presented build on one another. If you followed them in order, your help desk will look like the one below. But now that you know how to build and customize a call center, what will your help desk look like?



http://developer.force.com/workbooks