Embedded Chat

Salesforce, Winter '27

Last updated: July 22, 2026

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

CONTENTS

Embedded Chat 1
Set Up Embedded Chat 3
Set Up Appointment Management 27
Embedded Flows 29
Embedded Static Resources 31
Test Your Embedded Chat Window 34
Limitations 35
Embedded Service Cookies 37

EMBEDDED CHAT

Add an embedded chat window to your company website for customers to get quick answers to their questions while browsing your pages. Include a pre-chat form to collect key information and customize the experience to match your branding.

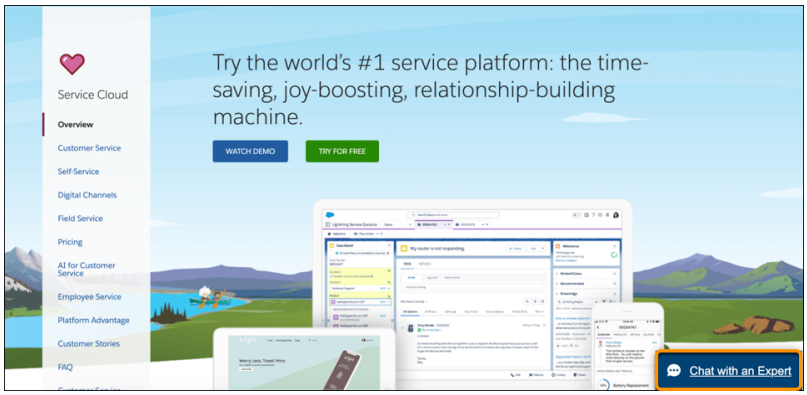
Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

The chat window sits unobtrusively on your web page. When customers want to chat, they click the button to launch.

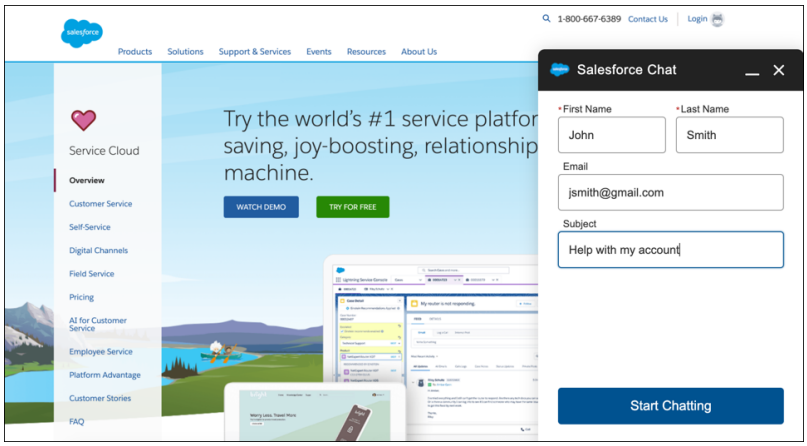
EDITIONS

Embedded Service setup node is available in: Lightning Experience

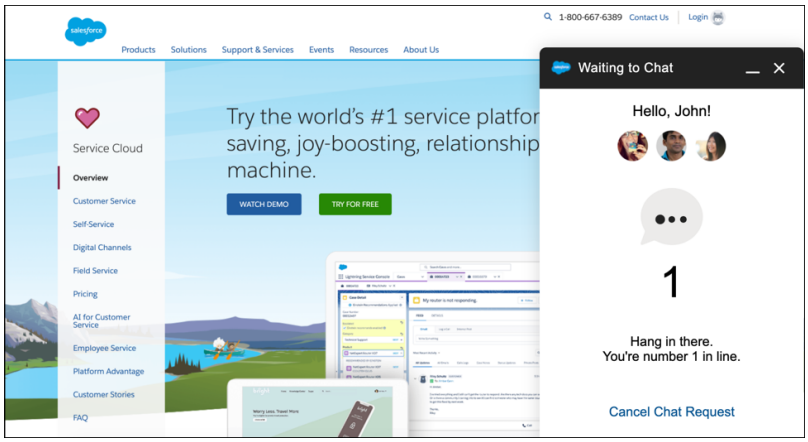
Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions with Service Cloud or Sales Cloud



Add a pre-chat form for customers to complete, which helps agents gather basic information like their contact details, before the conversation.



Customers can start chatting while viewing your web page. They're also able to minimize the chat window so it's not in the way. The chat persists across your web pages and their subdomains, so they can continue browsing while waiting or chatting with an agent.



Embedded chat windows use a Chat deployment that you can quickly configure. Then, simply add the code snippet to the web pages where you want the chat window to be available. Setting up Embedded Chat doesn't change how your agents chat with customers in their console.

Set Up Your Embedded Chat Window

Ensure that your org meets the following prerequisites before setting up Embedded Chat. Follow these steps to configure the embedded chat window and add to your web pages.

Set Up Appointment Management

Appointment Management gives your users an easy way to schedule, modify, and cancel appointments. The experience is powered by Flow Builder. You craft the perfect interaction and decide when and how to create related records like work orders.

Add an Embedded Flow to Automate a Task

Use an embedded screen flow to help customers or users complete simple, automated tasks. An embedded flow is also guest-user friendly because users don't need to log in to access the flow.

Create Embedded Static Resources

Create and upload JavaScript files to Salesforce as static resources and then apply them to your website or Experience Builder site. With static resources, you can change one or many pages with a few clicks in code settings on existing deployments.

Test Your Embedded Chat Window

After adding Embedded Chat to your web pages or Experience Builder site, check the customer's experience.

Embedded Chat Limitations

Before setting up Embedded Chat, consider general guidelines, limitations, and cookies and customer browser restrictions.

Embedded Chat Cookies

Embedded Chat uses cookies to improve functionality and accelerate processing times. By saving a user's settings, cookies can enhance the user's experience and the performance of your Embedded Service.

Set Up Your Embedded Chat Window

Ensure that your org meets the following prerequisites before setting up Embedded Chat. Follow these steps to configure the embedded chat window and add to your web pages.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

To set up Embedded Chat, your org must meet these prerequisites:

- Lightning Experience must be enabled to set up Embedded Service
- Service Cloud License if you have Performance or Unlimited editions
- Digital Engagement License if you have Enterprise Edition. If you're on Performance, Unlimited, or Developer editions and don't have a Service Cloud License, you also need the Digital Engagement License.
- Chat must be enabled in your org
- A Chat button and a Chat deployment must be set up and available in your org
- An Experience Builder site (preferable) or a Salesforce site must be set up on your org and available for guest user access
- Ensure your browsers are supported for Embedded Chat. We support the same browsers as Experience Builder sites. See Supported Browsers for Experience Cloud Sites for more information.

Note: Administrators must use Lightning Experience to access the Embedded Service setup pages. However, other users in the org aren't required to have access to or use Lightning Experience with Embedded Chat.

To set up Embedded Chat, follow these steps:

1. Create an Embedded Service Deployment
   Create an Embedded Service deployment for each embedded chat window that you're using.
2. Specify Your Chat Window's Settings
   Select your chat deployment and chat button. Customize the pre-chat form and offline support form. Select custom Lightning components. Set chat images, dimensions, and labels. Learn about available Chat features in Embedded Chat.
3. Customize the General Branding for Your Chat Window
   Select the colors and font used in your chat window to reflect your company's brand identity.
4. Add Your Embedded Chat to a Website
   Salesforce generates a unique Embedded Service code snippet from the information you provide during setup. Copy and paste the code snippet to include chat on your web pages. Add the optional meta tag code snippet to improve performance for both mobile and desktop.

EDITIONS

Embedded Service setup node is available in: Lightning Experience

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

USER PERMISSIONS

To set up Embedded Chat:
- Customize Application AND Modify Metadata

To create, update, or delete deployments:
- Customize Application AND Modify Metadata

5. Add Your Chat Window to an Experience Site
   Use the Embedded Service component to add your chat window to an Experience Builder site. Create a snippet settings file to use snippet-only settings with your site's chat window.

SEE ALSO:

Verify that Lightning Experience Is Turned On

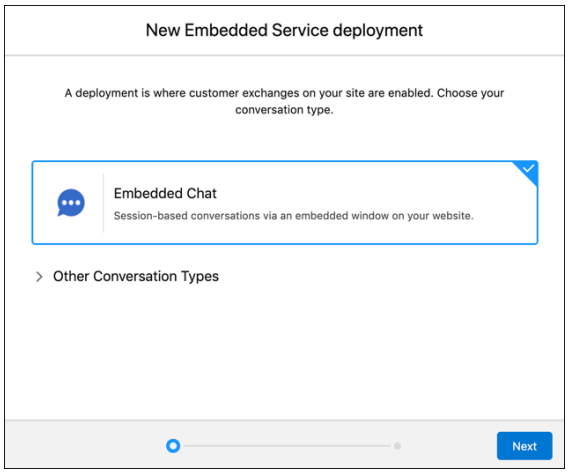
Create an Embedded Service Deployment

Create an Embedded Service deployment for each embedded chat window that you're using.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

The Embedded Chat setup uses an Experience Builder site or Salesforce site, so that you can associate users with a guest user profile. Check if you already have any sites that you can use. If a site is already available, you can use that as your endpoint. Otherwise, create a site for this purpose.

1. From Setup, enter *Embedded Service* in the Quick Find box, then select **Embedded Service**.
2. In the Embedded Service configuration page, click **New Deployment**.
3. Choose a conversation type for your deployment by selecting **Embedded Chat** and click **Next**.



4. In the `Embedded Service Deployment Name` field, enter a name for your chat deployment.

   Tip: The deployment is shown in a list on the Embedded Service page. If you have several deployments, use a descriptive name so it's easy to distinguish each one.

5. The `API Name` field is automatically populated.
6. In the `Site Endpoint` menu, select an Experience Builder site or Salesforce site from the drop-down list.

EDITIONS

Embedded Service setup node is available in: Lightning Experience

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

USER PERMISSIONS

To set up Embedded Chat:
- Customize Application AND Modify Metadata

If nothing appears in the menu, it's because there aren't any sites configured in your org.

7. Click **Save**.

Note: For existing deployments that are not active, a conversation type can be required to enable it on your site (see step 3). After you select your conversation type and save, the choice is permanent for that deployment.

SEE ALSO:

Specify the Chat Settings for Your Embedded Chat Deployment

Add an Embedded Flow to Automate a Task

Set Up Appointment Management

Specify Your Chat Window's Settings

Select your chat deployment and chat button. Customize the pre-chat form and offline support form. Select custom Lightning components. Set chat images, dimensions, and labels. Learn about available Chat features in Embedded Chat.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

EDITIONS

Embedded Service setup node is available in: Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions with Service Cloud or Sales Cloud

Specify the Chat Settings for Your Embedded Chat Deployment

Embedded Chat uses a Chat deployment ID and button. The Chat configuration details that you provide are used to generate the chat code snippet that you add to your web pages.

Customize the Embedded Pre-Chat Form

Gather contact information and learn about customer needs before chatting. Create a custom pre-chat form and associate Salesforce records like leads, cases, and contacts. You can also customize the field labels used on the form.

Customize the Embedded Chat Offline Form

Allow customers to submit a case form with Embedded Chat when your support agents are offline. Customize the form depending on your business use case. Choose which fields appear, change the order, and mark certain fields as required.

Customize Post-Chat Surveys

Use Salesforce Feedback Management to gather feedback at the end of a chat session. When customers choose to participate at the end of a session, a survey opens within the chat window to complete.

Customize Additional Chat Branding

Set the base font size, width and height, and images for your Embedded Chat deployment.

Use Custom Components With Your Chat Window

Create custom components to control the user interface for your chat window.

Specify the Chat Settings for Your Embedded Chat Deployment

Embedded Chat uses a Chat deployment ID and button. The Chat configuration details that you provide are used to generate the chat code snippet that you add to your web pages.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

Before you specify the Chat settings, ensure that you have a Chat deployment and a Chat button available to use with Embedded Chat.

1. From Setup, enter *Embedded Service* in the Quick Find box, then select **Embedded Service**.
2. Select the Embedded Service deployment that you want to work with by clicking the arrow beside the deployment name and selecting **View**.
3. In the Embedded Service configuration page, go to the `Chat settings` section and click **Start**.
4. In the `Chat Deployment` menu, select the Chat configuration that you want to use with the chat window from the dropdown list.
5. In the `Chat Button` menu, select the Chat button or automated invitation that you want to use with the chat window from the dropdown list.
6. Select **Show queue position** if you want to display the customer's place in line while they wait to chat with a support agent or to be transferred. Make sure that the Chat button you selected has **Enable Queue** selected in your Chat button settings.
7. Click **Save**.

After you've provided the Chat settings, get the code to add the chat window to your web pages.

SEE ALSO:

Add Your Embedded Chat to a Website

EDITIONS

Embedded Service setup node is available in: Lightning Experience

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

USER PERMISSIONS

To set up Embedded Chat:

- Customize Application AND Modify Metadata

Customize the Embedded Pre-Chat Form

Gather contact information and learn about customer needs before chatting. Create a custom pre-chat form and associate Salesforce records like leads, cases, and contacts. You can also customize the field labels used on the form.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

Important: Before you start, make sure that the "Enable Support QuickAction Rest endpoint for Guest Users" permission is turned on. Without the permission, your customers can't request a chat.

EDITIONS

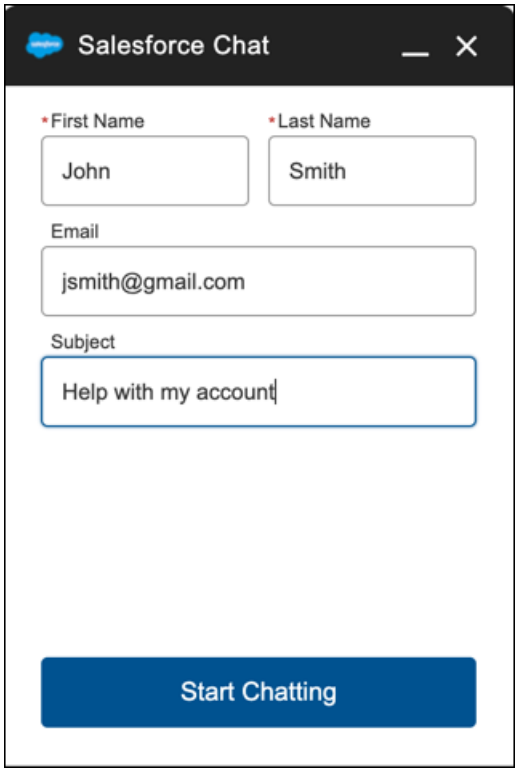
Embedded Service setup node is available in: Lightning Experience

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

USER PERMISSIONS

To set up Embedded Chat:

- Customize Application AND Modify Metadata



When you design your pre-chat form, select a use case that automatically associates information from the form to Salesforce records. When a customer enters their name or email address, Salesforce matches the information with a contact or lead record. If no match is found, a new record is created.

For example, if you select the Service scenario, Salesforce relates the customer's information with Case and Contact records.

Table 1: Use Cases to Associate Pre-Chat Information with Salesforce Records

If you select:	Then Salesforce associates the pre-chat information with these record types:
Sales	Lead records
Service	Contact records Case records
Basic	Contact records

You can select the fields that are shown on the pre-chat form, change the order of the fields, and make fields required. To make the form more user-friendly to customers, add up to four fields.

Important: Create and edit your pre-chat forms from Embedded Service setup, not at global action. Editing global action leads to unexpected results. Each time the pre-chat form is edited, the global action is recreated and prior updates to action details or predefined fields are reverted. Embedded Service setup creates quick actions specifically for pre-chat. Using quick actions for other purposes causes pre-chat setup to fail.

To set up the pre-chat form:

1. From Setup, in the Quick Find box, enter `Embedded Service`, and then select **Embedded Service**.
2. Select the Embedded Service deployment that you want to work with, click the arrow next to the deployment name, and then select **View**.
3. Click **Edit** next to Chat Settings.
4. In the Pre-chat section, move the radio button to **Active**.
5. Click **Edit**.
6. Select the use case for the pre-chat form.
   a. Select the main reason that customers use chat. The reason that you select affects the type of Salesforce record that's created from the information that customers enter into the pre-chat form.
   b. Select the record type that's created from the pre-chat form. The dropdown menu shows record types that are available on the object.
   c. Click **Next**.
7. Select the fields shown on the pre-chat form.
   a. To change the order of the fields, use the up arrow and down arrow.

      Note: For the Service scenario, fields are grouped by object. Contact fields always appear above Case fields.

   b. To add a field, click **+**. You can add only fields that are available on the object.
   c. To remove a field, click **X**.
   d. To require customers to fill out a field, select **Required** next to the field.
   e. Click **Save**.
8. Optionally, select a custom Lightning Component to replace the standard component. To learn more about custom components for pre-chat, see Lightning Web Components for Embedded Service in the *Embedded Service Developer Guide*.

Tip: Turn off a pre-chat form by viewing your deployment in Setup and moving the pre-chat page slider to **Inactive**.

To create custom labels for your form:

1. From Setup, in the Quick Find box, enter `Rename Tabs and Labels`, and then select **Rename Tabs and Labels**.
2. Set your primary deployment language from the Select Language dropdown list.
3. To rename a tab, select **Edit** next to the Contacts, Leads, or Cases based on your pre-chat objective. Select **Reset** to revert to the original name.
4. Enter the singular and plural forms of the new tab name. Also, if applicable for the language, select **Starts with a vowel sound** for labels that begin with a vowel to ensure the proper article (such as a or an). Then select **Next**. When you rename a tab or an object, you can't use the name of another standard tab, custom object, external object, or custom tab.
5. Enter your labels for the standard field labels. If necessary, enter both a singular and plural form for each label. Select **Starts with a vowel sound** for labels that begin with a vowel. Then save your work.

Important: If you're customizing the First and Last Name fields, complete each label separately. The First Name and Last Name labels (defined by Contacts and Leads in the Rename Tabs and Labels Setup) are used by the pre-chat form.

Note: You can use the pre-chat APIs to send nonstandard pre-chat details along with what's available in setup. For more information, see Expand the Pre-Chat Stage in the *Embedded Service Developer Guide*.

SEE ALSO:

Customize Embedded Chat Labels

Rename Object, Tab, and Field Labels

Customize the Embedded Chat Offline Form

Allow customers to submit a case form with Embedded Chat when your support agents are offline. Customize the form depending on your business use case. Choose which fields appear, change the order, and mark certain fields as required.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

Important: Before you start, enable Web-to-Case and turn on the "Enable Support QuickAction Rest endpoint for Guest Users" permission for your org. If the permission isn't set, your customers can't request a chat.

EDITIONS

Embedded Service setup node is available in: Lightning Experience

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

USER PERMISSIONS

To set up Embedded Chat:

- Customize Application AND Modify Metadata

To set up an offline support form:

1. From Setup, in the Quick Find box, enter `Embedded Service`, and then select **Embedded Service Deployments**.
2. To select the deployment that you want to work with, click the arrow next to the deployment name, and then select **View**. To create a new deployment, select the **New Deployment** button and following the steps.
3. Click **Edit** next to Chat Settings.
4. In the Offline support section, move the radio button to **Active**. If you see a warning about enabling guest access for your associated Salesforce site, click the link to enable it.

5. Click **Edit**.
6. Select a Case record type.
7. Select the fields shown on the form.
   a. To change the order of the fields, use the up arrow and down arrow.
   b. To add a field, click **+**. You can add only fields that are available on the object. Add up to 4 fields to make it user-friendly.

      Note: If you add a field with the `Unique` attribute and a visitor enters a duplicate field value, the case submission confirmation is displayed even though the case isn't created. Add unique fields to this form only if you're certain that visitors are unlikely to enter duplicate values.

   c. To remove a field, click **X**.
   d. To require customers to fill out a field, select **Required** next to the field.
8. Optionally, enter a URL for the offline support header image. The image appears in the header of the chat window when the customer opens the offline support form.
9. Save your changes.
10. Refresh your code snippet and paste it into your website.

Important: To turn off your offline support form at any time, view your deployment in Setup and move the Offline support slider to **Inactive**. Refresh and repaste the code snippet into your website each time you turn offline support on or off.

To create custom field names in your offline form, follow these steps:

1. From Setup, in the Quick Find box, enter *`Rename Tabs and Labels`*, and then select **Rename Tabs and Labels**.
2. Set your primary deployment language from the Select Language dropdown list.
3. Select **Edit** next to the Cases tab option. To revert to the original name, select **Reset**.
4. Enter the singular and plural forms of the new tab name. Also, if applicable for the language, select **Starts with a vowel sound** for labels that begin with a vowel to ensure the proper article (such as a or an). Then select **Next**. When you rename a tab or an object, you can't use the name of another standard tab, custom object, external object, or custom tab.
5. Enter your labels for the standard field labels. If necessary, enter both a singular and plural form for each label. Select **Starts with a vowel sound** for labels that begin with a vowel. Then save your changes.

Customize Post-Chat Surveys

Use Salesforce Feedback Management to gather feedback at the end of a chat session. When customers choose to participate at the end of a session, a survey opens within the chat window to complete.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

EDITIONS

Embedded Service setup node is available in: Lightning Experience

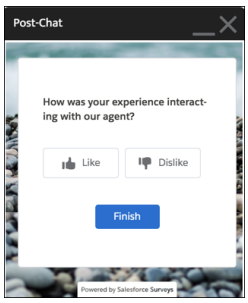
Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Note: This feature is available with the Salesforce Feedback Management license.

USER PERMISSIONS

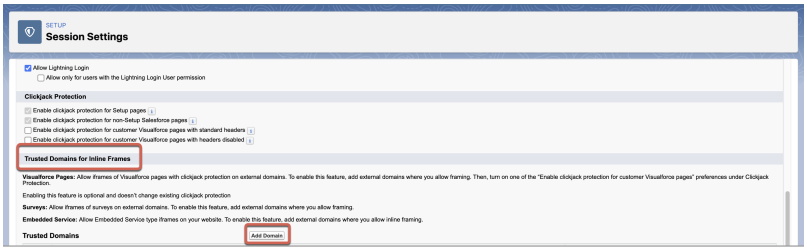
To set up Embedded Chat:
- Customize Application AND Modify Metadata

To select a post-chat survey:
- Salesforce Surveys Advanced Features



To open a post-chat survey within the chat window on your website, allow inline framing of surveys within the URL on the Session Settings page:

1. From Setup, in the Quick Find box, enter `Session Settings`, and then select **Session Settings**.
2. Navigate to the **Trusted Domains for Inline Frames** section and select **Add Domain**.



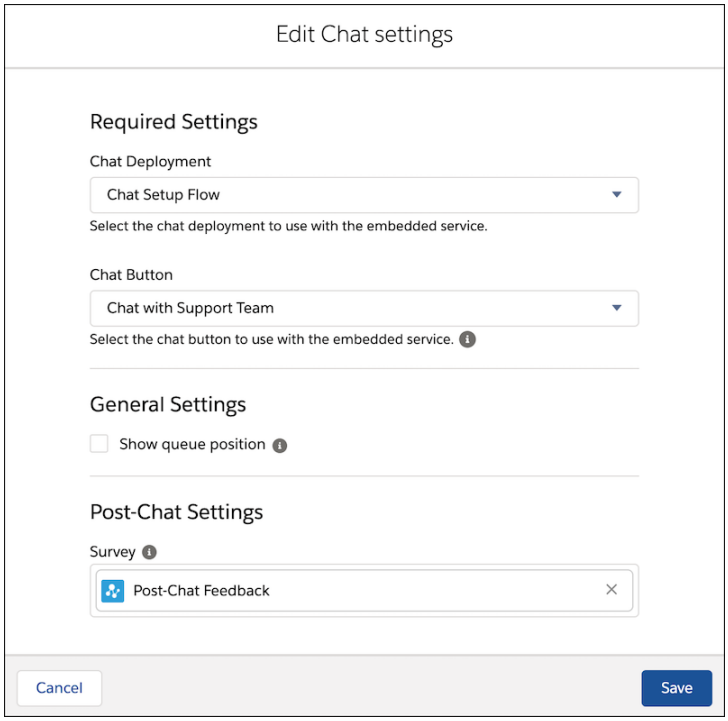
3. Enter your domain name and select **Surveys** as the IFrame Type.

   Use an accepted domain format: `*.example.com` or `https://example.com`.

4. **Save** your work.

Surveys must be enabled with at least one active survey. Only active surveys can be used to gather post-chat feedback.

1. From Setup, in the Quick Find box, enter `Embedded Service`, and then select **Embedded Service Deployments**.
2. To add a post-chat survey, select **View** by an existing embedded service deployment.
3. Click **Edit** next to Chat Settings.
4. In the Chat Setting header, click **Edit**.
5. In the Post-Chat Settings section, search and select any active survey.



6. Save your changes.

Note: : If your company domain hosts the chat, you must allow list it.

A survey invitation is generated each time a customer decides to provide feedback. If the pre-chat form is configured to associate customer information with Salesforce records, the survey invitation is also associated with the same Salesforce records.

SEE ALSO:

Salesforce Feedback Management

Customize Additional Chat Branding

Set the base font size, width and height, and images for your Embedded Chat deployment.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

1. From Setup, enter `Embedded Service` in the Quick Find box, then select **Embedded Service**.
2. Select the Embedded Service deployment that you want to work with by clicking the arrow beside the deployment name and selecting **View**.
3. Click **Edit** next to Chat Settings.
4. In the Additional branding section, click **Edit**.
5. In the Images & Dimensions tab, set your chat images:
   - Agent Avatar (40 px by 40 px)
   - Logo (36 px by 36 px)
   - Pre-Chat Background (320 px by 100 px)
   - Waiting State (250 px by 60 px)
6. Set the width and height of your chat window. If you don't make any changes, the default sizes of 320 px and 498 px are used.
7. Set the base font size for your chat window.
8. Click **Finish** to close the window.

EDITIONS

Embedded Service setup node is available in: Lightning Experience

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

USER PERMISSIONS

To set up Embedded Chat:
- Customize Application AND Modify Metadata

Customize Embedded Chat Labels

Create your own custom labels and manage the label language for your Embedded Chat deployment.

Translate Your Chat Window

The primary language for a chat window is set in your embedded code snippet. An extra step in Translation Workbench is required for custom labels or platform-only languages.

Customize Embedded Chat Labels

Create your own custom labels and manage the label language for your Embedded Chat deployment.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

Custom labels are created within an individual deployment. Each Chat State has a set of default labels to customize your support needs and to match your website's tone.

EDITIONS

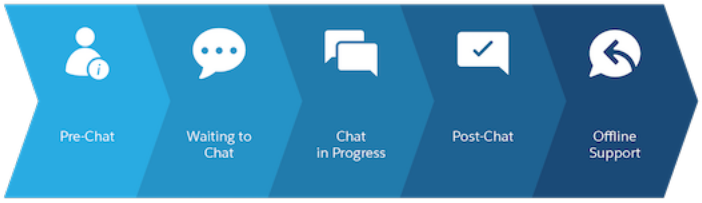
Embedded Service setup node is available in: Lightning Experience

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

USER PERMISSIONS

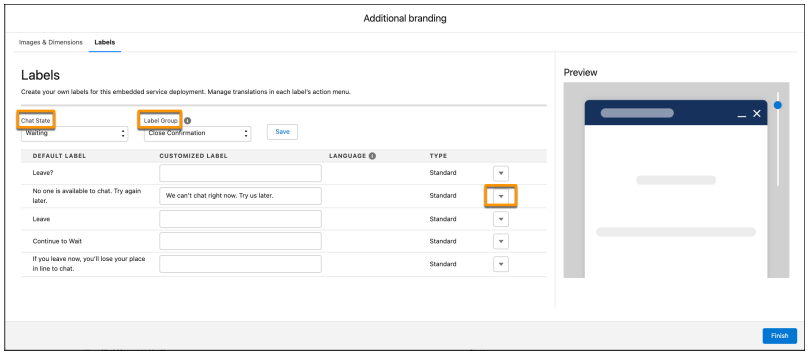
To set up Embedded Chat:

- Customize Application AND Modify Metadata



Also, a General section covers labels that appear throughout the chat process. To create your custom labels, follow these steps:

1. From Setup, in the Quick Find box, enter `Embedded Service`, and then select **Embedded Service Deployments**.
2. To select the Embedded Service deployment that you want to work with, click the arrow next to the deployment name, and then select **View**.
3. Click **Edit** next to Chat Settings.
4. In the Additional branding section, click **Edit**.
5. Click the **Labels** tab. Organized by Chat State, each Label Group focuses on a step in the chat window for the state.

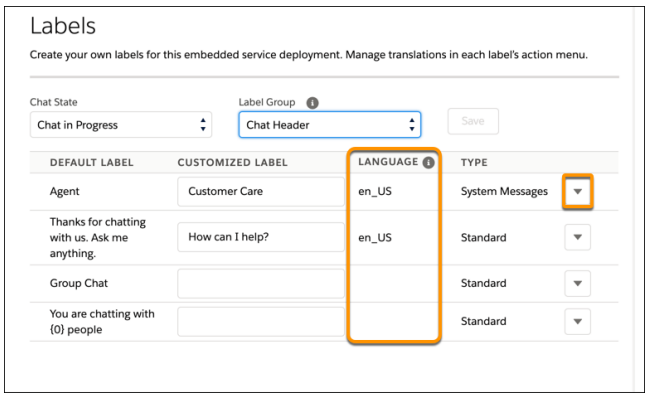


6. Select your Chat State and Label Group to create custom labels.
7. Under Customized Label, enter new labels for your embedded component. The preview highlights where the selected label appears in the chat window.

8. Save your customized labels before moving to another Chat State or Label Group.
9. To close the window, click **Finish**.

> **Important:** Some labels appear in multiple Chat States, so what you change can be reflected in other Label Groups. To return labels to the default choice, use the arrow at the end of each line.

The Language column indicates the language that each customized label was originally written in. To reset your customized labels, click the arrow next to the label type. Then select **Reset to Default**.



SEE ALSO:

- Translate Your Chat Window
- Customize the Embedded Pre-Chat Form
- Customize the Embedded Chat Offline Form

Translate Your Chat Window

The primary language for a chat window is set in your embedded code snippet. An extra step in Translation Workbench is required for custom labels or platform-only languages.

> **Important:** The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

EDITIONS

Embedded Service setup node is available in: Lightning Experience

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

Set your primary language in your Embedded Service code snippet. All of your chat window labels have a default translation set for any Salesforce fully supported or end-user language that you've added.

Follow these additional steps if you would like to create custom labels or if you want to include a platform-only language:

1. To translate labels for your chat window, view your Embedded Service deployment in Setup.
2. Open your Chat, Field Service or Flows settings, and click **Edit** next to Additional branding.
3. Click the **Labels** tab to view and customize the labels in your chat window.
4. Select the arrow next to a label and click **View Label Record**, where you can select to translate the label in the Translation Workbench.

Tip: For any platform-only language, you must use custom labels and set all the labels to that language.

Important considerations for translations:

- If no language is set in your code snippet or if it's invalid (due to an empty string, wrong format, or language not supported), the language of the **Site Guest User** for the Site of your Embedded Service deployment is used.
- When Translation Workbench is disabled, the language of the **Site Guest User** for the Site of your Embedded Service deployment is used.
- To ensure that translated labels are packaged in an Embedded Service deployment, in Package Manager setup add **Language Translations** as the Component Type. Select the languages you wish to package with the deployment and click **Add to package**.
- Not all languages are automatically available for Salesforce components. To use these end-user and platform-only languages, enable them in Languages Settings under **Company Settings**.

Tip: To use translation, you must allow cookies for your browser. For example, enable "Allow third-party cookies" in Safari iOS.

SEE ALSO:

Set the Language in the Code Snippet

Support Users in Multiple Languages

Use Custom Components With Your Chat Window

Create custom components to control the user interface for your chat window.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

EDITIONS

Embedded Service setup node is available in: Lightning Experience

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

SEE ALSO:

Lightning Web Components for Embedded Service for Web

Customize the General Branding for Your Chat Window

Select the colors and font used in your chat window to reflect your company's brand identity.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

Before you start, get the color codes for your company's brand from the team that maintains the website. You can specify branding colors by entering their Hex code or RGB code, or by selecting a color in the color palette.

Important: We don't support adding custom CSS to your embedded component. We support using only the branding options in setup and in the customizable parameters in the code snippet. If you have custom CSS in your embedded component, it's your responsibility to test your chat window each release to ensure that it functions properly.

EDITIONS

Embedded Service setup node is available in: Lightning Experience

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

USER PERMISSIONS

To set up Embedded Chat:

- Customize Application AND Modify Metadata

To customize the branding of your chat window:

1. From Setup, enter *Embedded Service* in the Quick Find box, then select **Embedded Service Deployments**.
2. Find the deployment that you want to work with, click the down arrow to the right and select **View**.
3. In the Embedded Service configuration page, go to the Branding section and click **Edit**.
4. Select the colors that you want to show in the chat window.

   Enter the color using the Hex code. To enter an RGB code or to select a color from the palette, open the color palette by clicking the down arrow in the color field.
5. Select the font that you want to use in the chat window.
6. Click **Finish**.

The screenshots show how the branding elements appear in the chat window.

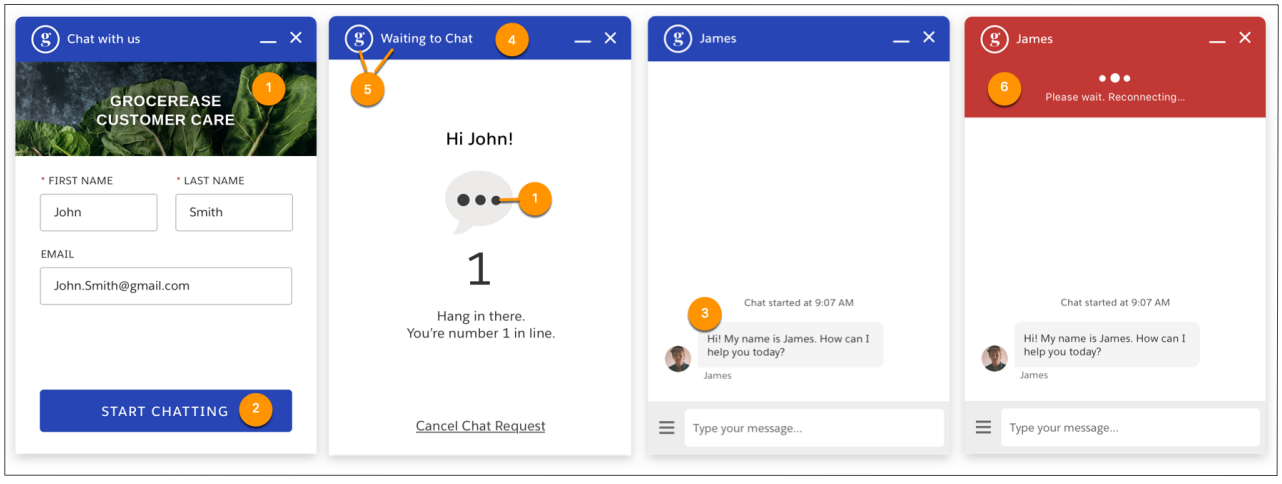


Table 2: Brand Elements That Can Be Customized in a Chat Window

Brand Element	Description
Brand Primary	Sets the color of the: • Pre-chat image backdrop • Pop message outlines • Loading animation • Loading animation when chat is minimized By default, the color is set to hex #222222 (Night Black). In the screenshots, brand primary elements are indicated by [1].
Brand Secondary	Sets the color of the: • Action buttons, such as "Start Chatting" or "Chat with an Expert" • Input field focus • Send button • Guest message chat payload • Secondary buttons • Chat when it is minimized and there's a new message By default, the color is set to #005290 (Nimbostratus Blue). In the screenshots, brand secondary elements are indicated by [2].
Contrast Primary	Sets the color of the: • Chat body text • Chat input text By default, the color is set to #333333 (dark gray). The background color is white. In the screenshots, contrast primary elements are indicated by [3].
Header	Sets the color of the: • Chat header bar • Chat button when the chat is minimized By default, the color is set to hex #222222 (Night Black). In the screenshots, the header element is indicated by [4].
Header Text	Sets the color of the: • Chat header text and icon By default, the color is set to #222222 (Night Black). In the screenshots, the header text element is indicated by [5].

Brand Element	Description
Header Alternate	Sets the color of the: • Chat state waiting header • Chat state header and announcement area By default, the color is set to #cb2b19 (red). In the screenshots, the header alternate is indicated by [6].
Font	Sets the font used in the chat widget. The following fonts are supported in mobile browsers for both iOS and Android: Georgia, Times New Roman, Arial, Courier New The following fonts are supported in mobile browsers for iOS only: Trebuchet MS, Verdana, and Lucida Console Note: If you select a font that isn't available for a chat visitor's mobile browser, the browser's default font is displayed.

Important: Changing the branding of an existing chat window changes the code snippet for the chat window. If you update the branding for an existing window, you must update the code snippet on your web pages. If you customized the code snippet, you must add those customizations to the new code snippet.

Customize Chat Window Branding for a Experience Site

Select the chat window colors and font to reflect your company's brand identity on an Experience Cloud site.

SEE ALSO:

Add Your Embedded Chat to a Website

Customize the Embedded Service Code Snippet

Customize Additional Chat Branding

Customize Chat Window Branding for a Experience Site

Select the chat window colors and font to reflect your company's brand identity on an Experience Cloud site.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

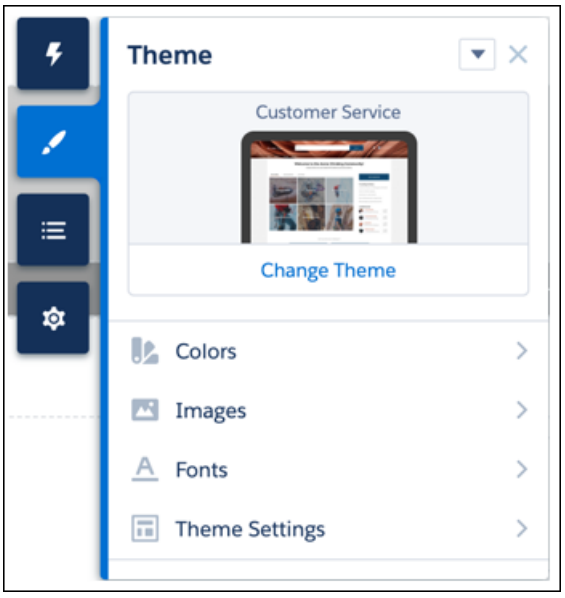
To customize chat window branding colors, images, and fonts on an Experience site, use the settings in the Embedded Chat component.

EDITIONS

Embedded Service setup node is available in: Lightning Experience

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

1. From the Experience Builder make sure that your chat window is selected. Navigate to the Theme tab.



2. Select the colors for your chat window.

   Note: For Experience sites, Action Color maps to the Brand Primary and Brand Secondary colors for web chat windows. Text Color maps to the Contrast Primary color for a web chat window.

3. Select your font and add images to your chat window:
   - Agent Avatar (recommend 40 px by 40 px)
   - Logo (recommend 36 px by 36 px)
   - Pre-Chat Background (recommend 320 px by 100 px)
   - Waiting State (recommend 250 px by 60 px)

4. **Save** your site.

   Note: These branding tokens are created in Embedded Service Setup and automatically pass to your site: width, height, header text, and header alternate.

SEE ALSO:

General Embedded Service Branding

Add Your Embedded Chat to a Website

Salesforce generates a unique Embedded Service code snippet from the information you provide during setup. Copy and paste the code snippet to include chat on your web pages. Add the optional meta tag code snippet to improve performance for both mobile and desktop.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

Note: Before you start, make sure you have access to the company's web pages. After adding the code snippet, customers can see and use the chat window. If you're not ready for them to access chat, include the code snippet on a private web page first.

Important: Avoid modifying the code snippet, which impacts how chat is loaded and whether conversations continue as users navigate from page to page.

EDITIONS

Embedded Service setup node is available in: Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions with Service Cloud or Sales Cloud

USER PERMISSIONS

To set up Embedded Chat:

- Customize Application AND Modify Metadata

Here are the steps to add your code snippet:

1. From Setup, enter *Embedded Service* in the Quick Find box, then select **Embedded Service**.
2. Select the Embedded Service deployment that you want to work with by clicking the arrow beside the deployment name and selecting **View**.
3. In the Embedded Service configuration page, go to the `Embedded Service code snippets` section and click **Get Code**.
4. Copy the chat code snippet and paste it immediately above the closing `</body>` tag on your web page, adding Embedded Chat to the page.

   Paste the chat code snippet into every web page where you want the chat window to appear. Don't place your code snippet in a header or on the same web page as a Chat button. Optionally, customize the chat code snippet.
5. Optionally, copy and paste the meta code snippet into the `<meta>` tag head section in every web page where you want the chat window to appear.
6. Click **Done**.

   The code snippets page closes.

Note: The `<meta>` tag code makes your web page responsive, so the chat window look good and performs well on different devices. For example, if a customer views your web page and chat window on a mobile phone, both are resized.

Important: If there are issues with behavior on other devices, update your `<meta>` tag to the code snippet that's provided. This tag is added to the head section of every page where the chat window is used. If you don't include an appropriate `<meta>` tag, the chat window appears as it does on a desktop, regardless of the user's device used to access the page.

Tip: If you're adding your Chat window to a Visualforce page, include these attributes on your `apex:page` tag to keep styling consistent with your web pages:

```
<apex:page showheader="false" sidebar="false" standardStylesheets="false">
```

Customize the Embedded Service Code Snippet

Use a text or HTML editor to customize the optional parameters in the Embedded Service code snippet.

Add Your Website to the CORS Allowlist

Add the URLs of the web pages where you intend to add the chat window to the CORS allowlist in your org. The web page where you add the chat window is the page that customers use to access chat.

SEE ALSO:

Customize the Embedded Service Code Snippet

Customizable Parameters in the Embedded Chat Code

Test Your Embedded Chat Window

Customize the Embedded Service Code Snippet

Use a text or HTML editor to customize the optional parameters in the Embedded Service code snippet.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

You can customize certain parameters that affect the appearance and behavior of the chat window. We respect the code snippet before your Embedded Chat settings when determining the chat window's appearance and behavior.

You can customize the code snippet once and paste it on all your web pages, or customize the code snippet for certain web pages.

EDITIONS

Embedded Service setup node is available in: Lightning Experience

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

USER PERMISSIONS

To set up Embedded Chat:

- Customize Application AND Modify Metadata

1. From Setup, enter *Embedded Service* in the Quick Find box, then select **Embedded Service**.
2. Select the Embedded Service deployment that you want to work with.
3. In the Embedded Service configuration page, go to the `Embedded Service Code Snippets` section and click **Get Code**.
4. Copy the code snippet to a text editor or HTML editor.
5. In the text editor or HTML editor, specify the image URLs for the chat window parameters.
6. Save your changes.
7. Go to the Experience Builder for your help center.
8. Below components, click **Settings**.
9. In the navigation tree, click **Security**.
10. Under `Content Security Policy`, select **Allow inline scripts and script access to allowlisted third-party hosts** in the `Script Security Level` dropdown.
11. In the code snippet, find the field `baseLiveAgentContentURL` and copy the URL.
12. Click **Add Trusted Site** and name your new site.
13. Click **Add Site** and paste the URL from `baseLiveAgentContentURL`.
14. Publish your site changes.

After modifying the code snippet, add the code to every web page where you want the chat window to appear.

Customizable Parameters in the Embedded Chat Code

You can customize certain parameters that affect the appearance and behavior of the chat window so that the chat experience reflects your company's branding. Use these parameters to customize the pre-chat background image, logo, waiting-state image, and your agent's avatar picture. You also can customize the wording that appears on the chat button and on the chat window when the chat is loading, when agents are online, when agents are offline.

SEE ALSO:

Add Your Embedded Chat to a Website

Customizable Parameters in the Embedded Chat Code

Customizable Parameters in the Embedded Chat Code

You can customize certain parameters that affect the appearance and behavior of the chat window so that the chat experience reflects your company's branding. Use these parameters to customize the pre-chat background image, logo, waiting-state image, and your agent's avatar picture. You also can customize the wording that appears on the chat button and on the chat window when the chat is loading, when agents are online, when agents are offline.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

EDITIONS

Embedded Service setup node is available in: Lightning Experience

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

The following parameters are customizable in the code snippet only.

- Set the domain for your chat window to persist across subdomains: `embedded_svc.settings.storageDomain = "..."`
- Set the language: `embedded_svc.settings.language = "..."`
- Display the default chat button: `embedded_svc.settings.displayHelpButton = "..."`
- Set a routing order: `embedded_svc.settings.fallbackRouting = ["...", "..."]`
- Use direct-to-button routing: `embedded_svc.settings.directToButtonRouting = function(prechatFormData) {"..."};`
- Set a chatbot avatar image: `embedded_svc.settings.chatbotAvatarImgURL = "..."`
- Load files for custom chat events:
  - `embedded_svc.settings.externalScripts = ["...", "..."]`
  - `embedded_svc.settings.externalStyles = ["...", "..."]`
- Pass nonstandard pre-chat details: `embedded_svc.settings.extraPrechatFormDetails = ["..."];`

The following parameters are customizable in both the code snippet and Embedded Service setup. Settings in your code snippet override what you enter in Embedded Service setup.

- Set the chat window size and base font size:
  - `embedded_svc.settings.widgetWidth = "..."`
  - `embedded_svc.settings.widgetHeight = "..."`
  - `embedded_svc.settings.widgetFontSize = "..."`

- Set the chat images:
  - `embedded_svc.settings.prechatBackgroundImgURL = "..."`
  - `embedded_svc.settings.smallCompanyLogoImgURL = "..."`
  - `embedded_svc.settings.waitingStateBackgroundImgURL = "..."`
  - `embedded_svc.settings.avatarImgURL = "..."`
- Customize the text on the chat window:
  - `embedded_svc.settings.defaultMinimizedText = "..."`
  - `embedded_svc.settings.disabledMinimizedText = "..."`
  - `embedded_svc.settings.offlineSupportMinimizedText = "..."`
  - `embedded_svc.settings.loadingText = "..."`

SEE ALSO:

Add Your Embedded Chat to a Website

Customize the Embedded Service Code Snippet

Embedded Service for Web Developer Guide

Add Your Website to the CORS Allowlist

Add the URLs of the web pages where you intend to add the chat window to the CORS allowlist in your org. The web page where you add the chat window is the page that customers use to access chat.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

You can use HTTP and HTTPS domains with Embedded Chat. The protocol for the URL that you allowlist must match the URL in the website or Experience site endpoint in the code snippet that's generated at the end of Embedded Service setup.

Important: This information applies to Embedded Service only.

EDITIONS

Embedded Service setup node is available in: Lightning Experience

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

USER PERMISSIONS

To set up Embedded Chat:
- Customize Application AND Modify Metadata

1. From Setup, enter *CORS* in the Quick Find box, then select **CORS**.
2. Select **New**.
3. Enter an origin URL pattern.

   The origin URL pattern must include the HTTP or HTTPS protocol and a domain name. The origin URL pattern can include a port. The wildcard character (*) is supported and must be in front of a second-level domain name. For example, `https://*.example.com` adds all subdomains of `example.com` to the allowlist.

   The origin URL pattern can be an IP address. However, an IP address and a domain that resolve to the same address are not the same origin, and you must add them to the CORS allowlist as separate entries.

Add Your Chat Window to an Experience Site

Use the Embedded Service component to add your chat window to an Experience Builder site. Create a snippet settings file to use snippet-only settings with your site's chat window.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

EDITIONS

Embedded Service setup node is available in: Lightning Experience

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

There are some differences to keep in mind when adding your chat window to an Experience site instead of a website:

- Don't use your Embedded Service code snippet. Use the Embedded Service component.
- To ensure that chat sessions are continued across pages, place the Embedded Service component in your site's template footer.
- Use the settings in the component. What you set in the component overrides what you select in Embedded Service setup.
- To create and automate invitations, see Add Chat Settings to an Experience Site.
- You can't fill in pre-chat fields for logged-in users via the code snippet. Use the component option, which fills in contact fields (name and email).
- To customize Embedded Chat branding colors in a site, use the settings in the component. Using Embedded Service setup to specify branding colors for Experience sites isn't supported.
- To ensure end-user session continuity, your web pages must have the same `storageDomain` and use the same deployments. To share a custom site domain between the site and other pages in the same domain, use the Storage Domain property in the component.
- If you create a sandbox org and copy your chat component from an Experience Builder production environment, set your new deployment to one available only in your sandbox.

To use Embedded Service code snippet settings in your site, use the **Snippet Settings File** field in the component.

1. Create a JavaScript file with your Embedded Chat settings. Refer to the Embedded Service for Web Developer Guide for more details.
2. Upload the file to your Static Resources. Give the file a name that's easy to remember and doesn't include spaces.
3. In your Embedded Service component settings, enter the static resource name (not the file name) in the **Snippet Settings File** field.
4. Save and publish your changes.

Add Chat Code Settings to an Experience Site

Use the Embedded Service component to add a Code Setting Name to an Experience Cloud site. This step allows you to modify with static resources, including inviting customers to chat.

SEE ALSO:

Embedded Chat Experience Component

Customize Chat Window Branding for a Experience Site

Add Chat Code Settings to an Experience Site

Use the Embedded Service component to add a Code Setting Name to an Experience Cloud site. This step allows you to modify with static resources, including inviting customers to chat.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

Important: Code setting modifications for Embedded Chat external pages and websites are not available.

Adding JavaScript code as a static resource provides the flexibility of changing your customer interactions, including adding invitations to chat, or other settings to a site. Before creating your Embedded Chat code settings, you must create and upload embedded static resources

To add Embedded Chat code settings to your site, use the `Code Settings Name` field in the component.

EDITIONS

Embedded Chat setup node is available in: Lightning Experience

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

USER PERMISSIONS

To set up static resources:

- Customize Application AND Author Apex

1. Select `Add Code Setting`.
2. Create a JavaScript file with your Embedded Chat settings for sites, including strings, arrays, JSON, and js functions. (See examples of Experience Site Static Resource Files.)
3. In your Embedded Service component settings, enter the static resource name (not the file name) in the `Code Setting Name` field on a specific page on your site.
4. Save and publish your changes.

Note: If you load static resource files for Experience sites, test your entire flow in the builder to discover any CSP errors. This step ensures that image URLs and assets from other domains specified in the static resource load during run time. Check the console first for CSP errors if something is missing.

Set Up Appointment Management

Appointment Management gives your users an easy way to schedule, modify, and cancel appointments. The experience is powered by Flow Builder. You craft the perfect interaction and decide when and how to create related records like work orders.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

Important: Enable Field Service and install the latest version of the Field Service managed package before setting up Appointment Management.

EDITIONS

Service Set up node available in: Lightning Experience

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

USER PERMISSIONS

To configure the Field Service managed package:
- Customize Application

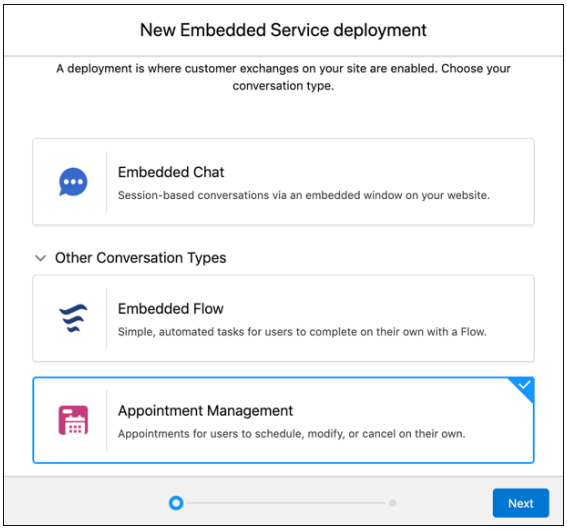
To create a permission set:
- Manage Profiles and Permission Sets

1. Update your Field Service permission sets.
   a. In the Field Service Admin app, on the Field Service Settings tab, Click **Permission Sets**.
   b. Ensure that the FSL Self Service permission set is up to date.
      If you don't have the FSL Self Service permission set in the Field Service managed package, you don't have the latest version.
   c. Give users the self-service permission set that comes with the Field Service package.
2. Create a scheduling flow with actions for your users to follow. You can create separate flows to book a new appointment, modify an existing appointment, or cancel an upcoming appointment.
   For a new appointment flow:
   a. Collect details from your users for a Work Order and Service Appointment and query the `getSlots` method in the Field Service Managed Package. Use an Apex class to query the method.
   b. For your query, you need the Service Appointment ID, the Scheduling Policy ID, and the Operating Hours ID.
   c. Serialize `getSlots` results into a JSON string and pass that as an input to the Appointment Scheduling Flows component to display the available slots for your user to select.
   d. With the user selected slot, attempt to book a new appointment by using the `ScheduleService.Schedule` method in the Field Service Managed Package. Record any success or failure messages returned from the method and add a loop in your flow to retry booking or display an error message.
   e. Save the flow as Embedded Appointment Management Flow type and Activate it. Use the flow in step 6.g.
3. For a modify appointment flow:
   a. The Embedded Service application passes a service appointment ID to your modify appointment flow.
   b. With the ID, look up the existing service appointment record. Add logic in your flow to modify details like the description or the time for the appointment by accepting relevant information from the user.
   c. You can again use the `getSlots` method with the Appointment Scheduling component to let your users pick a new time.
   d. Save the flow as Embedded Appointment Management Flow type and Activate it. Use the flow in step 6.g.
4. For a cancel appointment flow:
   a. The Embedded Service application again passes a service appointment ID to your cancel appointment flow.

b. Using the ID, look up the service appointment record and set its Status to Canceled to cancel the appointment.

c. Save the flow as Embedded Appointment Management Flow type and Activate it. Use the flow in step 6.g.

5. To create and configure a connected app, follow the Connected Apps instructions. For authentication, use the Authenticate Apps with OAuth help.
6. Create and configure an Embedded Service deployment.
   a. From Setup, in the Quick Find box, enter `Embedded Service`, then select **Embedded Service Deployments** and click **New Deployment**.
   b. Choose **Appointment Management** as your conversation type and **Next**.



Note: If you don't see the selection, Appointment Booking isn't enabled.

   c. On the New Embedded Service Deployment panel, in `Embedded Service Deployment Name` enter your appointment booking name.

Note: The deployment is shown in a list on the Embedded Service page. If you have several deployments, use a descriptive name so it's easy to distinguish each one.

   d. The `API Name` field is automatically populated.
   e. Under `Site Endpoint`, select the site you want to use and save your changes.
   f. In `Embedded Service Deployment Settings`, click **Start** next to the `Appointment Management settings`.
   g. Optionally, select **Let Customers Schedule New Appointments** and **Let Customers View Their Appointments** on the Appointment Management Settings page. Save your changes.
   h. In the Scheduling flows section, click **Edit**.
   i. To set the flows, use the new versions you created in the previous steps for New Appointment, Modify Appointment, and Cancel Appointment.
   j. For authorization with an Experience site login, see Connected Apps. Refer to Authorize Apps with OAuth for custom login methods.
   k. To customize the images used in the flow, click **Edit** under `Additional Branding`. Enter your Logo Image URL.

l. To go back to the Deployment home, click **APPOINTMENT_BOOKING** at the top of the page.

m. To activate the embedded service, click the slider to **Deployment header**.

n. Click **Get Code** next to embedded code snippets.

o. On the Embedded Service Code Snippets page, click **Copy to Clipboard**, save the code snippets for later, and click **Done**.

7. Create a Cross-Origin Resource Sharing (CORS) record.

   a. From Setup, in the Quick Find box, enter `CORS`, then select **CORS** and click **New**.

   b. Enter the correct URL domain of the site where you place the embedded service, `https://MyDomainName.my.site.com`, and save your work.

Note: To access or view appointments with Appointment Management, the user must have a Customer Community User profile with the FSL Self Service permissions set.

Note: Appointment Management can also be set up with Lightning Scheduler. See the Set Up Scheduling for Authenticated External Users instructions first to meet any prerequisites before following the previous steps. Add Inbound Flows to a Website provides more background.

Add an Embedded Flow to Automate a Task

Use an embedded screen flow to help customers or users complete simple, automated tasks. An embedded flow is also guest-user friendly because users don't need to log in to access the flow.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

Important: Legacy Embedded Flows is also scheduled for retirement on June 17, 2025. You can continue using this product until then. After the retirement, use an Experience Cloud site to create and embed preexisting flows.

Before you begin, create a flow using Flow Builder. Then add it to an Embedded Service deployment.

Embedded Service supports only flows of the Screen Flow type. Users don't need to log in to access an embedded flow.

1. From Setup, in the Quick Find box, enter `Embedded Service`, and then select **Embedded Service Deployments**.
2. Click **New Deployment**.
3. Select **Embedded Flow** as your conversation type, and click **Next**.

EDITIONS

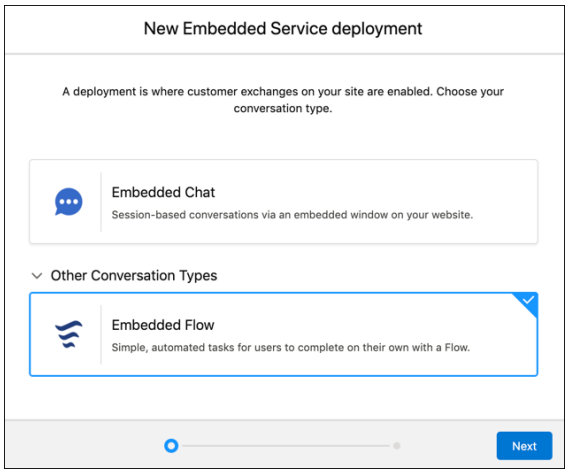
Embedded Service setup node is available in: Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions with Service Cloud and Sales Cloud

USER PERMISSIONS

To set up embedded flows:
- Customize Application AND Modify Metadata

To create, update, or delete deployments:
- Customize Application AND Modify Metadata



4. On the New Embedded Service deployment panel, in the `Embedded Service Deployment Name` field enter your flow name.

   Tip: The deployment is shown in a list on the Embedded Service page. If you have several deployments, use a descriptive name so it's easy to distinguish each one.

5. The `API Name` field is automatically populated.
6. Under `Site Endpoint`, select the site you want to use and save it.
7. In Embedded Service Deployment Settings, click **Start** next to the `Flow Settings`.
8. Under Flow Name, select the flow you want to use in your Embedded Service deployment and save it.
9. In Additional branding, edit image dimensions and labels, and click **Finish**.

Example:

- An insurance company can create an embedded screen flow so that policyholders can search for in-network medical providers without logging in.
- An online retailer can create an embedded screen flow to streamline how new customers create accounts.

To use a different flow in this Embedded Service deployment or to add custom labels, click **Edit** next to Flow Settings.

SEE ALSO:

Create an Embedded Service Deployment

Flow Builder Tour

Flow Types

Create Embedded Static Resources

Create and upload JavaScript files to Salesforce as static resources and then apply them to your website or Experience Builder site. With static resources, you can change one or many pages with a few clicks in code settings on existing deployments.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

Note: The Lightning Platform manages and distributes the static resources, acting as a content distribution network. Caching and distribution are handled automatically.

Create your JavaScript static resource files, including strings, arrays, JSON, and JS functions. Here are some examples to get you started:

Channel Menu Static Resources

Experience Site Static Resources

Then upload JavaScript files to reference in your Channel Menu Experience site or Channel Menu external website deployment. Add files for Embedded Chat to Experience sites only (not available for website pages).

Here are the steps to add your resources:

1. From Setup, in the Quick Find box, enter `Static Resources`, then select **Static Resources**.
2. Click **New**.
3. To identify the resource in Visualforce markup, enter text in the Name box. Use this static resource name, not the file name, for reference later in code settings.

   Create a name that's unique in your org, and provide a different name for each static resource. Start with a letter, use alphanumeric characters, and no spaces. Don't end with an underscore or add two consecutive underscores.
4. In the Description text area, specify an optional description of the resource.
5. To navigate to a local copy of the resource that you want to upload, click **Browse** next to File.

   Note: A single static resource file is up to 5 MB, and an organization can have up to 250 MB of total static resources.
6. Set the Cache Control to **Public**. Share the static resource data on the Salesforce server with other users in your organization for faster load times.
7. To apply the resources, return to your code settings in Channel Menu or Experience sites.

   Tip: If you make changes, you can reupload static resources without changing your code settings name in setup or in the code snippet and property builder.

Important: If you change your Channel Menu setup, the results can take up to 10 minutes to appear for the first time on your website due to caching.

Note: For Chat website pages, see Customize an Automated Invitation.

EDITIONS

Setup node is available in: Lightning Experience

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

USER PERMISSIONS

To create and edit static resources:

- Customize Application AND Author Apex

Experience Site Static Resource Files
Create JavaScript files to upload as a static resource and apply to Embedded Chat for your Experience Cloud site. Here's a code example for chat invitations to get you started.

SEE ALSO:

Channel Menu Code Settings
Chat Code Settings for Experience Sites

Experience Site Static Resource Files

Create JavaScript files to upload as a static resource and apply to Embedded Chat for your Experience Cloud site. Here's a code example for chat invitations to get you started.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

Here is sample JavaScript code to upload as a static resource and invite your customers to chat on your site pages (not available for external web pages). To create the file:

1. Specify `_invitationResource` on the Global window object

   ```
   window._invitationResource = (function() {})();
   ```

   Tip: The contents of the file are loaded on to window object. Make your CSS classes and HTML ids as specific as possible.

2. Set invitation HTML to `embedded_svc.config.invitation.htmlMarkup`
3. Set invitation CSS to `embedded_svc.config.invitation.styles`
4. Set invitation JS to `embedded_svc.config.invitation.invitationAPIs`

```
window._invitationResource = (function() {
    console.log("Chat invitation file loaded");

    embedded_svc.config.invitation.htmlMarkup = '';
    // Insert invitation's custom HTML markup

    embedded_svc.config.invitation.styles = '';
    // Insert invitation's CSS styles (be specific!)

    embedded_svc.config.invitation.invitationAPIs = '';
    // Utilize these existing methods to accept/reject invites:
    // - embedded_svc.inviteAPI.inviteButton.acceptInvite()
    // - embedded_svc.inviteAPI.inviteButton.rejectInvite();
})();
```

Here is the full code sample for a standard chat invitation. Add your image URL link for the `embeddedServiceAvatar`.

```
window._invitationResource = (function() {
    console.log("Invitation loaded");
```

EDITIONS

Channel Menu setup node is available in: Lightning Experience

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

USER PERMISSIONS

To create and edit static resources:

- Customize Application AND Author Apex

```
    embedded_svc.config.invitation.htmlMarkup =
        '<div class="embeddedServiceInvitation" id="snapins_invite" inert="true"
aria-live="assertive" role="dialog" aria-atomic="true">'+
          '<div class="embeddedServiceInvitationHeader" aria-labelledby="snapins_titletext"
 aria-describedby="snapins_bodytext">'+
                '<img id="embeddedServiceAvatar" src="https://yourimagehere.png"></img>'+

                '<span class="embeddedServiceTitleText" id="snapins_titletext">Need
help?</span>'+
                '<button type="button" id="closeInvite" class="embeddedServiceCloseIcon"
aria-label="Exit invitation">&times;</button>'+
            '</div>'+
            '<div class="embeddedServiceInvitationBody">'+
                '<p id="snapins_bodytext">How can we help you?</p>'+
            '</div>'+
            '<div class="embeddedServiceInvitationFooter"
aria-describedby="snapins_bodytext">'+
                '<button type="button" class="embeddedServiceActionButton"
id="rejectInvite">Close</button>'+
                '<button type="button" class="embeddedServiceActionButton"
id="acceptInvite">Start Chat</button>'+
            '</div>'+
        '</div>';

    embedded_svc.config.invitation.styles =
        '#snapins_invite { background-color: #FFFFFF; font-family: "Arial", sans-serif;
overflow: visible; border-radius: 8px;}'+
        '.embeddedServiceInvitation { background-color: transparent; max-width: 290px;
max-height: 210px; -webkit-box-shadow: 0 7px 12px rgba(0,0,0,0.28); -moz-box-shadow: 0 7px
 12px rgba(0,0,0,0.28); box-shadow: 0 7px 12px rgba(0,0,0,0.28); }'+
        '@media only screen and (min-width: 48em) { /*mobile*/ .embeddedServiceInvitation
 { max-width: 332px; max-height: 210px; } }'+
        '.embeddedServiceInvitation > .embeddedServiceInvitationHeader { width: inherit;
line-height: 32px; padding: 10px; color: #FFFFFF; background-color: #222222; overflow:
initial; display: flex; justify-content: space-between; align-items: stretch;
border-top-left-radius: 8px; border-top-right-radius: 8px; }'+
        '.embeddedServiceInvitationHeader #embeddedServiceAvatar { width: 32px; height:
32px; border-radius: 50%; }'+
        '.embeddedServiceInvitationHeader .embeddedServiceTitleText { font-size: 18px;
color: #FFFFFF; overflow: hidden; word-wrap: normal; white-space: nowrap; text-overflow:
ellipsis; align-self: stretch; flex-grow: 1; max-width: 100%; margin: 0 12px; }'+
        '.embeddedServiceInvitationHeader .embeddedServiceCloseIcon { border: none;
border-radius: 3px; cursor: pointer; position: relative; bottom: 3%; background-color:
transparent; width: 32px; height: 32px; font-size: 23px; color: #FFFFFF; }'+
       '.embeddedServiceInvitationHeader .embeddedServiceCloseIcon:focus { outline: none;
 }'+
       '.embeddedServiceInvitationHeader .embeddedServiceCloseIcon:focus::before { content:
 " "; position: absolute; top: 11%; left: 7%; width: 85%; height: 85%; background-color:
rgba(255, 255, 255, 0.2); border-radius: 4px; pointer-events: none; }'+
        '.embeddedServiceInvitationHeader .embeddedServiceCloseIcon:active,
.embeddedServiceCloseIcon:hover { background-color: #FFFFFF; color: rgba(0,0,0,0.7);
opacity: 0.7; }'+
        '.embeddedServiceInvitation > .embeddedServiceInvitationBody { background-color:
```

```
#FFFFFF; max-height: 110px; min-width: 260px; margin: 0 8px; font-size: 14px; line-height:
 20px; overflow: auto; }'+
        '.embeddedServiceInvitationBody p { color: #333333; padding: 8px; margin: 12px 0;
 }'+
        '.embeddedServiceInvitation > .embeddedServiceInvitationFooter { width: inherit;
color: #FFFFFF; text-align: right; background-color: #FFFFFF; padding: 10px;
border-bottom-left-radius: 8px; border-bottom-right-radius: 8px; }'+
       '.embeddedServiceInvitationFooter > .embeddedServiceActionButton { font-size: 14px;
 max-height: 40px; border: none; border-radius: 4px; padding: 10px; margin: 4px; text-align:
 center; text-decoration: none; display: inline-block; cursor: pointer; }'+
        '.embeddedServiceInvitationFooter > #acceptInvite { background-color: #005290;
color: #FFFFFF; }'+
        '.embeddedServiceInvitationFooter > #rejectInvite { background-color: #FFFFFF;
color: #005290; }';

    embedded_svc.config.invitation.invitationAPIs =
        '(function() {'+
            'document.getElementById("closeInvite").addEventListener("click", function()
{ embedded_svc.inviteAPI.inviteButton.rejectInvite(); });'+
            'document.getElementById("rejectInvite").addEventListener("click", function()
 { embedded_svc.inviteAPI.inviteButton.rejectInvite(); });'+
            'document.getElementById("acceptInvite").addEventListener("click", function()
 { embedded_svc.inviteAPI.inviteButton.acceptInvite(); });'+
           'document.addEventListener("keyup", function(event) { if (event.keyCode == 27)
 { embedded_svc.inviteAPI.inviteButton.rejectInvite(); }})'+
        '})();';
})();
```

When your file is complete, return to Embedded Static Resources for static resource upload instructions.

Test Your Embedded Chat Window

After adding Embedded Chat to your web pages or Experience Builder site, check the customer's experience.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

To test the chat window from the agent's perspective, log in to Salesforce as a Chat user with all the Chat support agent permissions.

EDITIONS

Embedded Service setup node is available in: Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions with Service Cloud or Sales Cloud

1. Open the Service Console app and use the Omni-Channel utility bar (at the bottom of the console if installed). Go online as the agent assigned to the chat button.

2. Test the chat from your customer's perspective. In another window or tab, navigate to a web page where you've placed the code snippet. If the chat button is available, click it to start the chat.

   The chat button is available as long as:

   - The number of chats doesn't exceed the button's queue size.
   - At least one agent has available capacity and the required skills listed in the associated routing configuration.

3. Verify that your pre-chat form and chat window open and appear as expected after initiating a chat session on the customer side.
4. In the console, verify that you receive a chat request when the chat is initiated.

If the chat window checks out, your customers can start using Embedded Chat to communicate with your agents.

SEE ALSO:

Permissions for Chat Support Agents

Add the Omni-Channel Utility

Embedded Chat Limitations

Before setting up Embedded Chat, consider general guidelines, limitations, and cookies and customer browser restrictions.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

EDITIONS

Embedded Service Setup is available in: Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions with Service Cloud or Sales Cloud

Feature Availability

- Embedded Chat is supported in Service Cloud and Sales Cloud with an add-on Digital Engagement license.
- Embedded Chat isn't supported on the Salesforce mobile app.

General

- You can't use Embedded Chat on the same web page or Experience Cloud site as a legacy Chat window.
- The Embedded Chat client is written in Aura, which isn't compatible with LWR templates.
- You can't use Embedded Chat and Appointment Management on the same web page or Experience Cloud site.
- When you host your Embedded Chat deployment on a Visualforce page, you must access the page using HTTPS. If you use HTTP, the chat window doesn't load.
- More than one code snippet added to one web page isn't supported with Embedded Chat.
- You can't add Embedded Chat to Lightning components. Use a Visualforce page, Experience Cloud site, or web property only.
- Field-level validation rules aren't supported with Embedded Chat.
- The visitor's recently viewed pages aren't updated during a chat. Agent can see only the pages the visitor views before the chat starts.

- No limit applies to the number of chats happening simultaneously in a Salesforce org. You have a limit of 40,000 chats per hour and per org. You also have a limit of 7,000 agents online per org simultaneously.
- If a user opens the chat in a browser tab, then navigates to a different browser tab, customer time-out can stop working. As a result, the chat conversation doesn't automatically end after the specified period. If customer time-out stops working in this scenario, encourage customers to maintain the chat in a separate browser window.

Settings and Pre-Chat

- The embedded chat window title shows the agent's name instead of the Chat Window Title setting. When you enable this setting in the Chat Configuration, it doesn't work in the chat window.
- To use a chat button for your embedded chat window, make sure the Prechat Form Page and Prechat Form URL fields are blank. If the fields aren't blank, Embedded Chat doesn't work. Use the pre-chat fields in Embedded Service setup instead.
- When you use Embedded Chat in an Experience Cloud site, you can't set pre-chat fields to fill in for logged-in users in your snippet settings file. Try the option to fill in contact fields in your component settings.
- Embedded Service setup creates quick actions specifically for pre-chat. Using these quick actions for other purposes causes pre-chat setup to fail.
- In offline support case forms, the Case Reason picklist field shows all picklist options regardless of what's specified in the record type.

Cookies and Tracking in Customer Browsers

- After users set a more restrictive third-party cookie policy in their browser, conversations work only from the web page where the chat was started. Chrome standard Incognito mode and the Microsoft Edge Balanced tracking prevention setting are also affected.
- After users set a more restrictive third-party cookie policy, such as Chrome standard Incognito mode or the Microsoft Edge Balanced tracking prevention setting, chat label translations aren't guaranteed to work as directed.
- For users blocking third-party cookies with a strict cookies policy in their browser, the chat button is missing. This applies to Firefox Enhanced Tracking set to Strict and Microsoft Edge tracking prevention set to Strict.
- Visitors can't chat across browser tabs or open new windows when using Internet Explorer 11, Microsoft Edge, or iOS devices. Chat doesn't work across new tabs and windows in Chrome Incognito.
- If a user blocks third-party cookies in Chrome standard Incognito mode, Embedded Service custom components and flow labels don't work.

Labels and Translations

- To access translation, you must allow cookies for your browser. For example, enable "Allow third-party cookies" in Safari iOS.
- Embedded Chat uses only the UTF-8 character set in Translation Workbench. Other sets can display inaccurately.

SEE ALSO:

Embedded Chat

Embedded Chat Cookies

Embedded Chat Cookies

Embedded Chat uses cookies to improve functionality and accelerate processing times. By saving a user's settings, cookies can enhance the user's experience and the performance of your Embedded Service.

EDITIONS

Available in: **Enterprise**, **Unlimited**, and **Developer** editions.

Important: The legacy chat product is scheduled for retirement on February 14, 2026, and is in maintenance mode until then. During this phase, you can continue to use chat, but we no longer recommend that you implement new chat channels. To avoid service interruptions to your customers, migrate to Messaging for In-App and Web before that date. Messaging offers many of the chat features that you love plus asynchronous conversations that can be picked back up at any time. Learn about chat retirement in Help.

Salesforce doesn't currently provide out-of-the-box functionality for end-user cookie consent management for Embedded Service. The platform is compatible with many existing third-party solutions. We recommend that you work with your internal IT teams or consult your implementation partners to identify the right solution for your organization's needs.

Embedded Service collects two different types of cookies:

- Required: Strictly necessary for the website to work correctly
- Functional: Used for analytics and performance

Note: Even though Embedded Service can run without the use of a functional cookie, site functionality can be reduced, depending on a cookie's purpose.

This table describes the Embedded Service cookies collected by Salesforce.

Table 3: Cookies for All Users

Cookie Name	Duration	Cookie Type	Description
`BrowserId`	1 year	Required	Used for security protections. Rendered on a subset of Salesforce domains, including .salesforce.com, .force.com, and the domains for Salesforce login pages, Lightning pages, and Experience Cloud sites.
`sfdc-stream`	3 hours	Required	Routes server requests within the Salesforce infrastructure for persistent sessions.
force-proxy-stream	3 hours	Required	Ensures user requests hit the same Salesforce proxy host to temporarily retrieve content from cache.
`liveagent_sid`	Session	Functional	Captures a unique pseudonymous ID for a specific browser session during chat. Embedded Service functionality isn't affected without this cookie.
liveagent_invite_rejected_[id]	Session	Functional	Remembers the rules set by an admin to hide a chat invite button after a visitor accepts or rejects it. Without

Cookie Name	Duration	Cookie Type	Description
			the cookie, the button repeatedly appears when triggered.
`language`	Session	Functional	Identifies the language for custom components, surveys, and flows, which support multiple languages. Without this cookie, translations for custom features can appear incorrectly.

SEE ALSO:

Service Cloud Chat Cookies