Oracle Fusion Service

How do I implement Embedded Service?

Oracle Fusion Service
How do I implement Embedded Service?

F91648-03

Author: DYETTER

Contents

Get Help

There are a number of ways to learn more about your product and interact with Oracle and other users.

Get Help in the Applications

Use help icons ⓘ to access help in the application. If you don't see any help icons on your page, click your user image or name in the global header and select Show Help Icons.

Get Support

You can get support at *My Oracle Support*. For accessible support, visit *Oracle Accessibility Learning and Support*.

Get Training

Increase your knowledge of Oracle Cloud by taking courses at *Oracle University*.

Join Our Community

Use *Cloud Customer Connect* to get information from industry experts at Oracle and in the partner community. You can join forums to connect with other customers, post questions, suggest *ideas* for product enhancements, and watch events.

Learn About Accessibility

For information about Oracle's commitment to accessibility, visit the *Oracle Accessibility Program*. Videos included in this guide are provided as a media alternative for text-based topics also available in this guide.

Share Your Feedback

We welcome your feedback about Oracle Applications user assistance. If you need clarification, find an error, or just want to tell us what you found helpful, we'd like to hear from you.

You can email your feedback to *oracle_fusion_applications_help_ww_grp@oracle.com*.

Thanks for helping us improve our user assistance!

1 Implementing Embedded Service

Embedded Service

What is Embedded Service?

Embedded Service is a syndicated widget that you can deploy on your Digital Customer Service application or any web page through the Engagement Engine single tag.

Embedded Service provides multi-functional help at the point of need and allows you to insert a full support experience in any page with modular components like Search Knowledge, Popular Articles, Top Actions, Create Service Request, Digital Assistant as an Agent and Live Chat with a human agent.

You can configure your Embedded Service widget using the Engagement Engine point-and-click user interface to be consistent with the branding of your site and your service process. Engagement Engine actions can trigger the Embedded Service experience based on one or more conditions regarding visitor or webpage criteria to create a proactive and personalized self-service experience.

Embedded Service natively embeds a modern web chat experience by combining assisted service by a human agent with a self-service chatbot to increase chat automation and put less strain on you human agent teams for recurring questions. The chatbot requires Oracle Digital Assistant (ODA) which is an optional add-on.

If you're licensed to use ODA, you can implement it as Digital Assistant as an Agent. This treats your intelligent chatbot as a natural extension to service with its own agent user, bot queue and routing rules. The routing logic for when a chat needs to go to a bot or a live agent is defined in the Fusion Service chat assignment rules. The Digital Assistant is treated as any other human agent and is assigned to a bot queue. The benefit of this Digital Assistant as an Agent is the centralized setup and reporting across queues with a warm hand-over to a human agent. This seamless transfer with full history in the chat transcript guarantees a full 360 view for agents inside the Fusion Service Center. Seamless transfer from the bot to a live Chat Agent can be configured in your Fusion Service deployment and Digital Assistant Skill.

What is Engagement Engine?

Oracle Engagement Engine enables you to unify rule definition and management into one scalable and responsive solution. Its point-and-click-interface allows for higher business agility to respond faster to changing customer journeys and makes these web journeys more engaging and personalized.

The primary tools of Engagement Engine are the Oracle Engagement Engine Rules Editor (Rules Editor) and the Oracle Engagement Engine Runtime. The main goal of the Rules Editor is to simplify the process of designing, testing, deploying, optimizing and measuring the impact of using rules to assist your business solution. The runtime is used when you publish your rules from the Rules Editor.

You use the application to create, edit and delete rules, and deploy rules to sites. In addition to enabling rule definition, you use the application to manage a multi-site environment. A site is a means of organizing rules with a common context. Engagement Engine enables you to create and edit an unlimited number of sites.

A multi-site environment has a number of uses:

- Use it to deploy rules to internal sites for testing and analysis before they are deployed on any public-facing environment.
- Use it to distinguish between public-facing sites depending on specified criteria. In other words, you can define different sites for different product ranges, or for different geographical locations.

If your Engagement Engine account has not been provisioned, you can request a new account by logging a service request with Oracle Support.

Required setup for Embedded Service

Here are the required tasks to set up Embedded Service.

Mandatory task list for Digital Customer Service

To use Embedded Service, you'll first need to follow the steps in the Mandatory Setup Tasks steps in *How do I enable Digital Customer Service*.

Enable profile options for Embedded Service

To use Embedded Service you must enable a number Profile Options for Fusion Service which are detailed in the following table.

First, though, you must ensure the the correct profile options for Self-Service Optimization are set.

Now configure the specific profile options to enable Embedded Service using the following table:

Profile Option	**Description**	**Default Value**
ORA_SVC_CSS_ES_ALLOWED_DOMAINS	A space-separated list of domains where you want to host Embedded Service. The default value is empty, which means Embedded Service will not load out-of-the-box. You must set this option to one or more valid domains where you will be hosting Embedded Service.	Empty
SVC_CHAT_INLAYS_ACCESS_ENABLED	Permits the Chat client to query the Chat Server configurations in anonymous mode. This profile option was originally created for chat inlay.	No
SVC_CHAT_ANONYMOUS_ACCESS_ENABLED	Enables anonymous access to chat.	No

Profile Option	Description	Default Value
SVC_CHAT_WAIT_TIME_ENABLED	Enables the display of estimated wait time for a customer before an agent is expected to accept the chat request.	Yes

1. In the Setup and Maintenance work area, click the Tasks icon, and then click the Search link.
2. In the Search field, enter **Manage Administrator Profile Values**, then click the link.
3. In the Manage Administrator Profile Values work area, do the following to allow Embedded Service to be embedded on a website:
   a. In the Profile Option Code field, enter: `ORA_SVC_CSS_ES_ALLOWED_DOMAINS` and click **Search**.
   b. In the Profile Value field, specify a space-delimited list of domains that can host Embedded Service.
   To host Embedded Service on a single domain, simply enter the domain. For example: `https://my.domain.com`
   To host Embedded Service on multiple domains, enter two or more domains separated by a single space. For example, `https://my.domain.com https://my.second-domain.com https://my.third-domain.com`
   c. For quick testing across environments, you can use "*". However, this is not recommended in the production environment as this would allow any domain to host your Embedded Service component.
4. Click **Save**.
5. Enable chat interactions by doing the following:
   a. In the Profile Option Code field, enter: `SVC_CHAT_INLAYS_ACCESS_ENABLED` and click **Search**.
   b. Set the Profile Value field to **Yes**.
   c. Click **Save**
6. Enable anonymous access by doing the following:
   a. In the Profile Option Code field, enter: `SVC_CHAT_ANONYMOUS_ACCESS_ENABLED`, and click **Search**.
   b. Set the Profile Value field to **Yes**.
   c. Click **Save**.
7. Enable the wait time setting by doing the following:
   a. In the Profile Option Code field, enter: `SVC_CHAT_WAIT_TIME_ENABLED`, and click **Search**.
   b. Set the Profile Value field to **Yes**.
   c. Click **Save**.
8. Click **Save and Close**.

Define an Embedded Service component

You can configure your Embedded Service widget using the Engagement Engine user interface to be consistent with the branding of your site and your service process.

1. Sign in to the *Engagement Engine Rules Editor*, then go to **Settings > Embedded Service > Create New Component**.
2. Enter a name for the new component, and choose a version.
   All configurable Embedded Service properties are listed below the name and version. Required properties are already added in the form.
3. To retrieve the Visual Builder Host URL follow the steps in *Retrieve the Oracle Visual Builder URLs*.

Remove the `/ic/builder` path from the end to leave only the host URL.

4. Copy and paste this value into the **Visual Builder Host URL** field.

   > **Note:** The base URL of the VB host where the Embedded Service app is deployed must not end in a forward slash. If one is present, delete it.

   All other optional Embedded Service Properties are found to the left and grouped by category.
5. Expand each category to view its properties by clicking its drop-down arrow. Add a property to the form by clicking the (**+**) icon.

   Once in the form, you can set a property value or remove the property (and restore the default) by clicking the (**x**) button.
6. Properties with a right arrow button are objects that have sub-properties.
   a. Click anywhere in the property to open a child form with the sub-properties.
   b. Note that some fields have multiple levels of nested sub-properties. In such cases, you'll only ever be editing one level at a time and your location in the hierarchy will be displayed in breadcrumbs at the top of the field. Click the left/back arrow beside the breadcrumbs to move up a level.
   c. Nested properties are also contained in a separate highlighted panel to make it clear where you're editing as you navigate between levels.

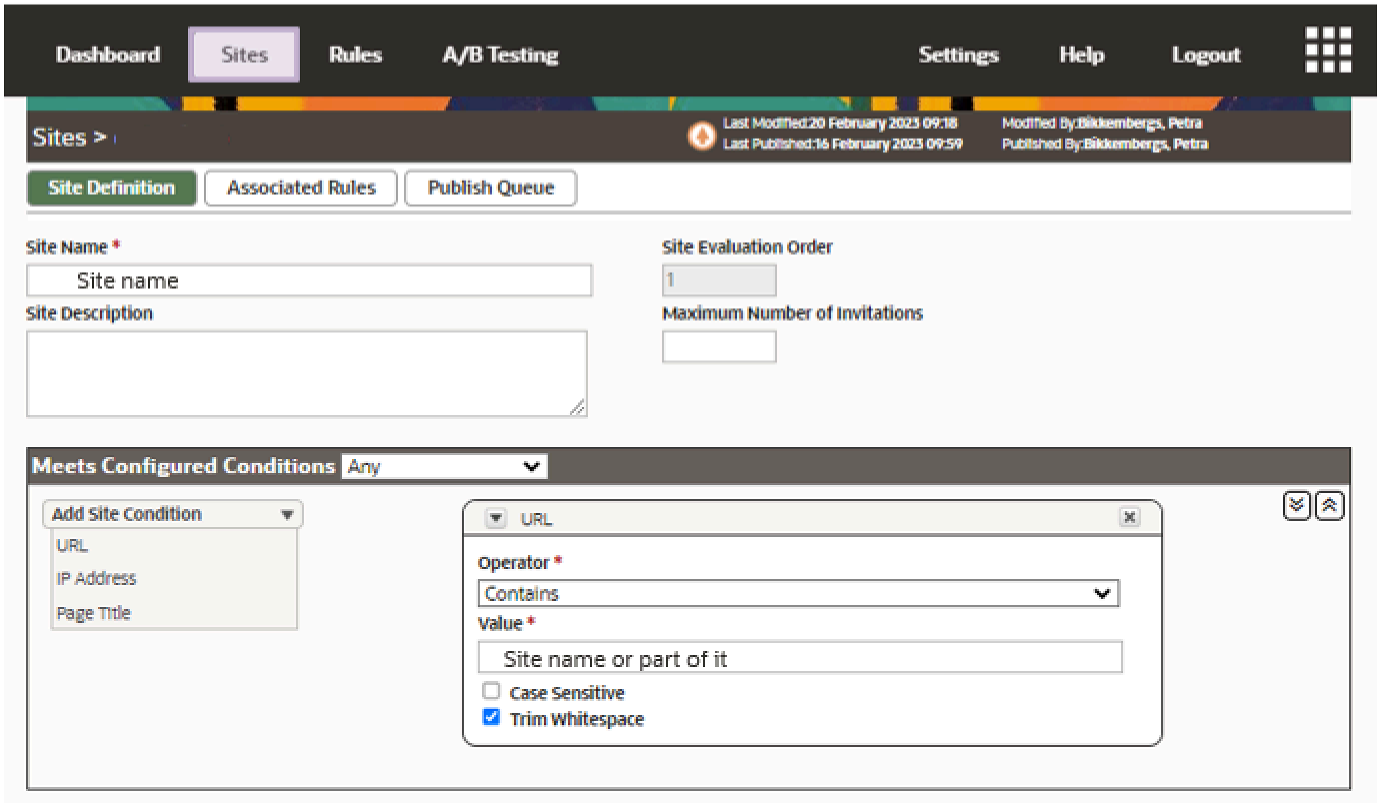
      For help with any property, click the help icon beside the field label for a description.
7. Once you are finished setting properties, click the **Save** or **Save and Publish** buttons to save your component.

Define your site

Now you define the sites where you want to enable Embedded Service.

1. Click **Sites** in the navigation bar of **Engagement Engine**.
2. Click **Create New Site**.
3. Enter a site name in the **Site Name** field.
4. Enter a site description in the **Site Description** field.

5. Click the **Add Site Condition** drop-down list in the **Meets Configured Conditions** section, and then select a condition such as the URL that contains your site name.



Last Modified:20 February 2023 09:18
Last Published:16 February 2023 09:59
Modified By:Bikkembergs, Petra
Published By:Bikkembergs, Petra

Site Name * Site name
Site Evaluation Order 1
Site Description
Maximum Number of Invitations

Meets Configured Conditions Any

URL
Operator * Contains
Value * Site name or part of it
☐ Case Sensitive
☑ Trim Whitespace

6. Save the site.

For more information, refer to *Overview of Rules* and *Site and Rule Association*.

Create and publish rules for Embedded Service

Follow this task to set up and publish rules for Embedded Service.

Use Engagement Engine rules in your Visual Builder application

Here's a code sample showing the root or shell page of your Visual Builder application, where eeid is your valid Oracle Engagement Engine Account ID:

```
<oj-dcs-ee-rules-engine eeid="<Your-EngagementEngine-AccountId>"></oj-dcs-ee-rules-engine>
```

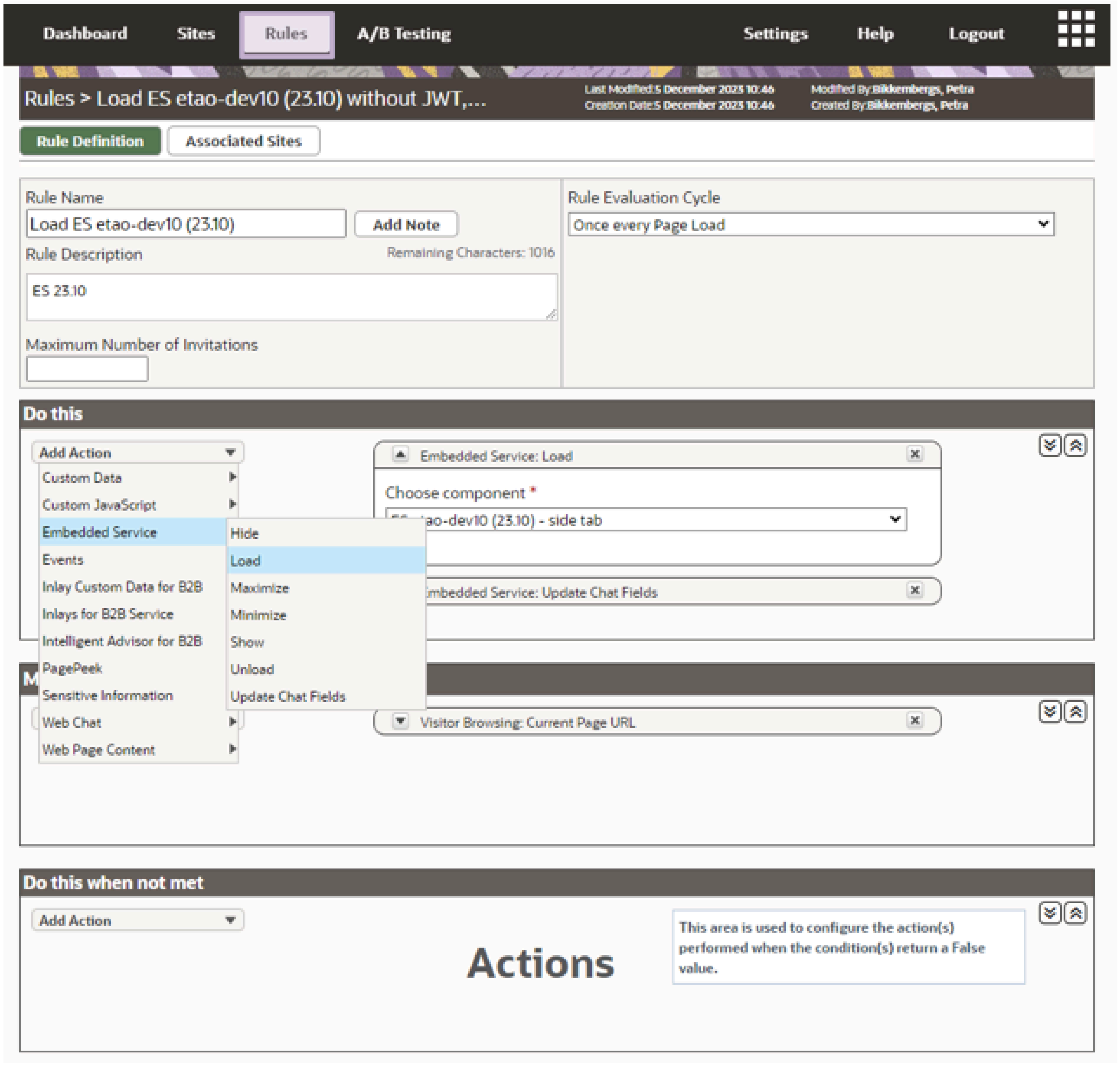
Create a load rule

1. In Engagement Engine, click **Rules** in the navigation bar.
2. Click **Create New Rules**.
3. Enter a rule name in the **Rule Name** field.
4. Enter a description in the **Rule Description** field.

5. Click the **Rule Evaluation Cycle** drop-down list and select the required evaluation cycle.
6. Click **Add Action** in the Do this area, and then select **Embedded Service > Load**.
7. Select one of the Embedded Service components from the **Choose component** drop-down list.
8. (optional) Click **Add Condition** in the Conditions area and add a condition.
9. (optional) Select the actions you'd like to perform in the **Do this when not met** section if the conditions you define are not met.

10. Save the rule, associate it with a site, and then publish it.

Here's a screenshot of the Embedded Service rule options.



Rules > Load ES etao-dev10 (23.10) without JWT,...

Last Modified:5 December 2023 10:46
Creation Date:5 December 2023 10:46
Modified By:Bikkembergs, Petra
Created By:Bikkembergs, Petra

Rule Name: Load ES etao-dev10 (23.10)

Rule Evaluation Cycle: Once every Page Load

Rule Description: ES 23.10

Remaining Characters: 1016

Maximum Number of Invitations

Embedded Service: Load

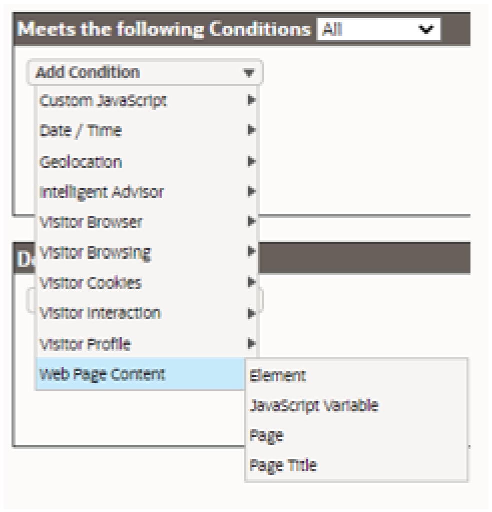
Choose component *: ES etao-dev10 (23.10) - side tab

Embedded Service: Update Chat Fields

Visitor Browsing: Current Page URL

This area is used to configure the action(s) performed when the condition(s) return a False value.

And here's a screenshot showing conditions to the rule being added.



Other Embedded Service Actions

If you would like to update Embedded Service at runtime, such as enabling it to handle a change in context or a user action on the host page, you can configure rules in Engagement Engine for other Embedded Service actions. The following actions are currently supported for Embedded Service:

- **Hide.** Hide the Embedded Service widget. Unlike with the Unload action, Embedded Service remains loaded but simply hides.
- **Load.** Load Embedded Service onto a page.
- **Maximize.** By default, Embedded Service will load as a minimized button. Invoke the Maximize action to expand the Embedded Service widget.
- **Minimize.** Minimize the Embedded Service widget and show the button.
- **Show.** Show the Embedded Service widget. Combine with the Hide action to hide and show Embedded Service based on certain page conditions.
- **Unload.** Remove Embedded Service entirely from the page. Embedded Service can only be restored again with the Load action, which takes time to again load and show the Embedded Service widget.
- **Update Chat Fields.** Change visible and, or hidden fields on the chat form.

If you anticipate showing Embedded Service again, it is recommended to use the Hide and Show actions rather than the Unload action as they are instantaneous.

For all of these actions above, if Embedded Service is already in the corresponding state, the action will have no effect.

Required Engagement Engine settings

Here are the required Engagement Engine settings.

Single Page Application

A single page application (SPA) is an application where the page loads only once on the visitor's web browser. An SPA is fully loaded on the initial page load and then page regions are updated with new page fragments loaded from the server on demand. Since the page load happens only once with an SPA, a different approach is required when using the Rules Editor with an SPA. For an SPA, the initial site evaluation and loading of rules is performed when the page loads. Subsequent site evaluations and loading of rules are performed when the hash fragment changes, or when you use the `HTML5 pushState` method.

On the **Settings** tab of the Engagement Engine rules editor, locate the **Set the Rules Editor to Work with a Single Page Application** checkbox. You must select this setting for your Digital Customer Service application.

HTTP Referrer URL

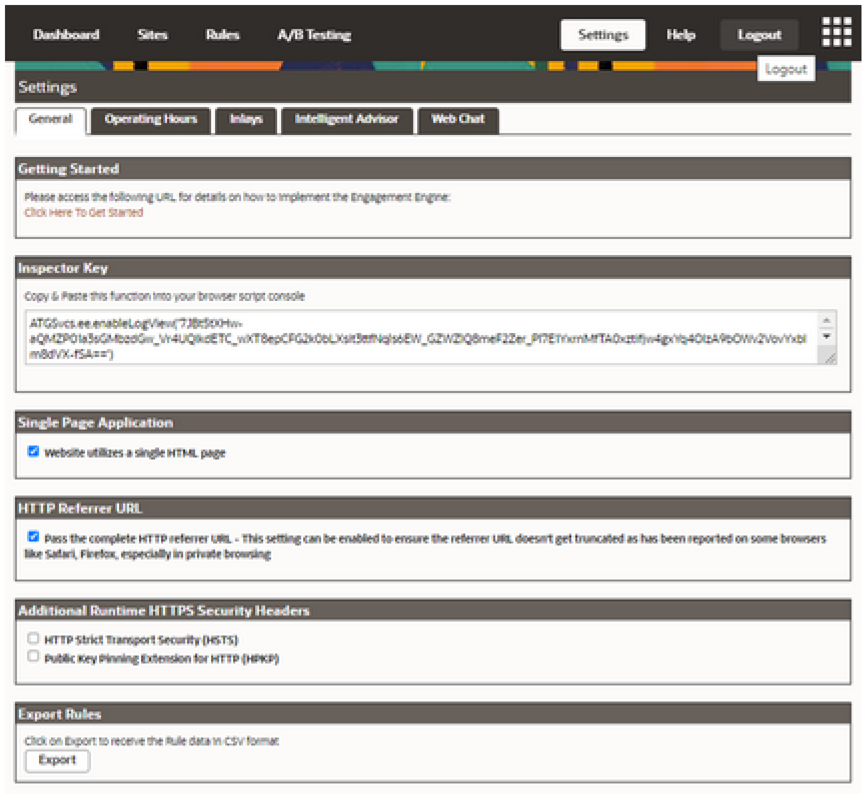
Select this setting to ensure that the complete URL is used for rule and site evaluations.

On some browsers such as Safari, Firefox, and Chrome, especially in private browsing, the HTTP Referrer URL of the page is truncated to display only the domain. This can lead to URL based rule or site conditions to be erroneously evaluated.

For instance, if the page was at `https://www.example.com/home/homepage` the referrer URL in the request sent by the browser would only be `https://www.example.com/` and thus, if the site/rule condition was to check if the URL contained the string `/home/homepage`, that condition would be wrongly evaluated to false. This behavior may also be reported for other browsers in any other conditions too, as it is dependent on the browser and its versions.

If this HTTP referrer URL setting is enabled, the referrer URL is ignored and the complete page URL is used by Engagement Engine product for rule evaluation and thus leads to the correct expected results, for both rule and site condition evaluations. The reason for introducing this checkbox was for backward compatibility as we did not want any sudden changes for existing customer's rule conditions.

The checkbox is located in the Engagement Engine rules editor under the **Settings** tab as shown in this screenshot:



Getting Started

Please access the following URL for details on how to implement the Engagement Engine:
Click Here To Get Started

Inspector Key

Copy & Paste this function into your browser script console

```
ATGSvcs.ee.enableLogView('7JBt5tXHw-aQMZP01a3sGMbzdGw_Vr4UQIkdETC_wXT8epCFG2k0bLXslt3ttfNqls6EW_GZWZlQ8meF2Zer_Pl7ETYxmMfTA0xztifjw4gxYq4OlzA9bOWv2VovYxbl m8dVX-fSA==')
```

Single Page Application

☑ Website utilizes a single HTML page

HTTP Referrer URL

☑ Pass the complete HTTP referrer URL - This setting can be enabled to ensure the referrer URL doesn't get truncated as has been reported on some browsers like Safari, Firefox, especially in private browsing

Additional Runtime HTTPS Security Headers

☐ HTTP Strict Transport Security (HSTS)
☐ Public Key Pinning Extension for HTTP (HPKP)

Export Rules

Click on Export to receive the Rule data in CSV format

For more information on Engagement Engine Rules Editor settings, refer to *Overview of Settings*.

Add Embedded Service to your website or DCS portal

Embedded Service is only usable by organizations that have tagged their pages with the Oracle Engagement Engine JavaScript tag.

Add Embedded Service to Your Website

We recommend you add the following JavaScript tag in a universal header or footer of your company's test website, so Engagement Engine is available across all pages on the test website.

Replace <customer_id> with the Engagement Engine account number that Oracle has provided.

```
<script type='text/javascript'>
window.EESvcs = { eeid:<customer_id> };
</script>
<script type='text/javascript' src="//ee.channels.ocs.oraclecloud.com/js/eesvcs.js">
</script>
```

That's it! Assuming you followed the previous sections to set up Embedded Service and you created and published the applicable Embedded Service Component, Site and Load Rule, you should now see Embedded Service loaded on the page.

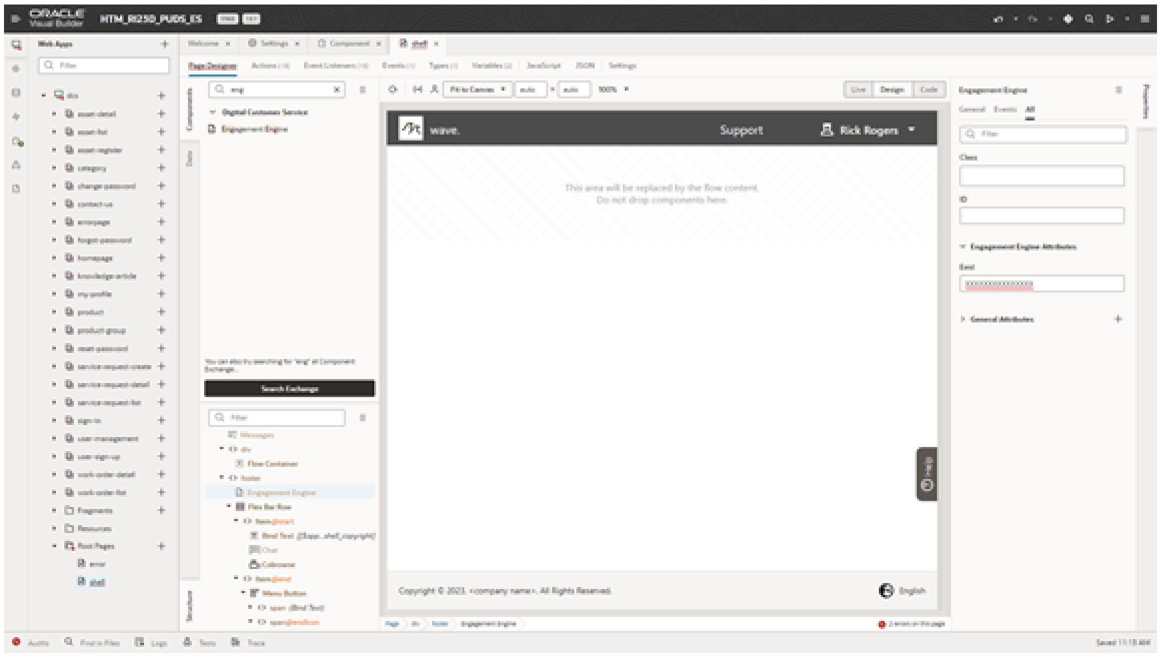
Add Embedded Service to Your DCS Portal

Here are the steps to deploy Engagement Engine on DCS and the Visual Builder app:

1. In Visual Builder, create a new Digital Customer Service Application using this *task*.
2. In your application, select the shell page under the **Root Pages** folder.
3. Now, search for Engagement Engine in the component browser.

   Note: You may be required to install this component from the Component Exchange.

4. In the Design View, drag and drop the Engagement Engine component from the Components Palette to the bottom of the Shell page.
5. With the Engagement Engine component selected on the shell page, you can add your **Engagement Engine Attributes** in the Properties editor by going to **Component settings > tab All > Engagement Engine Attributes**.



For more information on Engagement Engine, refer to *Administering Oracle Engagement Engine*.

For information on including page tags on your web page, refer to *Add Page Tags to Your Test Website*.

If you're adding the Engagement Engine component and page tags using Oracle Visual Builder, take a look at *Add the Engagement Engine Component with Visual Builder*.

Use Engagement Engine rules in your Visual Builder application

Here's a code sample showing the root or shell page of your Visual Builder application, where eeid is your valid Oracle Engagement Engine Account ID:

```
<oj-dcs-ee-rules-engine eeid="<Your-EngagementEngine-AccountId>"></oj-dcs-ee-rules-engine>
```

Configure components in Embedded Service

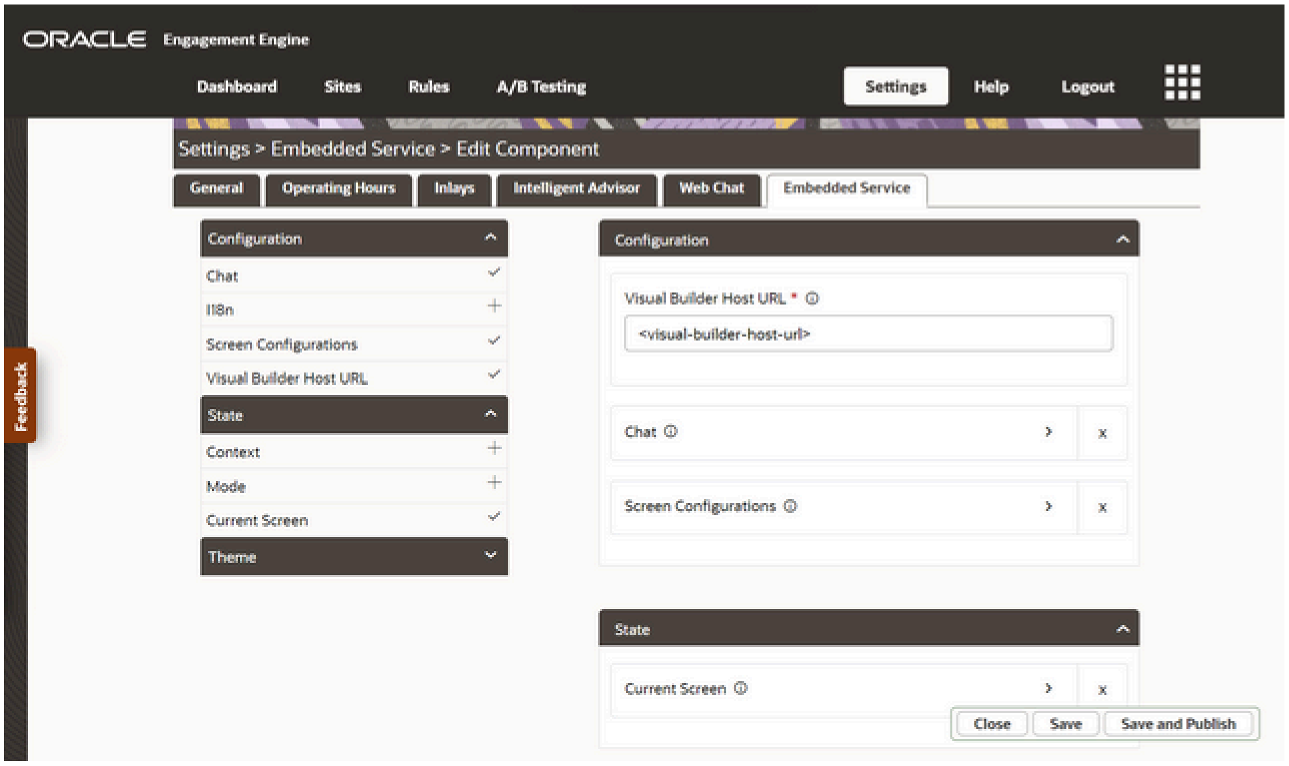
You can implement Embedded service as a modular component. This example shows how to implement the Chat component.

By default, Embedded Service is configured to have chat and knowledge displayed in a single embedded widget. While this is generally the preferred configuration when hosting Embedded Service on many websites, you may choose to offer a Chat only configuration when hosting Embedded Service on a DCS application that already offers knowledge in the portal itself.

Depending on your use case, you can use various components in your support journey such as Chat, Knowledge Search, Popular Articles, and Top Actions and Create Service Request. This task outlines how you configure the Embedded Service Chat component to support conversational interactions with a Human Agent or DA as an Agent (when used with Oracle Digital Assistant). You can choose to use Embedded Service in Chat Only mode on your web pages by configuring the widget in Engagement Engine.

You configure settings to tell Embedded Service to remain on the chat screen rather than redirecting to the home screen after disconnecting a chat. The back button is removed from the chat screen so there's no way the user can navigate anywhere else in Embedded Service.

Here's a screenshot of the settings screen:



Settings > Embedded Service > Edit Component

Configuration
Chat ✓
I18n +
Screen Configurations ✓
Visual Builder Host URL ✓
State
Context +
Mode +
Current Screen ✓
Theme

Configuration
Visual Builder Host URL *
<visual-builder-host-url>
Chat
Screen Configurations

State
Current Screen

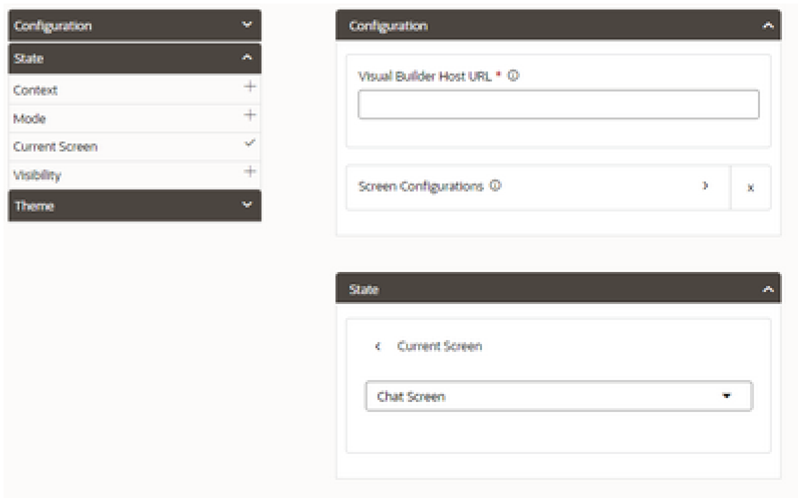
Configure Embedded Service in the Engagement Engine for Chat Only

To configure Embedded Service to only offer chat, set the following properties in the Embedded Service component in Engagement Engine:

1. Open Engagement Engine https://admin.ee.channels.ocs.oraclecloud.com/editor/.
2. Login with your credentials, then go to **Settings > Embedded Service**.
3. Open the **Category State**.
4. Select the **Current Screen** property by clicking the (**+**) plus sign.

   To view individual properties, click the (**i**) help icon.

   The Current screen of the Embedded Service widget is shown. When Embedded Service is first loaded, this is the initial screen displayed.
5. In the right column, click **Current screen property**.
6. Select the **Chat Screen** as the initial screen that will appear when the widget is loaded.



7. Open the **Configuration** category.
8. Select **Screen Configurations** property by clicking the (**+**) plus sign.

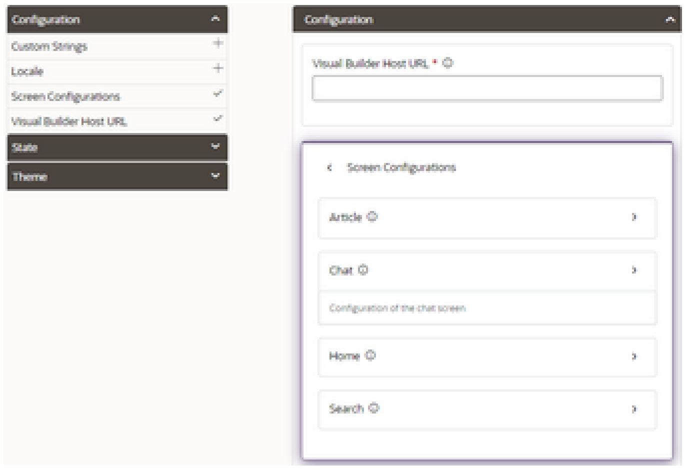
   To view individual properties, click the (**i**) help icon.

   Configure the screens in Embedded Service.
9. In the right column, click the **Screen Configurations** property.

10. In the right column, select **Chat** from the drop-down list.

    To view individual properties, click the (**i**) help icon.

    Here's a screen shot of the chat configuration screen:



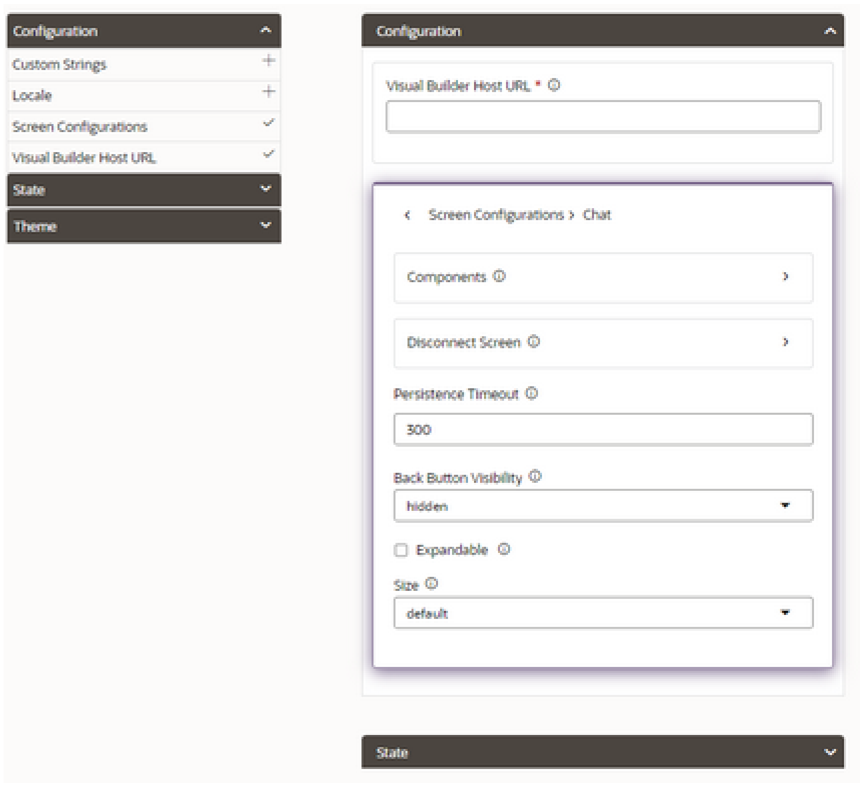
11. Click the drop-down list of the **Back Button Visibility** property.

    To view individual properties, click the (**i**) help icon.

    This screen is where you configure visibility of the back button. You'll set this property to **hidden** to prevent a user from navigating away from the screen. This can be used to offer only chat or only knowledge search or even just one article in Embedded Service without allowing users to navigate to other screens.

12. Select **hidden**.

    By default this property is set to **visible**.

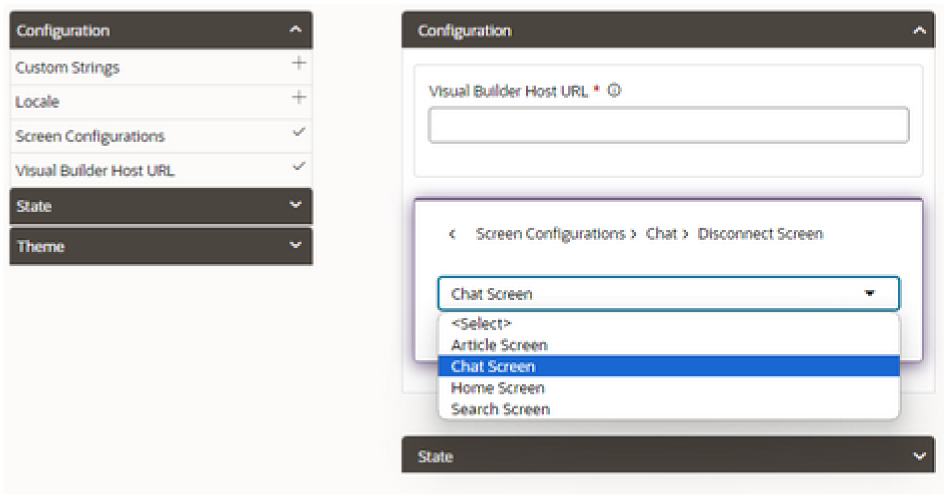


Persistence Timeout: 300
Back Button Visibility: hidden
Expandable
Size: default

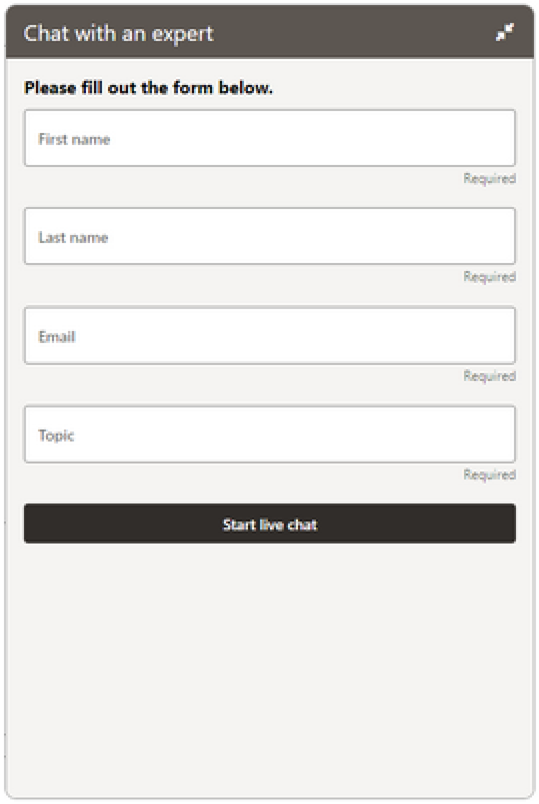
13. Click the the **Disconnect Screen** property.

    To view individual properties, click the (**i**) help icon.
14. Choose the screen to navigate to after a chat is disconnected.
15. Select **Chat Screen** from the drop-down list.

    By default this property is set to **Home Screen**.



And here's an example of the completed Chat widget:


Chat with an expert
Please fill out the form below.
First name
Required
Last name
Required
Email
Required
Topic
Required
Start live chat