



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Deployment Guide

Chat Channel

5/7/2025

Chat Channel

Contents

- **1 Chat Channel**
 - 1.1 Configuring the Backend Server and Chat Server to Support a Chat Channel
 - 1.2 Configuring Chat as the Default Channel of Engagement

When you install Genesys Web Engagement, the **Provisioning Tool** automatically configures the Backend Server and Chat Server to support a chat channel for routing chat interactions.

If you need to, you can configure this manually by completing the "Configuring the Backend Server and Chat Server to Support a Chat Channel" procedure.

You can also complete the "Configuring Chat as the Default Channel of Engagement" to specify chat as the default channel of engagement. You do this by using the **wmsg.connector.defaultEngagementChannel** option, which is only intended for development purposes and should not be used in a production environment because it turns off the pacing algorithm.

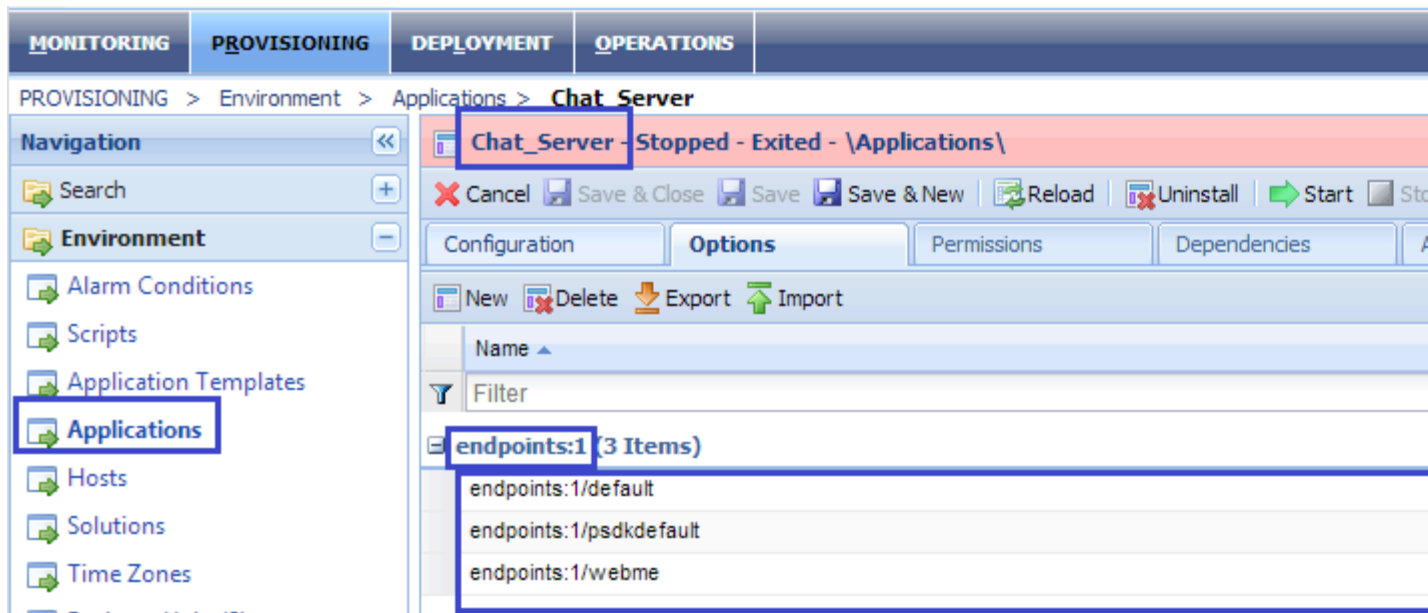
Configuring the Backend Server and Chat Server to Support a Chat Channel

Prerequisites

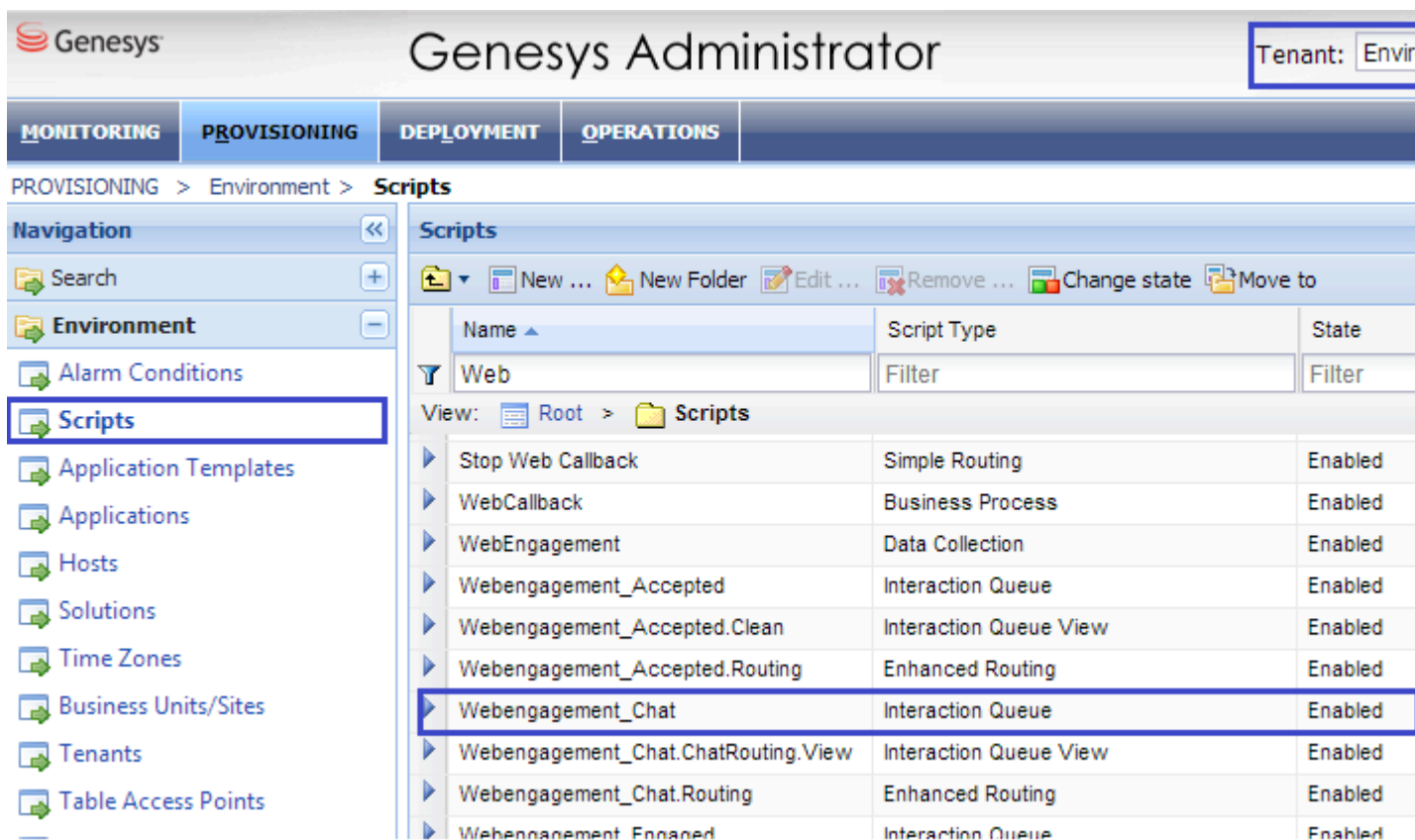
- On your Backend Server application, you have a connection to one of following:
 - Chat Server — See **Configuring the Backend Server Application** for details.
 - A cluster of Chat Servers — See **Configuring a Connection to a Cluster of Chat Servers (Optional)** for details.

Start

1. In Genesys Administrator, open your Chat Server application - either the one you connected to directly on the Backend Server, or the Chat Server on your Application Cluster (you must complete the following steps for each Chat Server application on your Application Cluster).
2. Select the **Options** tab and find the endpoints section for your tenant: **[endpoints:tenant ID]**. For example, if Chat Server works with the Environment tenant, there should be a section called **[endpoints:1]**.
3. Set the endpoint value for the **endpoints:tenant ID/webme** option to the name of the Interaction Queue where the chat interaction should be placed.
Note: Each Interaction Queue can be related to one routing strategy, either Orchestration Server or Universal Routing Server.



The **webme** option is set to Webengagement_Chat



The Webengagement_Chat Interaction Queue

4. Configure the Chat Server endpoint for the Web Engagement Backend Server application by opening

the Backend Server application and select the **Options** tab. In the `[service:wes]` section, set the value of the `wes.connector.chatServer.queueKey` option to the name of the endpoint you specified in the Chat Server application option in Step 3. The format is `tenant ID:endpoint name`.

The screenshot shows the Oracle Service Cloud Administration console. The left sidebar contains a navigation menu with categories like Navigation, Environment, Applications, Hosts, Solutions, Time Zones, Business Units/Sites, Tenants, Table Access Points, Formats, and Fields. The 'Applications' category is selected. The main pane shows the 'Web_Engagement_Backend_Server_OG' application, which is 'Stopped - Exited'. The 'Options' tab is active, displaying a list of configuration options for the 'service:wes' section. The 'wes.connector.chatServer.queueKey' option is highlighted, showing its value as '1:webme'.

Name	Section
service:wes/wes.connector.chatServer.identifyCreateContact	service:wes
service:wes/wes.connector.chatServer.queueKey	service:wes
service:wes/wes.connector.chatServer.queueWebengagement	service:wes
service:wes/wes.connector.chatServer.refreshPeriod	service:wes
service:wes/wes.connector.chatServer.requestPoolSize	service:wes
service:wes/wes.connector.chatServer.sessionRestoreTimeout	service:wes
service:wes/wes.connector.interaction.copyUserData	service:wes
service:wes/wes.connector.interactionServer.wcb.queueSubmit	service:wes

The `wes.connector.chatServer.queueKey` option is set to 1:webme

- Specify the Interaction Queue that is used as a starting point to route chat interactions. In the `[service:wes]` section, set the value of the `wes.connector.chatServer.queueWebengagement` option to the same queue you specified for the Chat Server endpoint in Step 3.

PROVISIONING > Environment > Applications > **Web_Engagement_Backend_Server_OG**

Navigation

- Search
- Environment
- Alarm Conditions
- Scripts
- Application Templates
- Applications**
- Hosts
- Solutions
- Time Zones
- Business Units/Sites
- Tenants
- Table Access Points
- Formats
- Fields
- Switching
- Routing/eServices

Web_Engagement_Backend_Server_... Stopped - Exited - \Applications\

Cancel Save & Close Save Save & New Reload Uninstall Start Stop

Configuration Options Permissions Dependencies

New Delete Export Import

Name	Section
Filter	Filter
log (11 Items)	
security (3 Items)	
service:pacing (8 Items)	
service:wes (8 Items)	
service:wes/wes.connector.chatServer.identifyCreateContact	service:wes
service:wes/wes.connector.chatServer.queueKey	service:wes
service:wes/wes.connector.chatServer.queueWebengagement	service:wes
service:wes/wes.connector.chatServer.refreshPeriod	service:wes
service:wes/wes.connector.chatServer.requestPoolSize	service:wes
service:wes/wes.connector.chatServer.sessionRestoreTimeout	service:wes
service:wes/wes.connector.interaction.copyUserData	service:wes
service:wes/wes.connector.interactionServer.wcb.queueSubmit	service:wes

The `wes.connector.chatServer.queueWebengagement` option is set to `Webengagement_Chat`

- Configure how contact management will behave when a chat session is instantiated. In the `[service:wes]` section, set the `wes.connector.chatServer.identifyCreateContact` option to one of the following values:
 - 1 — Do not identify and do not create a contact
 - 2 — Identify, but do not create a contact
 - 3 — Identify and create a contact (if absent).

The screenshot shows the Genesys Web Engagement Backend Server configuration interface. The 'PROVISIONING' tab is active, and the 'Applications' section is selected in the navigation pane. The main pane displays the configuration for 'Web_Engagement_Backend_Server_OG'. The 'Options' tab is selected, showing a list of options under the 'service:wes' section. The 'identifyCreateContact' option is highlighted, and its value is set to 2.

Name	Section
log (11 Items)	
security (3 Items)	
service:pacing (8 Items)	
service:wes (8 Items)	
service:wes/wes.connector.chatServer.identifyCreateContact	service:wes
service:wes/wes.connector.chatServer.queueKey	service:wes
service:wes/wes.connector.chatServer.queueWebengagement	service:wes
service:wes/wes.connector.chatServer.refreshPeriod	service:wes
service:wes/wes.connector.chatServer.requestPoolSize	service:wes
service:wes/wes.connector.chatServer.sessionRestoreTimeout	service:wes
service:wes/wes.connector.interaction.copyUserData	service:wes
service:wes/wes.connector.interactionServer.wcb.queueSubmit	service:wes

The `wes.connector.chatServer.identifyCreateContact` option is set to 2

The default value (3) is applied if the option is absent or specified incorrectly.

Note: Your Chat Server must have a connection to Universal Contact Server in order to control contact management through the chat session.

- Configure the chat session behavior by setting the following three options in the `[service:wes]` section:
 - `wes.connector.chatServer.refreshPeriod` — Specifies the frequency (in seconds) of chat session updates in the chat widget. The allowed range is from 1 to 5 seconds.
 - `wes.connector.chatServer.requestPoolSize` — Specifies the count of threads that serve the communication between the chat widgets and Chat Server(s)
 - `wes.connector.chatServer.chatSessionRestoreTimeout` — Specifies the timeout (in seconds) during which Genesys Web Engagement tries to restore a broken chat session if the Chat Server becomes unavailable.

The screenshot shows the Genesys Administrator interface. The left sidebar has a 'Navigation' pane with 'Applications' selected. The main pane shows the 'Options' tab for the 'Web_Engagement_Backend_Server_OG' application. The 'service:wes' section is expanded, showing a list of chat-related options.

Name	Section
service:wes/wes.connector.chatServer.identifyCreateContact	service:wes
service:wes/wes.connector.chatServer.queueKey	service:wes
service:wes/wes.connector.chatServer.queueWebengagement	service:wes
service:wes/wes.connector.chatServer.refreshPeriod	service:wes
service:wes/wes.connector.chatServer.requestPoolSize	service:wes
service:wes/wes.connector.chatServer.sessionRestoreTimeout	service:wes
service:wes/wes.connector.interaction.copyUserData	service:wes
service:wes/wes.connector.interactionServer.wcb.queueSubmit	service:wes

Chat-related options in the service:wes section

End

Configuring Chat as the Default Channel of Engagement

Specifying chat as the default channel tells the default SCXML strategy to ignore results provided by the pacing algorithm. As a result, the engagement attempt is always activated on the chat channel and the count of ready agents is ignored. If the `wmsg.connector.defaultEngagementChannel` option is not specified or specified with empty value, the pacing algorithm is used.

Start

1. In Genesys Administrator and navigate to **Provisioning > Environment > Applications**.
2. Open the application for the Web Engagement Backend Server.
3. Set the chat channel as the default channel of engagement. In the `[service:wmsg]` section, set the value

of the **wmsg.connector.defaultEngagementChannel** option to proactiveChat.

End

Next Steps

- Return to the [Genesys Web Engagement Features](#) page.