



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.

Orchestration Server Deployment Guide

Interacting with eServices

12/18/2025

Interacting with eServices

Contents

- 1 Interacting with eServices
 - 1.1 Configuring eServices Application with Type T-Server
 - 1.2 Pulling Multimedia Interactions from Interaction Queues
 - 1.3 Interaction Server Options
 - 1.4 Submitters
 - 1.5 Pulling from Multiple Interaction Servers
 - 1.6 Orchestration Default Interaction Queue
 - 1.7 Reporting on Virtual Queues

For an architecture diagram, see [eServices Interaction Flow](#).

Starting with ORS 8.1.400.27, in your Configuration Environment, the ORS Application, as a **client** of **eServices**¹ Interaction Server, must have connections to each Interaction Server Application of Interaction Server type in its **Connections** list. Genesys recommends this method for interacting with eServices/Interaction Server. In addition, this method enables ORS to **support multiple Interaction Servers**. Two **options** support this feature: switch-multi-links-enabled (mandatory) and mcr-pull-after-error-timeout (optional).

Note: Previously, the configuration for interacting with eServices/Interaction Server involved importing two templates to create two Interaction Server Application objects:

- An Interaction Server Application template for the Application object that you were using for the actual installation.
- A T-Server Application template (with its default configuration) for the second Application object.

Starting with ORS 8.1.400.27, you create the Interaction Server Application(s) using only the Interaction Server Application template. There is no need to create an Interaction Server Application using a T-Server Application Template for the second Application object. For backward compatibility, both methods of deployment are supported.

Configuring eServices Application with Type T-Server

The information below describes the Legacy method of enabling ORS to operate with **eServices** interactions by configuring two Interaction Server Application objects. The recommended method is described in [Pulling from multiple Interaction Servers](#).

1. Import two templates to create the Interaction Server Application objects.
 - Use an Interaction Server Application Template for the Application object that you are using for the actual installation.
 - Use a T-Server Application Template (with its default configuration) for the second Application object.
2. Create the Interaction Server Application objects. Ensure that both of the Interaction Server Application objects have different Application object names.
3. In the second Application object (based on the T-Server template), specify the multimedia Switch in the **Switches** tab.

Note: The first Interaction Server Application (created by using the Interaction Server Application Template) has no association in its properties with a multimedia Switch. ORS determines this association, based on the Tenant that is specified in the **General** tab of the ORS Application. Alternatively, the second Interaction Server Application (based on the T-Server template) must have a multimedia Switch configured in its properties, and it must be the same one that is used by the first Interaction Server Application.
4. On the **Server Info** tab of each Interaction Server Application, configure the same port number and ensure both Applications exist on the same Host.
5. Save the configuration.

Pulling Multimedia Interactions from Interaction Queues

Note: Composer 8.1.4 and Interaction Server 8.5.1 are required for Enhanced Pulling of multimedia interactions. Other eServices components can be 8.5.0.

Starting with 8.1.4, an ORS/Interaction Server protocol extension enhances the mechanism of getting the appropriate interactions for processing from Interaction Server to ORS. With Enhanced Pulling, the pull mechanism uses the strategy name and allows Interaction Server to set different priorities for different types of interactions (see Interaction Server Options section below). You can also use Composer's Workflow Block, Maximum Interactions property to configure the limit for each strategy. With the old pull mechanism, interactions were pulled for a specific view.

As a result of Enhanced Pulling, ORS does not need to constantly pull Interaction Server to determine if any interactions are present. This enhanced pull mechanism can result in improvements in performance and reliability for both ORS and URS.

Enhanced Pulling and Interaction Queues

As a result of Enhanced Pulling, ORS 8.1.4 supports two types of interaction queues:

1. "Enhanced" where SCXML-based applications are assigned to views instead of being assigned to interaction queues (as in previous releases) and an object called a Submitter is used for this (described below). Composer-created interaction process diagrams (associated with SCXML-based routing application) must be published into Configuration Server using Composer 8.1.4. The `Orchestration` section in the Annex of the Interaction Queue Script object is not present.
2. "Legacy" where SCXML-based applications are assigned to interaction queues and ORS 8.1.4 pulls interactions equally from all interaction queues as occurred in 8.1.3. The `Orchestration` section in the Interaction Queue Script object contains option `application` and value of that option specifies name of the Enhanced Routing Script object that handles interactions pulled from any View (filter) associated with the interaction queue. Composer-created interaction process diagrams (associated with SCXML-based applications) are published into Configuration Server using Composer 8.1.3 or earlier.

Starting with release 8.1.400.36, ORS provides the capability to define which ORS cluster will process interactions. For deployments with Enhanced Pulling of multimedia interactions, you configure the `cluster-id` option as described in Configuration Options, Interaction Submitter Script Options.

Starting with release 8.1.400.55, ORS includes additional logging for the Enhanced and Legacy pulling methods of multimedia interactions. The ORS log now prints Warning messages regarding certain pulling conditions.

Interaction Server Options

See the [eServices 8.5 Reference Manual](#) for information about required configuration options for Interaction Server.

Note: The Interaction Server pull options are in addition to the push parameters you supply in Composer's View Properties dialog box. For more information, see the Views property in Composer's [Interaction Queue block](#).

Submitters

Orchestration Server supports Submitter objects configured in Composer. In the Composer GUI, the connection between an interaction queue View and a Workflow is represented by a Submitter object. A Submitter supplies parameters that control how Interaction Server submits interactions to Orchestration Server and subsequently to Universal Routing Server. For more information, see Submitter Objects in the Composer [Interaction Queue Block](#) topic.

Pulling from Multiple Interaction Servers

Starting with Release 8.1.400.27, Orchestration Server supports the pulling of interactions from multiple [Interaction Servers](#) belonging to the same [Tenant](#) with one multimedia [Switch](#). In order to support more than one Interaction Server for the same Switch, the ORS Application level option, [switch-multi-links-enabled](#) in the orchestration section, must be set to true. This feature is supported for both [Enhanced Pulling](#) and the Legacy Pulling mechanism.

Pulling Interactions

When this feature is enabled, ORS will send pull requests to each Interaction Server on its Connections list belonging to the same Tenant. In order to limit access by any Interaction Server to some interaction Queue\View\Submitter\EnhancedRoutingScript, you must configure permissions on those objects only for the Accounts and Access Groups that were configured on the Interaction Server Application.

Important

ORS cannot pull interactions from Interaction Server if that Interaction Server has an association with more than one Tenant.

Queuing Interactions

If two Interaction Servers are configured, ORS will pull interactions from both servers and the routing strategy (workflow) will place those interactions into different queues. In this case, for a particular interaction, ORS will send RequestPlaceInQueue to same Interaction Server that it pulled that interaction from.

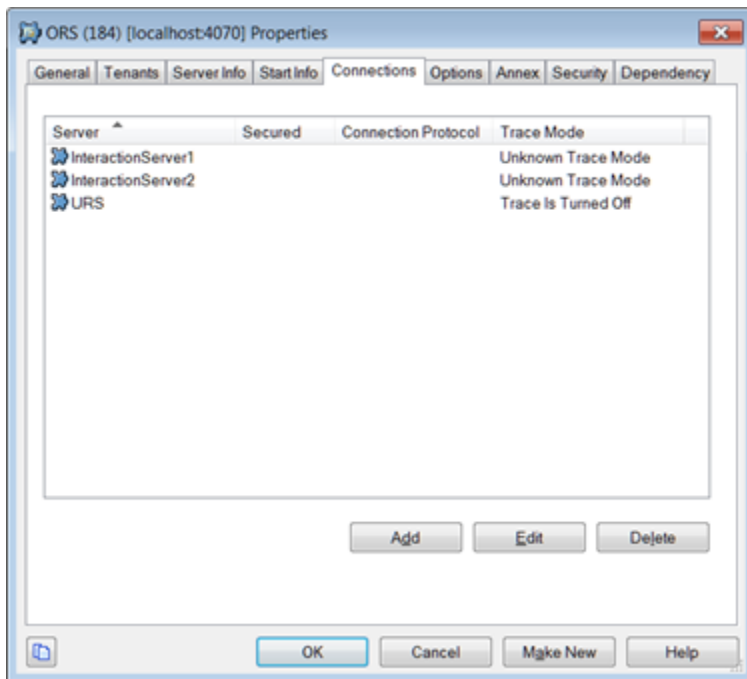
Routing Interactions

If two Interaction Servers are configured (one for e-mail, one for chat) and one set of agent targets is logged in to handle e-mail interactions, and another set of agents is logged in to handle chat interactions, ORS pulls interactions from both Interaction Servers and the routing workflow routes those interactions to the targets. In this case, for a particular interaction, ORS will route the interaction to the target on same Interaction Server from which it pulled the interaction.

Connections to Interaction Servers

ORS can be connected to Interaction Servers in one of the following ways:

- Previous to 8.1.400.27, as described in [Configuring eServices Application with Type T-Server](#), you configured an Interaction Server Application Template and a T-Server Application Template.
- Starting with 8.1.400.27, you configure ORS to directly connect to Interaction Server Application objects.



Configuring Options

Two options are associated with this feature: `switch-multi-links-enabled` and `mcr-pull-after-error-timeout`. Option `switch-multi-links-enabled` is mandatory; `mcr-pull-after-error-timeout` is optional.

- The existing Application level option, `switch-multi-links-enabled`, must be set to `true` in the ORS Application object.
- An optional Application level option, `mcr-pull-after-error-timeout`, can be set.

`mcr-pull-after-error-timeout`

Option section: `orchestration`

Default value: 120 (seconds)

Valid values: Any non-negative integer from 60 to 3600

Value changes: Immediately upon notification.

This option specifies the time interval, in seconds, when RequestPull for the same strategy or View/Queue will be re-sent to the same Interaction Server, if the previous request triggered a response with one of the specific errors listed below.

Interaction Server Error Messages

If Interaction Server receives RequestPull and some object (such as a strategy\Submitter\View\Queue) that is necessary to process the RequestPull is not visible/accessible for this Interaction Server, it will respond with specific error(s). Depending on object, Interaction Server will respond with one of the following errors:

```
__we_error_unknown_view_in_request (42)
__we_error_view_definition_invalid (83)
__we_error_strategy_does_not_exist (107)
__we_error_strategy_has_no_views (108)
__we_error_strategy_is_disabled (109)
```

If ORS receives such an error from an Interaction Server, it waits for the specified `mcr-pull-after-error-timeout` before repeating the RequestPull. ORS then sends all subsequent interaction-related requests to the same Interaction Server that this interaction was pulled from. Consult the [eServices](#) documentation for configuration information in these cases.

Orchestration Default Interaction Queue

The name of the Orchestration Server default interaction queue is `orchestration_system`. It should be created manually under each Tenant that contains an eServices deployment that the Orchestration Server works with.

Note: The Orchestration SCXML application should not be provisioned for this interaction queue, either directly (via the `orchestration` section in the Annex) or indirectly (via an Interaction Submitter > Interaction View).

Usage

ORS uses the `orchestration_system` queue as a value of the queue attribute, if that attribute is not explicitly defined in the `<ixn:createmessage>` action. Also, if queue was not explicitly defined in that action, ORS will automatically request the selected media server to send a "created" message and to clean up the interaction created by the media server for that message.

The processing of `<ixn:createmessage>` action in ORS if the queue attribute is not defined is as follows:

1. ORS will send an ESP request to create a message of the specified type to the server, defined in the `<ixn:createmessage>` action, with parameter "Queue" = "orchestration_system".
2. As soon as an ACK response is received with new InteractionID, ORS will automatically send an ESP request to send the created message to the server that is defined in the `<ixn:createmessage>` action.

Automatic clean-up of interactions, submitted into interaction queue orchestration_system, is as follows:

1. All ORS nodes constantly monitor events for interaction queue orchestration_system.
2. Upon processing of EventInteractionSubmitted, each node will determine the node that "owns" the interaction by the content of KeyValuePair "ORSI:..." in the User Data.
3. The ORS node that owns that interaction will automatically pull it from the orchestration_system queue by InteractionID.
4. As soon as the interaction is pulled, the ORS node will request Interaction Server to stop processing that interaction and the interaction will be discarded.

Reporting on Virtual Queues

Starting with 8.1.4, ORS/Universal Routing protocol allows attaching of Virtual Queue data to multimedia interactions. The attached user data is provided by Universal Routing Server and can then be used by a Reporting Solution, such as a Solution created with [Genesys Info Mart](#). The user data is attached to multimedia interactions and partial data is provided in virtual queue T-Library events.

The list of keys applicable to multimedia interactions includes: RVQID, RVQDBID, RTargetRule, RTargetAgentGroup, RTargetPlaceGroup, RVPQID, Reason, RTargetRuleSelected, RTargetTypeSelected, RTargetObjectSelected, RTargetAgentSelected, RTargetPlaceSelected, RTenant, RRequestedSkillCombination.

For example data, see the [Contact Server 8.1 API Reference](#), Query Interactions topic, Examples section.