



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

High Availability

12/20/2025

---

## Contents

- 1 High Availability
  - 1.1 Warm Standby
  - 1.2 ORS Node
  - 1.3 HA Deployment and Session Creation
  - 1.4 Data Synchronization Between Primary and Backup ORS
  - 1.5 Recovery of Failed Sessions
  - 1.6 Recovery of Voice Calls Without Persistence
  - 1.7 Limitations for ORS HA Deployment

# High Availability

Starting with version 8.1.3, Orchestration server can operate in a high-availability (HA) configuration, providing you with redundant systems.

The Framework Management Layer currently supports two types of redundant configurations: warm standby and hot standby. ORS offers the warm standby redundancy type.

## Warm Standby

Genesys uses the expression "warm standby" to describe a high-availability configuration in which a Backup-server remains initialized and ready to take over the operations of a Primary server.

HA architecture implies the existence of redundant applications. In the case of ORS, HA is achieved with Primary and Backup ORSs. These applications are configured with redundancy so if one fails, the other takes over its operations without significant loss of data or impact to business operations.

## ORS Node

A pair of Primary and Backup Orchestration Servers is called an ORS "Node". An ORS Node has a Node ID, which is a DBID of an ORS Application configured as a Primary in Configuration Server.

An ORS application running in Backup mode does not:

- Create a new session
- Serve an existing session
- Perform the pulling of multimedia interactions
- Service the HTTP port

**Note:** Starting with Release 8.1.3, ORS does not support Super Nodes and Master Super Nodes in the ORS Cluster architecture.

## HA Deployment and Session Creation

The ORS Node (Primary/Backup pair) that is responsible for a given interaction processing is the Assigned Node. Within the same cluster, an alternative Node (Primary/Backup pair), called a Reserved Node, can be another Node in single Data Center or another Node in another Data Center.

- When receiving an appropriate event on a DN loaded with a routing strategy, the Primary server/application of the assigned Node starts call processing by trying to create a session.

- A Backup application and a Reserved Node are watching the Primary application during the timeout specified in the `orchestration/call-watching-timeout` option. This continues until the session is created by Primary application (Assigned Node) or until the timeout expired.

If session for a call was successfully created, then the Primary continues the processing of the interaction.

## Session Creation Sequence

If the Primary application failed to create a session during the specified timeout or the Backup application starts running in Primary mode, the new Primary will make an attempt to create a session for a given call. If the Assigned Node (Primary/Backup pair) fail to create a session, the Reserved Node will make an attempt to create a session.

## Data Synchronization Between Primary and Backup ORS

ORS synchronizes voice treatment request data between primary and backup Orchestration Servers to ensure seamless switchovers. With this feature configured, ORS is able to send the appropriate event (`dialog.playsound.done`) to the session when treatment is ended even after ORS failover/switchover. After ORS failover/switchover, this session can be recovered and continued as normal.

## How it Works

An SCXML session initiates the application of a treatment to a voice call. ORS sends `T-Lib RequestApplyTreatment` to T-Server/SIP Server. When the treatment is applied to a voice call, the information about the associated SCXML session and the treatment applied to a particular call is stored in the primary ORS. The primary ORS then sends this information to the backup ORS. When the treatment is ended, the primary ORS sends the `dialog.playsound.done` event to the appropriate session even after failover/switchover. The persisted session is recovered and the SCXML application continues working normally.

**Note:** If an HA pair of ORS Application is properly configured except for the absence of the synch port, a warning message is logged.

## Configuring the Sync Port

1. In Genesys Administrator, configure a new synch port on the primary ORS Application object, as follows:
  - a. Open the Configuration Tab of the primary ORS Application object.
  - b. In the Server Info section, click Add in the Listening Ports field.
  - c. On the General tab of the Port Info dialog box, configure the synch port as follows:
    - i. ID - Enter synch
    - ii. Port - Enter the port number
    - iii. Connection Protocol - Enter synch  
Refer to the example in the diagram below.
    - iv. Click OK

- d. Click **Save** and **Close** to save the new port in the primary ORS.
2. Configure a new synch port on the backup ORS Application object, as follows:
  - a. Open the Configuration Tab of the backup ORS Application object.
  - b. In the Server Info section, click **Add** in the Listening Ports field.
  - c. On the General tab of the Port Info dialog box, configure the synch port as follows:
    - i. **ID** - Enter synch
    - ii. **Port** - Enter the port number
    - iii. **Connection Protocol** - Enter synch  
Refer to the example in the diagram below.
    - iv. Click **OK**.
  - d. Click **Save** and **Close** to save the new port in the backup ORS.

The screenshot shows the 'Port Info' dialog box with the 'General' tab selected. The fields are configured as follows:

- \* ID:** synch
- \* Port:** 7660
- Connection Protocol:** synch (selected from a dropdown menu)
- HA sync:** ☒ True
- Select Listening Mode:** (empty dropdown menu)
- Description:** (empty text area)

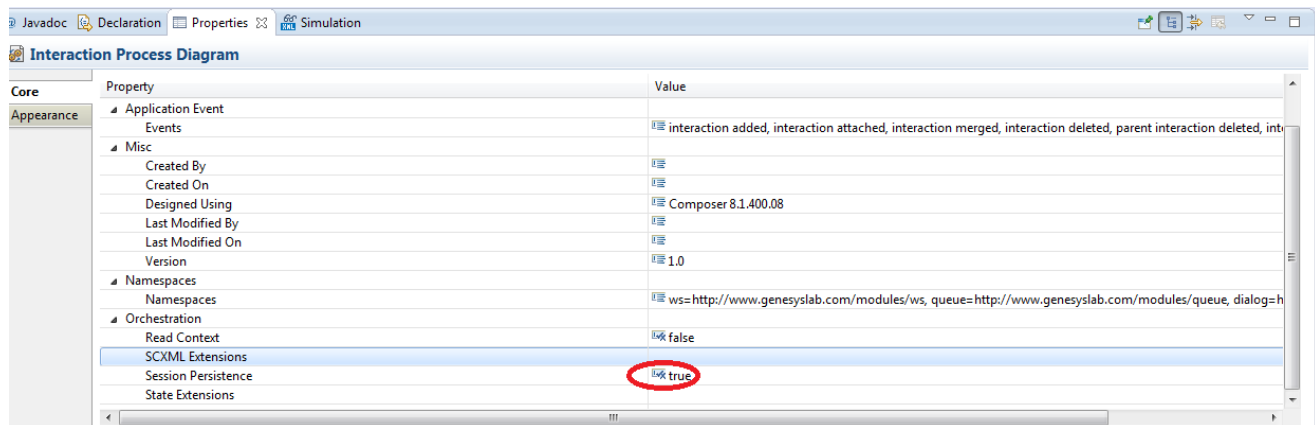
At the bottom right, there are **OK** and **Cancel** buttons.

## Configuring Persistence in ORS Application

Configure the ORS Application to work with **persistent storage**. Refer to "Configure the ORS Application to work with persistence storage" in the table of **Deployment Tasks**. Also see the **Recovery of Voice Calls Without Persistence** section below.

## Configure Persistence in SCXML Session

Configure persistence for the SCXML session. Using **Composer** in your SCXML workflow, set the `_persist` attribute into true in the `<scxml>` element by setting the Session Persistence attribute in a Core Property of the **interaction process diagram** (see the example in the figure below).



**Note:** The synchronization mechanism is controlled by two Application-level options set in the orchestration section— **backup-synch-max-buffer** and **backup-synch-max-age**. Genesys recommends that you use the default values for these options.

## Recovery of Failed Sessions

The ORS 8.1.3 and later HA architecture provides the possibility to recover a failed sessions for a given call. When the Management Layer detects failure of a Primary ORS, it starts running the Backup Orchestration server in Primary mode. The new Primary attempts to recover the persisted sessions.

**Note:** Not all persisted sessions can be recovered and some session events may be missed.

During an active session, when the Backup ORS application becomes a new Primary, the session can be explicitly restored when a proactive recovery mechanism is enabled in an SCXML application. Enabling proactive recovery for an SCXML application is controlled with the `<scxml>` tag attribute `_sendSessionRecovered`, which has values of true and false.

When set to true, ORS publishes a `session.recovered` event to a session. The SCXML application should specify its own transition to handle session recovery. See the *Orchestration Developer's Guide* for details.

**Note:** Composer 8.1.3 allows the `_sendSessionRecovered` attribute to be added to the root `<scxml>` element via the Extensions attribute of an interaction process diagram. For information on `_sendSessionRecovered`, see SCXML Elements in the [SCXML Language Reference](#). Also see [Interaction Process Diagrams](#).

## Recovery of Voice Calls Without Persistence

Starting with 8.1.400.45, ORS can be configured for voice call processing recovery upon an ORS primary/backup switchover without the Cassandra persistence/recovery feature. Requires Universal Routing Server 8.1.400.27+.

To disable Cassandra persistence:

- Set the `persistence/cassandra-nodes` option to `unknown` in all ORS Applications in the cluster (primary and backup) to disable connections to Cassandra.
- Set the `orchestration/sessionid-with-nodeid` option to `true` in all ORS Applications in the cluster (primary and backup) to enforce ORS to generate the session ID with the ORS node ID.

## Configure Primary and Backup Instances

Configure data synchronization between primary and backup instances for all ORS nodes. This includes configuring the sync port and setting the options as described above.

### [+] Responsibility of ORS Primary Instance

**The responsibility of the ORS primary instance is to:**

- Via primary/backup synchronization, notify the ORS backup instance each time a session is created to process some call, or if the session was terminated. The only data to be transmitted in that notification is `CallUUID` (Interaction ID).

#### Important

A double switchover scenario (that is, when the backup instance that takes over from a primary instance also fails, the primary instance taking over again) is not supported by design.

The restart functionality is based on the call watching mechanism in ORS. A backup ORS instance sees the same events as the primary instance. When the primary instance creates a session, the backup also creates (but not starts) its own session. As session creation requires some time, the backup instance creates the session in advance and is ready to start the session when it is required in the event of a failover.

Session restart on switchover for a backup instance which takes on the role of a primary involves a considerably big load on the system as in addition to handling new incoming calls, it should be able to restore sessions for old ones. A limited worker thread pool resource also imposes some limitations.

So, the session restart period is not the best time for doing additional processing, taking into account that the former primary may not be started or initialized yet.

But sessions for new calls that arrive after the switchover will be secured by the current backup when it is started and ready to work.

### [+] Responsibility of ORS Backup Instance

**The responsibility of the ORS backup instance is to:**

- Monitor all calls that the ORS primary instance is processing.

- Process notifications from the primary ORS instance about session terminations. If a session processing a particular call is terminated, the ORS backup instance may drop all call-related data since processing of that call is finished and there is nothing to recover for that call upon a switchover.

The ORS backup instance creates sessions simultaneously with the ORS primary instance, but starts them upon switchover to the primary role. In addition, the ORS backup instance will also:

- Publish a “session.restart” SCXML event for each session.
- Throttle the recovery process to avoid overloading of the SCXML engine.

## Configure ORS Application Options

### Important

For voice call processing recovery, all options described below must have the same value in the primary and backup ORS Applications. Synchronization between primary and backup instances must be provisioned for all ORS nodes.

### Existing Options

Set the existing ORS options as follows to enable call processing recovery for all ORS nodes in a cluster:

- Set the `new-session-on-reroute` option to `true` in all ORS Applications in the cluster (primary and backup).
- Set the `same-node-for-cons-prim-calls` option to `true` in all ORS Applications in the cluster (primary and backup).

### New Options

Configure the following options introduced in release 8.1.400.45 as required:

#### `restart-session-on-switchover`

Option section: `orchestration` in an ORS Application;  
Application in Enhanced Routing Script

Configuration Object: `Enhanced Routing Script, ORS Application`

Default value: `false`

Valid values: `true`, `false`

Value changes: Immediately, but new value will apply only to sessions created after that change.

Set to true to have the ORS backup instance attempt to restart the processing of voice calls that have been processed by the ORS primary instance and sessions that have not been terminated at the moment of switchover.

### session-restart-timeout

Option section: orchestration in an ORS Application;  
Application in Enhanced Routing Script

Configuration Object: Enhanced Routing Script, ORS Application

Default value: 50

Valid values: Any integer value starting from 10

Value changes: Immediately.

Specify the timeout (msec) between cycles of session starts upon call processing recovery. The restart quota per cycle is equal to the number of engine-working threads defined with the session-processing-threads option in the ORS scxml section (default = 8).

## New session.restart Event

A new **Asynchronous Event**, session.restart, supports this feature.

Attribute: Interactionid – string. The ID of the interaction whose processing has been restarted. This event is distributed when ORS restarts the processing of calls after an ORS switchover without using persistence. After that event, ORS will distribute all SCXML events from the beginning of call existence while the current ORS instance was running in backup mode.

## Strategy Modifications

When recovery without persistence is configured, ORS restarts call processing from the very beginning of the SCXML strategy after an ORS switchover. Depending on call flow and strategy design, it is possible that a strategy may not execute properly upon a switchover. For example, a call may have certain user data already set; some other components such as Media Control Platform may not process the same call a second time. As a result, if using session recovery without persistence, Genesys recommends modifying your SCXML strategies to treat the session.restart event using an adjusted logic to handle such scenarios.

For more information on the session.restart asynchronous event, refer to the **Session Interface** section on the *Core Extensions* topic.

## Limitations for ORS HA Deployment

- In the case of a multimedia interaction, only the Node that pulls an interaction is allowed to work with it. There are no assigned or reserved Nodes in this case.

- A session that was created on one Node cannot be transferred to another Node.