



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 Developer's Guide

Orchestration Extensions

6/10/2026

Contents

- 1 Orchestration Extensions
 - 1.1 `_genesys` Object
 - 1.2 Extension Namespaces
 - 1.3 Core Extensions (Session and Web)
 - 1.4 Queue Interface
 - 1.5 Interaction Interface
 - 1.6 Dialog (Treatment) Interface
 - 1.7 Statistic Interface
 - 1.8 Classification Interface
 - 1.9 Resource Interface
 - 1.10 Elasticconnector
 - 1.11 Agent

Orchestration Extensions

`_genesys` Object

Every SCXML session will have a global root object (`_genesys`) from which an application will have access to objects, properties, and functions to access Orchestration extensions. Each used extension is accessed as a property of `_genesys`. For example, the Interaction Interface and the Queue Interface will have the corresponding root object off of the `_genesys` object: `_genesys.ixn` and `_genesys.queue`.

Extension Namespaces

Each of the Orchestration Extensions have an associated set of namespaces for actions, events, content, and functions. The following are the namespaces for the Orchestration Extensions:

Extension	Action Namespace*	Event prefix	Object and Function root
Core Extension - Session	<i>www.genesyslab.com/modules/session</i>	session.xxx	<code>_genesys.session.xxx</code>
Core Extension - Web	<i>www.genesyslab.com/modules/ws</i>	ws.xxx	<code>_genesys.ws.xxx</code>
Queue	<i>www.genesyslab.com/modules/queue</i>	queue.xxx	<code>_genesys.queue.xxx</code>
Interaction	<i>www.genesyslab.com/modules/interaction</i>	interaction.xxx, voice.xxx, msgbased.xxx, chat.xxx	<code>_genesys.ixn.xxx</code>
Dialog	<i>www.genesyslab.com/modules/dialog</i>	dialog.xxx	<code>_genesys.dialog.xxx</code>
Statistics	<i>www.genesyslab.com/modules/statistic</i>	statistic.xxx	<code>_genesys.statistic.xxx</code>
Classification	<i>www.genesyslab.com/modules/classification</i>	classification.xxx	<code>_genesys.classification.xxx</code>
Resource	<i>www.genesyslab.com/modules/resource</i>	resource.xxx	<code>_genesys.resource.xxx</code>
Elasticconnector	<i>www.genesyslab.com/modules/elasticconnector</i>	elasticconnector.xxx	<code>_genesys.elasticconnector.xxx</code>
Agent	<i>www.genesyslab.com/modules/agent</i>	agent.xxx	<code>_genesys.agent.xxx</code>

* The namespaces used by an application must be included in the `<scxml>` element using the `xmlns`

attribute.

Core Extensions (Session and Web)

These interfaces provide functionality that is not specific to an external interface. This includes capabilities for creating sessions and accessing external http based servers.

See [Core Extensions](#).

Queue Interface

This Interface provides the ability to allow the orchestration logic to request the appropriate resource for some processing and return the appropriate address information for the resource. The current URS functionality (queuing, prioritization, and so on) will be used for this interface. In addition to core queuing and target selection, this functional module interface will also support outbound interaction processing.

See [Queue Interface](#).

Interaction Interface

An interaction represents the various types of communications between a resource and a customer:

- Conversation-based communication between a customer and given contact center or enterprise resources using a single logic media (certain media may support more than one media type over an interaction (voice and video) (for example, a voice call or a chat session). This type of interaction involves a series of information being communicated back and forth.
- Message-based communication with a customer (for example, an incoming e-mail or sms or an outgoing e-mail or sms).

See [Interaction Interface](#).

Dialog (Treatment) Interface

The dialog interface defines the functionality that allows Orchestration logic to do the following:

- Run a particular dialog application (VXML, HTML, and so on) on a specific interaction and by a specific resource.
- Collect the results of the dialog application that was run.

See [Dialog Interface](#).

Statistic Interface

This interface provides statistical information to the orchestration logic.

See [Statistic Interface](#).

Classification Interface

This interface provides the ability to classify and screen interaction content to help the orchestration logic determine what the customer wants.

See [Classification Interface](#).

Resource Interface

This functional module contains enumeration objects that can be used in other functional modules.

See [Resource Interface](#).

Elasticconnector

This extension helps remove the management of connectivity to Elasticsearch cluster from the SCXML strategy. It also provides simplified access to Elasticsearch APIs.

See [Elasticsearch Connector](#).

Agent

This extension is used for implementing agent management features like logout, DND, and ability to change agent state for voice media.

See [Agent Extension](#).