



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

Application-Level Options

12/17/2025

Contents

- 1 Application-Level Options
 - 1.1 orchestration Section
 - 1.2 persistence Section
 - 1.3 scxml Section
 - 1.4 log Section
 - 1.5 log-filter-data Section
 - 1.6 mcr Section
 - 1.7 performance-alarms Section
 - 1.8 elasticsearch Section
 - 1.9 scxml-doc-request-headers Section
 - 1.10 gts Section
 - 1.11 elasticsearch-request-headers Section

Application-Level Options

orchestration Section

This section describes orchestration section options.

backup-synch-max-age

Option section: `orchestration`

Configuration object: `ORS Application object`

Default value: `600`

Valid values: Any integer greater than `10`

Value changes: Immediately upon notification

This option specifies the maximum age (in seconds) of the data in the synchronization buffer of the primary ORS. This synchronization data is sent to the backup ORS to ensure data between the two HA servers is synchronized.

Note: Genesys recommends that you use the default values for this option.

backup-synch-max-buffer

Option section: `orchestration`

Configuration object: `ORS Application object`

Default value: `100`

Valid values: Any integer greater than `100`

Value changes: Immediately upon notification

This option specifies the maximum size (in KB) of the synchronization buffer in the primary ORS. This buffer is used to hold synchronization data before it is sent to the backup ORS to ensure data between the two HA servers is synchronized.

Note: Genesys recommends that you use the default values for this option.

call-watching-timeout

Option section: `orchestration`

Configuration object: `ORS Application object`

Default value: 5000

Valid values: 1000 - 300000

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the maximum amount of time in milliseconds that a reserved ORS node waits for the assigned node to start call processing. If the assigned node fails to create a session during this timeout, the reserved node attempts to create a session.

For example: `call-watching-timeout=5000`

`complete-assoc-on-target-node`

Option section: `orchestration`

Configuration object: ORS Application object

Default value: `false`

Valid values: `true`, `false`. This option must have the same value for all ORS nodes in a cluster.

Value changes: Changes to this option are applied dynamically.

Introduced in 8.1.400.53, this option improves how ORS performs a re-association of an interaction with a remote target session started on different ORS node within a cluster. It defines the way ORS handles the `interaction.added` and `interaction.present` events when associating an interaction with a remote target session running on an ORS node other than the node executing the `<ixn:associate>` action. Previously, in a multimedia deployment, ORS may wrongly distribute the `interaction.present` event to the remote target session before the interaction was pulled.

When this option is set to `true`, the remote ORS generates the `interaction.added` and `interaction.present` events after the interaction is associated with the target session. This is the proper mode for handling interactions associated with a remote target session. Genesys recommends it for new deployments, for clusters where ORS nodes can be upgraded simultaneously, or for clusters where all ORS nodes are already upgraded to version 8.1.400.53 or higher.

When this option is set to `false`:

- For voice interactions, ORS sends both the `interaction.added` and `interaction.present` events to the remote target session.
- For multimedia interactions, ORS sends the `interaction.added` event to the remote target session. The remote ORS node generates an `interaction.present` event.

Use this mode in deployments where ORS nodes in the cluster are upgraded gradually. Set the option value to `true` as soon as the upgrade of all nodes is completed.

Please note that if you want the remote target session to be able to work effectively with a re-associated interaction, be sure to explicitly place this interaction in any interaction queue that ORS works with. ORS does not automatically do this placement because it does not know strategy developer intentions and subsequent actions.

def-stat-object-type

Object: ORS Application

Option section: `orchestration`

Default value: `agent`.

Valid values:

Option Value	Description
<code>agent</code>	Agent
<code>agent_place</code>	Agent Place
<code>agent_group</code>	Group of Agents
<code>place_group</code>	Group of Places
<code>queue</code>	Queue
<code>route_point</code>	Routing Point

Value changes: Take effect immediately

This option, introduced in Release 8.1.400.17, can be used for [Direct Statistic Subscription](#). This value will be used by default in case an object type is skipped in the object description in the strategy.

def-statserver-name

Object: Name of an object specified below in Configuration Layer as described below.

Location in Configuration Layer by precedence: `Routing Point`, `T-Server`, `Tenant`, `ORS Application`

Valid value: The name of any available Stat Server

Default value: The first available Stat Server that has the “current” Tenant in its Tenants list

Value changes: In 8.1.400.17, value changes take effect immediately except at Tenant level, when value changes take effect upon restart. In 8.1.400.18, value changes at all levels (including Tenant) take effect immediately.

ORS will look up the `def-statserver-name` option in following order:

1. `Routing Point`, current for interaction/session (section `orchestration`)
2. `T-Server` (section `orchestration`)
3. `Tenant` (section `orchestration`)
4. `ORS` (section `orchestration`)

This option, introduced in Release 8.1.400.17, can be used for [Direct Statistic Subscription](#).

external-url

Option section: `orchestration`

Configuration object: `ORS Application object`

Default value: `None`

Valid values: Any valid URL

Value changes: For the next interaction

Use to redirect HTTP requests in scenarios when an HTTP request about a specific session is delivered to an ORS node that is not handling the session. This URL will be populated into the Location header of a 307 Temporary Redirect response as a redirect target.

Configure this option on the Primary ORS Application object. Set its value to the URL of Load Balancer located in front of the ORS node. Another ORS node will use this option and populate its value into the Location header of a 307 Temporary Redirect response as a redirect target. It allows delivery of the HTTP request to the session owner.

filter-eval-expr

Option section: `orchestration`

Configuration object: `ORS Application object`

Default value: `false`

Valid values: `true` or `false`

Value changes: Immediately upon notification

ORS 8.1.400.31 introduced option, `filter-eval-expr`, which works with `METRIC:eval_expr` and Debug Logging Segmentation (see option `log-trace-segments`). This option enables hiding of sensitive data in expression and result fields of the `eval_expr` log metric. If set to `true`, ORS verifies that string data against the regular expression defined in the `ors-regex-<name>` option.

functions-by-urs

Option section: `orchestration`

Configuration object: `ORS Application object`

Default value: `true`

Valid values: `true` or `false`

Value changes: Immediately (starting with ORS Release 8.1.300.39)

This option was introduced in ORS Release 8.1.300.36.

Orchestration Server allows you to retrieve information directly from the Configuration Layer without sending requests to Universal Routing Server when using the following `_genesys.session` functions: `isSpecialDay`, `timeInZone`, `dateInZone`, `dayInZone`, `listLookupValue`, and `getListItemValue`.

To enable this functionality, use the option `functions-by-urs`.

If set to `true`, Orchestration Server sends requests to Universal Routing Server.

If set to `false`, Orchestration Server retrieves information directly from Configuration Layer.

Note: The `get-list-item-by-urs` option, which covers only one function, is obsolete and should not be used. It is replaced by the `functions-by-urs` option, which covers six functions.

`getlistitem-binary-conversion`

This option, introduced in Release 8.1.400.15, defines how Transaction List item attributes stored in binary format will be treated by function `_genesys.session.getListItemValue(list, item)` when item is returned as an OBJECT of key/value pairs.

Option section: `orchestration`

Default value: `ignore`

Valid values: `ignore`, `string`, `array`

Value changes: Immediately upon notification.

When set to `ignore`, no value is returned (as if it does not exist in the List).

When set to `string`, the value is converted to string. If there are bytes with a zero value, then the string will contain bytes only until the first zero byte.

When set to `array`, each byte is converted to element of array. Previously, binary attributes were converted to array.

Also, if Transaction List item attributes are stored in non-string format and the `_genesys.session.getListItemValue(list, item, key)` function retrieves the value of a particular key, it is returned as string. Previously, in this scenario, the `_genesys.session.getListItemValue()` function ignored non-string attributes.

`get-list-item-by-urs`

This option, which covers only one function, is obsolete and should not be used. It is replaced by the `functions-by-urs` option, which covers six functions.

`heartbeat-backup-status`

Option section: `orchestration`

Configuration object: `ORS Application object`

Default value: `503`

Valid values: Valid HTTP response codes

Value changes: Immediately.

This option, introduced in ORS Release 8.1.300.32, allows you to define a valid HTTP response code sent by an ORS instance running in Backup mode to the requestor, after receiving a heartbeat query.

This option can be used to provide HTTP health monitoring support to determine whether the ORS instance is operating in Primary or Backup mode.

When an ORS instance running in Primary mode receives a specific GET request for `/heartbeat` URI in the format `http://<ORS_host>:<ORS_HTTP_port>/heartbeat`, it responds with `200 OK`.

If an ORS instance is running in Backup mode, it responds with `503 Service Unavailable` or with the HTTP response code defined in the new configuration option `heartbeat-backup-status`.

Genesys recommends configuring health monitoring in deployments with Load Balancing.

`http-pending-max-time`

Option section: `orchestration`

Configuration object: `ORS Application object`

Default value: `600`

Valid values: Any non-negative integer between 1 and 600

Value changes: Takes effect after restart.

Orchestration Server supports pipelining of HTTP requests according to RFC 2616, Section 8.1.2.2. As a result, when ORS receives several HTTP messages over the same connections, the responses are in the same order. This option and `http-orphan-session-action` implement this functionality.

This option, introduced in Release 8.1.300.43, defines the maximum time, in seconds, that Orchestration Server will process an HTTP request before sending a timeout response (HTTP message with status code 408) to the client.

`http-orphan-session-action`

Option section: `orchestration`

Configuration object: `ORS Application object`

Default value: `none`

Valid values: `none`, `terminate`, `<valid scxml event name>`

Value changes: Immediately upon notification.

This option, introduced in ORS Release 8.1.300.43, defines the Orchestration Server action if a response to an HTTP request to create an SCXML session cannot be sent to the client or when a timeout message is already sent. Valid values:

- `none`: No action will be taken.
- `terminate`: The SCXML session will be terminated.
- `<valid scxml event name>`: The event will be sent to the SCXML session.

See `http-pending-max-time` option for additional information on this option.

ixnfm-idle-session-ttl

Option section: `orchestration` (in Application) or Application (in Enhanced Routing Script)

Configuration Object: Enhanced Routing Script, ORS Application

Default value: 0

Valid values: Any non-negative integer between 0 and 3600.

Value changes: Immediately upon notification.

Defines the maximum time (in seconds) that an SCXML session may exist only if interactions have been successfully redirected and there are no other interactions subsequently attached to that session. If configured in Enhanced Routing Script, the value will override value from the ORS Application for sessions created from that Script. Value 0 disables this feature.

Use to eliminate the possibility of the scenario where an SCXML session gets "stuck" in memory caused by an unsuccessful detach action execution and/or incorrect strategy design. This scenario can prevent the creation of new sessions if the number of sessions reaches the limit of concurrent active sessions per ORS Application.

log-trace-segments

Option section: `orchestration`

Default value: `all`

Valid values: Any combination of these comma-separated segment names (case-sensitive):

`OrsInternal, OrsInit, OrsSynch, Cluster, ItxLink, ItxManager, CallMonitor, ORSURS, Persistence, Schedule, SessionManager, ThreadSync, ScxmlCommon, ScxmlSession, ScxmlMetric, ScxmlQOS, ORSScxml, ScxmlConfig, ScxmlIxentialAction, ScxmlOrsAction, ScxmlDFM, ScxmlEvent, ScxmlClassification, ScxmlFMDialog, ScxmlFMInteraction, ScxmlFMQueue, ScxmlFMResource, ScxmlFMSession, ScxmlFMStat, ScxmlFMWeb, ScxmlProperty, ScxmlIO, SelfMonitoring, ScxmlMetricEvalExpr, ElasticConnector, ScxmlCacheStats`

Value changes: Immediately upon notification.

Use this option for Debug Log Segmentation.

Debug Log Segmentation

Starting with Release 8.1.300.52, ORS supports Debug Log Segmentation, which provides more precise control of the logging when the `log-trace-segments` option is configured. The debug segment header `{<segment_name>:<log_level>}` is now added right after the timestamp. For example: `10:31:02.249 {ScxmlMetric:3} METRIC <event_queued sid='ML06K6519D5UPAGSHNE41VJBSS00000M' name='interaction.added' type='external' thread='15800' />`

The `log-trace-segments` option, using a comma separated list, specifies the types of information that are written to the Debug log files. Setting this option logs all messages that fall into the specified

segment or list of segments. This option works with the **x-server-trace-level** option, which must be set to 1, 2 or 3.

Background: Orchestration Server produces a large number of Debug level log messages. The sheer volume of log messages can make it difficult to isolate the specific messages needed to troubleshoot different types of deployments. The ORS Debug Log Segmentation feature groups log messages by functional segments, such as common module, API, or feature group. Segment names are added at the beginning of each log message.

- Use the **!** character before a segment to exclude messages falling under this segment.
- The **all** keyword lists all segments (used as regular segment). All debug messages are printed with the log level defined by the **log/x-server-trace-level** option.

To control the log levels more selectively for each segment, use this syntax:
<segment_name>:<log_level>.

- If no log level is specified, the default value (**x-server-trace-level**) is applied.
- The log level specified with the excluded segment (with **!** prefix) will be ignored.

In the case of a syntax error, the default value will be applied.

Important

If an unknown segment is specified, ORS ignores it without generating an error.

Segment Descriptions

N	Segment Name	Functionality
1.	OrsInternal	Processing of ORS internal objects and events
2.	OrsInit	ORS instance initialization
3.	OrsSynch	Synchronization between primary and backup instances
4.	ScxmlQOS	Message for SCXML engine load/session creation time upon each session start
5.	Cluster	Messages related to ORS node initialization within ORS cluster (cluster assignment, cluster configuration, etc.)
6.	AgentFM	Reserved
7.	ItxManager	Low-level messages for eServices communication
8.	ItxLink	Low-level messages for eServices communication
9.	CallMonitor	Call/Interaction processing events/requests
10.	AgentMonitor	Reserved
11.	ORSURS	Communication with URS
12.	Persistence	Operations with Cassandra persistent storage
13.	Schedule	Scheduling via persistence layer
14.	SessionManager	Session management-related messages
15.	ThreadSync	Internal inter-thread communication
16.	ScxmlCommon	Common operations within SCXML engine (such as HTTP doc requests preparation and execution, caching, and so on)
17.	ScxmlSession	Session-related non-metric SCXML messages
18.	ScxmlMetric	SCXML METRIC messages
19.	ORSScxml	ORS core messages about SCXML sessions created/terminated
20.	ScxmlConfig	Configuration options used by SCXML engine
21.	ScxmlIxnAction	Messages related to SCXML actions execution
22.	ScxmlOrsAction	Messages related to internal processing of SCXML action data

(validation, compilation)	
23.ScxmlDFM	Messages from Distributed Function Module
24.ScxmlEvent	Messages about low-level SCXML event generation/processing
25.ScxmlClassification	Messages from Classification Function Module
26.ScxmlFMDialog	Messages from Dialog Function Module
27.ScxmlFMInteraction	Messages from Interaction Function Module
28.ScxmlFMQueue	Messages from Queue Function Module
29.ScxmlFMResource	Messages from Resource Function Module
30.ScxmlFMSession	Messages from Session Function Module
31.ScxmlFMStat	Messages from Statistics Function Module
32.ScxmlFMWeb	Messages from Web Function Module
33.ScxmlProperty	Messages about low-level SCXML object properties processing
34.ScxmlIO	Messages from SCXML engine internal fetching module (low-level HTTP
doc operations)	
35.SelfMonitoring	Messages related to Performance Monitoring
36.ScxmlMetricEvalExpr	eval_expr log metrics
37.ElasticConnector	Messages from ElasticConnector Function Module
38.ScxmlCacheStats	SCXML internal cache statistics messages

Examples

- `all,!0rsInternal,!0rsURS` - All segments enabled except `0rsInternal` and `0rsURS`.
- `ScxmlIO,ThreadSync` - Enable `ScxmlIO` and `ThreadSync` only.
- `all,0rsURS` - All segments enabled (the same as `all`).
- `!0rsURS` - All segments disabled (the same as empty)
- `CallMonitor,!all` - All segments disabled (the same as empty).
- `all,0rsURS:1` - All segments enabled with default trace level, but trace level for `0rsURS` segment will be changed to 1.
- `""` - All segments disabled (the same as `!all`, or `x-server-trace-level=0`).
- `all,!0rsURS:3` - All segments enabled except `0rsURS`.
- `all:3` - The same as `all` with `x-server-trace-level=3`.

Debug Log Segment Header Updates

Starting with ORS 8.1.400.31, a new Debug Log Segment header `ScxmlMetricEvalExpr` is added for `METRIC:eval_expr`. In a log file, it will appear as follows:

```
14:47:11.414 [T:8436] {ScxmlMetricEvalExpr:3} METRIC <eval_expr sid='ors2521'
expression='global._data =
new Object();' result='[object Object]' thread='8436' />
```

As a result, the `log-trace-segments` option adds `ScxmlMetricEvalExpr` as a new value. Also see `filter-eval-expr`.

Starting with ORS 8.1.400.40, in support of the [Elasticsearch Connector](#) feature, a new Debug Log Segment header, `ElasticConnector` is added. Log file example:

```
14:30:10.259 {ElasticConnector:1} Response from Elasticsearch host:port:
```

```
max-session-create-time
```

```
Option section: orchestration
```

Configuration object: ORS Application object

Default value: 3600

Valid values: Any integer between 5 and 3600

Value changes: Immediately

This option, introduced in ORS Release 8.1.300.45, defines the maximum session creation time in seconds. If session creation will take longer, ORS will enter an overload mode and stop creating sessions until all pending session creation requests are completed.

map-composer-log-levels

Option section: orchestration

Configuration object: ORS Application object

Default value: false

Valid values: true or false

Value changes: Immediately

This option, introduced in ORS Release 8.1.300.29, allows mapping of log levels defined in the <log> action element into the specific Genesys log events. If enabled (set to true), Orchestration Server will generate log events of the Alarm, Standard, Interaction, Trace or Debug levels. The log levels defined in the level attribute of the <log> element are mapped as follows:

- Info (log level 1) is mapped into the Trace log event 23023
- Error (log level 2) is mapped into the Standard log event 23011
- Warning (log level 3) is mapped into the Interaction log event 23022
- Debug (log level 4) is mapped into the Debug log event 23026
- Alarm (log level 5) is mapped into the Alarm log event 23012

Setting the scxml-log-filter-level option value to 1, 2, 3, 4 or 5 will override the logging results of the map-composer-log-levels option

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 is associated with **pulling from multiple Interaction Servers**. It can work with switch-multi-links-enabled. Option switch-multi-links-enabled is mandatory when pulling from multiple Interaction Servers; mcr-pull-after-error-timeout is optional. The mcr-pull-after-

error-timeout option specifies the time interval, in seconds, when the 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 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 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.

mcr-pull-by-this-node

Option section: `orchestration`

Configuration object: `ORS Application object`

Default value: `false`

Valid values: `true` or `false`

Value changes: in setting the next timer interval

This option specifies that the pulling of eServices interactions will be allowed to be performed by this node if set to `true`.

Note that by default this option is set to `false`, so if the default is left on all nodes, no pulling will occur.

Note that it is allowed to have more than one node to pull interactions.

For example: `orchestration/ mcr-pull-by-this-node = true`

mcr-pull-cycle-quota

Option section: `orchestration`

Configuration object: `ORS Application object`

Default value: `100`

Valid values: Any non-negative integer less than 50000

Value changes: Immediately upon notification

This option, introduced in 8.1.400.11, allows you to define how many multimedia interactions ORS will request to be pulled from queues in a given pulling cycle during enhanced pulling. This number is divided equally and rounded up to the nearest integer among the routing strategies that have no outstanding pull request. If there is an outstanding pull request for a strategy, then no pull request will be sent and this strategy will not be used in the calculation of the per-strategy quota for this particular pulling cycle.

mcr-pull-interval

Option section: orchestration

Configuration object: ORS Application object

Default value: 1000

Valid values: Any positive integer.

Value changes: in setting the next timer interval

This option provides the number of milliseconds between attempts to pull interactions from the queues/views managed by the ORS.

For example: orchestration/mcr-pull-interval = 5000

mcr-pull-limit

Option section: orchestration

Configuration object: ORS Application object

Default value: 5000

Valid values: Any positive integer from 0 to 50000

Value changes: in setting the next timer interval

This option provides the maximum number of pulled interactions that each node in a cluster (or ORS working in standalone mode) is allowed to have at any given time. A value of 0 disables the pulling of multimedia interactions completely for this instance of ORS. This option is not applicable to the **enhanced pulling of multimedia interactions** feature. The **eServices** options control the number of pulled interactions for enhanced pulling.

mcr-queue-on-fails

Option section: orchestration

Configuration object: ORS Application object

Default value: An empty string

Valid values: Name of a valid interaction queue or an empty string

Value changes: Immediately

This option specifies the interaction queue into which an interaction is placed if session creation for the multimedia interaction fails, instead of returning it to the same interaction queue. The feature is enabled if the option value is not an empty string.

new-session-on-reroute

Option section: orchestration

Configuration object: ORS Application object

Default value: false

Valid values: true or false

Value changes: Immediately

This option, introduced in ORS Release 8.1.300.39, controls Orchestration Server behavior when T-Server performs default routing.

In the scenario of T-Server performing default routing, T-Server redirects a call that is processed by some strategy on some Routing Point. The call is then diverted and queued on another Routing Point with call state 22.

The option new-session-on-reroute allows you to explicitly specify how ORS should process the call. It also enables ORS and Universal Routing Server (URS) to handle this scenario in a consistent fashion.

- If new-session-on-reroute is set to true, ORS will terminate the existing live session or abort the session recovery process (whatever takes place) when it receives events related to call redirection.

Upon being diverted from the Routing Point (only) with call state 22 (only), ORS breaks the existing call/session association (if any). As a result, the call becomes free to create a new session upon arriving at the default destination DN loaded with the strategy.

- If new-session-on-reroute is set to false, ORS reverts to its previous behavior where it does not break the existing call/session association when it receives events related to call redirection. As for live sessions, their execution will be continued, and they will receive the usual events indicating that the initial Routing Point party disappeared. As a result, the live session can handle the situation when the call being handled is moved to another DN by T-Server in the middle of the routing process.

It is necessary to keep the both ORS option new-session-on-reroute and Universal Routing Server option use_ivr_info synchronized (both are either true or false).

parse-start-params

Option section: orchestration

Configuration object: ORS Application object

Default value: false

Valid values: true or false

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies whether to enable or disable serialization/deserialization of session parameters. When true, parameters delivered to sessions started using `invoke` or `session:start` will retain their original type. When false, the session parameters will be converted to string.

For example: `parse-start-params = true`

`same-node-for-cons-prim-calls`

Option section: `orchestration`

Configuration object: ORS Application object

Default value: `false`

Valid values: `true` or `false`

Value changes: Changes to this option will take effect immediately, but only for consult calls created after that change.

For compatibility with ORS 8.1.3, Release 8.1.400.33 introduced the `same-node-for-cons-prim-calls` option. Its purpose is to control call distribution compatibility between ORS nodes when a deployment contains nodes from both 8.1.3 and 8.1.4 releases.

When set to `false`, ORS 8.1.4 and 8.1.3 call distribution works the same: ORS determines the ownership of consult calls in the same way as it determines the ownership of any other call. The only exceptions are ORS versions 8.1.400.30 through 8.1.400.32, where compatibility with 8.1.3 was broken. When set to `true`, consult calls will be handled by the same node that handled the related primary call.

`scxml-log-filter-level`

Option section: `orchestration`

Configuration object: ORS Application object

Default value: empty (disabled)

Valid values: any valid combination of custom-defined and Genesys log levels where

Custom log level: An Integer composed of the digits 1, 2, 3, 4, or 5 and less than or equal to 9 digits in length (minimum of 1 and maximum of 555555555).

Genesys log levels: `STD`, `INT`, `TRC`, `ALR` and `DEBUG`. These are not case-sensitive.

Value changes: Immediately

This option was introduced in ORS Release 8.1.300.29. If this option is configured and the level parameter of the `__Log` function or level attribute of the `<log>` action element in the SCXML application matches the custom log level value, ORS generates corresponding Genesys log events of the Standard (23011), Alarm (23012), Interaction (23022), Trace (23023) or Debug (23026) levels. The value of `scxml-log-filter-level` option can contain:

- A semi-colon list of colon-delimited pairs of custom and Genesys log levels.
- A semi-colon list of custom log levels (no explicit mapping to Genesys log levels). ORS always generates a log event of the Standard (23011) level.
- Any combination of both value types listed above.

Setting the `scxml-log-filter-level` option value to 1, 2, 3, 4 or 5 will override the logging results of the `map-composer-log-levels` option.

For example, if the SCXML application contains:

```
<log> action element:  
    <log expr="'Custom Log Level 45'" label="'Test example'" level="45" />  
    <log expr="'Custom Log Level 222'" label="'Test example'" level="222" />
```

or `__Log()` function:

```
__Log('Custom Log Level 45','Test example','45');  
__Log('Custom Log Level 222','Test example','222');
```

a. `scxml-log-filter-level = 45:TRC;222:INT`

ORS will generate following events:

```
Trc 23023 METRIC <log sid='31RJ0AHJQP3851BFQ6Q5JR0LA4000001' expr='Custom Log Level 45'  
label='Test example' level='45' />  
Int 23022 METRIC <log sid='31RJ0AHJQP3851BFQ6Q5JR0LA4000001' expr='Custom Log Level 222'  
label='Test example' level='222' />
```

b. `scxml-log-filter-level = 45;222`

ORS will generate following events:

```
Std 23011 METRIC <log sid='31RJ0AHJQP3851BFQ6Q5JR0LA4000001' expr='Custom Log Level 45'  
label='Test example' level='45' />  
Std 23011 METRIC <log sid='31RJ0AHJQP3851BFQ6Q5JR0LA4000001' expr='Custom Log Level 222'  
label='Test example' level='222' />
```

c. `scxml-log-filter-level = 45:TRC;222`

ORS will generate following events:

```
Trc 23023 METRIC <log sid='31RJ0AHJQP3851BFQ6Q5JR0LA4000001' expr='Custom Log Level 45'  
label='Test example' level='45' />  
Std 23011 METRIC <log sid='31RJ0AHJQP3851BFQ6Q5JR0LA4000001' expr='Custom Log Level 222'  
label='Test example' level='222' />
```

Note: For more information on the `__Log` function, refer to the function description on [this page](#).

`sessionfm-fetch-timeout`

Option section: `orchestration`

Configuration object: `ORS Application object`

Default value: `60`

Valid values: any Any integer from 0 to 86400

Value changes: Immediately

This option, introduced in ORS Release 8.1.300.32, allows you to set a custom value for the timeout attribute used in the <session:fetch> action element. The option specifies the timeout in seconds that ORS will use when the timeout attribute is omitted in the <session:fetch> action element.

When ORS is operating on Windows, due to the operating system default settings, the timeout reaches a 21-second maximum regardless of the value configured in the `sessionfm-fetch-timeout` option.

`session-hung-timeout`

This option is obsolete and no longer used.

`statserver-source`

Object: Transaction of Statistic type

Option section: `orchestration`

Default value: False

Valid values: False, True

Value changes: Upon ORS restart.

This option, introduced in Release 8.1.400.17, defines the source for the statistical data in the Statistic configuration. For more information, see [Direct Statistic Subscription](#).

`switch-multi-links-enabled`

Option section: `orchestration`

Configuration object: ORS Application object

Default value: false

Valid values: true or false

Value changes: after restart

This option can be used as follows:

- It determines whether ORS supports multi-link switch configurations (load sharing Network T-Servers or IVR Servers). When set to true, ORS enables support of multi-link switch configuration. When set to false, ORS does not enable multi-link switch support.
- This option must be set to true to enable support for [pulling from multiple Interaction Servers](#). Note: For multiple Interaction Server support, it is not necessary to set this option on the load sharing Switch object as described below.

Load Sharing Switch Object

The `switch-multi-links-enabled` option, specified on the load sharing switch object (`Switches\SharedSwitchObject\Annex\gts`), determines whether the switch is working in load-balancing mode; that is, it is served by multiple Network T-Servers or IVR T-Servers. Orchestration uses this option to determine whether to enable connection to more than one Network T-Server or IVR T-Server serving this switch.

`switch-multi-links-enabled`

Default value: 0

Value	Description
1	A network or IVR switch in load-balancing mode.
Any other integer	Not a network or IVR switch in load-balancing mode.

Changes Take Effect: After ORS restart.

This option should be used only in a configuration in which Network T-Servers or IVR T-Servers are working in load-balancing mode; that is, when there is no duplication in notification events received in Orchestration via connections to these T-Servers. Currently, load balancing mode is supported only for Network T-Servers and IVR T-Servers. This option was introduced in ORS 8.1.2+.

`statserver-source`

Object: Transaction of Statistic type

Option section: `orchestration`

Default value: False

Valid values: False, True

Value changes: Immediately upon notification.

Starting with Release 8.1.400.17, ORS can now request data directly from Stat Server instead of going through Universal Routing Server. This option defines the source for the statistical data in the Statistic configuration. For more information, see [Requesting data directly from Stat Server](#).

`thread-synch-ipv`

Option section: `orchestration`

Configuration object: ORS Application object

Default value: any

Valid values: any, 4, 6.

Value changes: During startup. Option changes take effect after restart only.

This option can improve strategy execution time by preventing delays that can occur when establishing the initial connection to session processing threads. The option specifies the protocol version for inter-thread communication in ORS. The value any indicates any version (this mode was used before introducing the option). The value 4 indicates the IPv4 protocol version. The value 6 indicates the IPv6 protocol version. Mixed modes are not allowed, such as 4, 6 or 6, 4.

For example: `thread-synch-ipv=4`

`webfm-event-hold-response`

Option section: `orchestration`

Configuration object: `ORS Application object`

Default value: `true`

Valid values: `true` or `false`

Value changes: Immediately

When this option is set to `true`, the HTTP response depends on how the event is processed:

- If a transition has been selected as a result of the event, response contains headers `Status-Code: 200`, `Reason-Phrase: OK`
- If no transition has been selected, response contains headers `Status-Code: 204`, `Reason-Phrase: No Content`
- If an error has occurred while processing the event, response contains headers `Status-Code: 400`, `Reason-Phrase: Bad Request`

When the option is set to `false`, once the event has been successfully published, ORS responds with `200 OK`.

`print-cache-stats-interval`

Option section: `orchestration`

Configuration object: `ORS Application object`

Default value: `60` (seconds)

Valid values: Any positive integer greater than 10

Value changes: Immediately upon notification

This option specifies the time interval, in seconds, for logging level 3 SCXML internal cache statistics messages in the `ScxmlCacheStats` log segment.

`sessionid-with-nodeid`

Option section: `orchestration`

Configuration object: `ORS Application object`

Default value: false

Valid values: true, false

Value changes: After ORS restart

This option is used to enable the new session ID generation scheme, introduced in 8.1.400.70. The scheme minimizes Cassandra lookups to determine node/session relations wherever possible. Custom session IDs are not affected. ORS works with old style session IDs, even if the new ID generation scheme is enabled. Different nodes and/or ORS applications in a cluster can use different ID generation schemes. Introduced in 8.1.400.70.

act-on-behalf-of

Option section: orchestration

Configuration object: ORS Application object

Default value: An empty string

Valid values: List of excluded ORS Node IDs separated by a comma or semicolon

Value changes: Immediately (this is a dynamic option)

Introduced in 8.1.400.70, this option is used to designate an existing node in the ORS cluster to act on behalf of an excluded node. It is the user's responsibility to ensure that only one existing node is acting on behalf of any excluded node. This option must be used only in unavoidable exceptional situations.

multivalued-prop-in-udata

Option section: orchestration

Configuration object: ORS Application object or Enhanced Routing Script object

Default value: false

Valid values: true, false

Value changes: Immediately; new value will take effect for sessions created after option change. When configured in an Enhanced Routing Script object, that option value takes precedence over the option configured in the ORS Application object.

Introduced in 8.1.400.70, this option is used to allow presentation of all values of key-value pairs having the same key name in an interaction's user data. All the values are provided in the form of an array ordered the same way the values were presented in the interaction user data. ORS does not provide a way to modify the multivalued key-value pairs in the interaction user data. But it does allow to use this form of multivalued property in other actions (for example, in the `createcall` action or the `<session:fetch method="esp">` action, where input parameters are specified as an ECMAScript object)). Using this option on an Enhanced Routing Script object may help avoid problems with SCXML strategies that expect only a single-value representation for a multivalued key.

web-start-accept-url

Option section: `orchestration`

Configuration object: `ORS Application object`

Default value: `true`

Valid values: `true`, `false`

Value changes: Dynamically

Introduced in 8.1.400.74, this option controls if an arbitrary URL can be used as a value for the `src` parameter in a `<session:start>` action or an `/scxml/session/start` web request, when starting a new session. When this option is set to `true`, ORS allows a URL to be used as a value for the `src` parameter. When this option is set to `true`, ORS allows an URL to be used as a value for the `src` parameter. When set to `false`, ORS accepts data only in the `script:<ERScript name>` format as a value for the `src` parameter. If data is not in the specified format, an HTTP error message error is returned.

use-only-enabled-script

Option section: `orchestration`

Configuration object: `ORS Application object`

Default value: `false`

Valid values: `true`, `false`

Value changes: Dynamically

Introduced in 8.1.400.79, this option controls whether or not ORS uses Enhanced Routing Script objects for session creation, based on object state. If this option is set to `true`, ORS uses only enabled Enhanced Routing Script objects. Otherwise, any Routing Script object can be used regardless of its state.

fin-acts-on-calldeleted

Option section: `orchestration`

Configuration object: `ORS Application object`

Default value: `false`

Valid values: `true`, `false`

Value changes: Dynamically

Introduced in 8.1.400.79, this option enables ORS to finalize all pending actions associated with a call upon receiving a `CallDeleted` event after the call is disconnected. In a blind transfer scenario to Route Point, ORS now generates an `error.interaction.redirect` event after it receives an `EventError` for the `<ixn:redirect>` action applied to a consultation call which is deleted due to transfer completion. To enable ORS to generate an `error.interaction.redirect` event in such

scenarios, set this option to `true`.

esp-response-binary-conversion

Option section: `orchestration`

Configuration object: ORS Application object

Default value: `string`

Valid values: `ignore`, `string`, `array`

Value changes: Immediately

Introduced in 8.1.400.87, this option allow users to define the way binary attributes of an ESP response message are converted to the JavaScript (JS) property of the corresponding SCXML event.

- `ignore` - binary attribute value is not added as a JS property in the SCXML event;
- `string` - value is converted to string and control characters like CR, LF, and so on, are converted to escaped characters;
- `array` - each byte is converted to a numeric element of an array.

keep-original-sesid-on-switchover

Option section: `orchestration`

Configuration object: ORS Application object

Default value: `false`

Valid values: `true`, `false`

Value changes: Immediately (new value will apply only to sessions that are created after the change)

Beginning with release 8.1.400.87, this option provides a possibility to retain the original session ID for backup voice sessions which have been restarted on ORS switchover. Set this option to `true` to allow other VXML/SCXML sessions to communicate with the restarted session. The existing `restart-session-on-switchover` option must also be set to `true`.

gcti-block-redirect-trans-on-queued

Option section: `orchestration`

Configuration object: ORS Application object

Default value: `0`

Valid values: `0`, `1`

Value changes: After restart

Beginning with release 8.1.400.93, this option allows handling redirect scenarios when a call is queued at the Routing Point with `AttributeCallState 23`. Setting this option's value to `1` successfully performs a `<ixn:redirect>` of such calls. Use this option to enable or disable the special mode protecting the stuck *redirect* transactions on `EventQueued (AttributeCallState 23)`.

- `1` - allows to perform either `<ixn:redirect>` or `<queue:submit>` (with `route=true`) actions to distribute calls from the Routing Point.
- `0` - is used for backward compatibility and preserves the old ORS behavior. Only the `<queue:submit>` action (with `route=true`) action can be used to distribute calls from the Routing Point.

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.

Introduced in 8.1.400.45. See [Recovery of Voice Calls Without Persistence](#). 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.

Introduced in 8.1.400.45. See [Recovery of Voice Calls Without Persistence](#). 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 session-processing-threads option in the ORS scxml section (default = 8).

sessionfm-fetch-esp-timeout

Option section: orchestration

Configuration Object: ORS Application object

Default value: 60

Valid values: Any integer from 10 to 86400

Value changes: Immediately.

Introduced in 8.1.401.04. Use this option to specify the timeout in seconds that ORS will use when the timeout attribute is omitted in the <session:fetch> action with the esp method. Previously, for such requests ORS used a hardcoded value of 60 seconds.

user-data-for-esp-hints-tpserver

Option section: `orchestration`

Configuration Object: ORS Application object

Default value: `false`

Valid values: `true`, `false`

Value changes: Immediately upon notification from Configuration Server.

Introduced in 8.1.401.04. Use this option to ensure that an interaction's user data is added to the ESP request sent to the third-party server specified in the `tpserver` property of the `hints` object of the `<ixn:terminate>` action. Previously, the ESP request was sent without user data.

persistence Section

This section describes persistence section options.

type

Option section: `persistence`

Configuration object: ORS Application object

Default value: `cassandra`

Valid values: `cassandra`, `redis`

Value changes: take effect during startup. Changes to this option are not applied dynamically.

This option specifies what type of session recovery must be used. Only supported types are Cassandra and Redis Cluster. If the option value is different from `redis`, ORS will try connect to Cassandra. Changing this option after startup will not have an effect.

redis-nodes

Option section: `persistence`

Configuration object: ORS Application object

Default value:

Valid values: A list of host names and ports for redis nodes in the format of `<HOST>:<PORT>; [<HOST>:<PORT>...]`

Value changes: take effect during startup. Changes to this option are not applied dynamically.

This option provides hosts and corresponding ports when using the Redis Cluster based persistence type. This is a semi-colon separated list of host and port of Redis Cluster nodes in the format, <HOST>:<PORT>. A host can be a hostname or an IP address.

For example: `persistence/redis-nodes = rhost:6379;143.244.220.15:8301;test47:1234.`

It is sufficient to configure a single alive node for ORS to connect to all cluster nodes.

`username`

Option section: `persistence`

Configuration object: `ORS Application object`

Default value:

Valid values:

Value changes: take effect during startup. Changes to this option are not applied dynamically.

Specifies the username that is used to authenticate with Cassandra or Redis Cluster.

`password`

Option section: `persistence`

Configuration object: `ORS Application object`

Default value:

Valid values:

Value changes: take effect during startup. Changes to this option are not applied dynamically.

Specifies the password that is used to authenticate with Cassandra or Redis Cluster.

`redis-connect-timeout`

Option section: `persistence`

Configuration object: `ORS Application object`

Default value: `100`

Valid values: Any integer value

Value changes: take effect during startup. Changes to this option are not applied dynamically.

Specifies the timeout in milliseconds, that Redis++ client library will use for establishing connection. It is important to configure this timeout as a small value to avoid blocking ORS worker thread execution if Redis becomes unavailable and reconnection is attempted as part of executing the Redis request.

If set to 0, the timeout is disabled.

redis-socket-timeout

Option section: `persistence`

Configuration object: `ORS Application object`

Default value: `100`

Valid values: Any integer value

Value changes: take effect during startup. Changes to this option are not applied dynamically.

ORS communicates with Redis synchronously. This option defines the timeout in milliseconds, Redis++ client library will use to send read or write request to Redis. It is important to configure this timeout as a small value to avoid blocking ORS worker thread execution if Redis becomes unavailable.

Note: Actual max time for execution of Redis request can exceed this option value.

If set to 0, request to Redis will never timeout and block ORS worker thread until completed successfully or get an error.

tls-enabled

Option section: `persistence`

Configuration object: `ORS Application object`

Default value: `false`

Valid values: `true, false`

Value changes: take effect during startup. Changes to this option are not applied dynamically.

If set to `true`, connections to the Redis instances will be encrypted. For details regarding Redis TLS(SSL), refer to Redis documentation, <https://redis.io/docs/connect/cli/>.

Note: TLS is not supported on Windows platform.

tls-cert

Option section: `persistence`

Configuration object: `ORS Application object`

Default value:

Valid values:

Value changes: take effect during startup. Changes to this option are not applied dynamically.

If the option **tls-enabled** is set to `true`, you can specify in this option a file name of TLS certificate. Example: `/path/to/client/certificate`

tls-key

Option section: persistence

Configuration object: ORS Application object

Default value:

Valid values:

Value changes: take effect during startup. Changes to this option are not applied dynamically.

If the option **tls-enabled** is set to `true`, you can specify in this option a corresponding file name of TLS private key to the certificate configured in **tls-cert**. Example: `/path/to/private/key/file`

tls-cacert

Option section: persistence

Configuration object: ORS Application object

Default value:

Valid values:

Value changes: take effect during startup. Changes to this option are not applied dynamically.

If the option **tls-enabled** is set to `true`, you can configure a trusted root certificate bundle using this option. Example: `/path/to/CA/certificate/file`

tls-cacertdir

Option section: persistence

Configuration object: ORS Application object

Default value:

Valid values:

Value changes: take effect during startup. Changes to this option are not applied dynamically.

If the option **tls-enabled** is set to `true`, you can configure a trusted root certificate directory instead of **tls-cacert**.

tls-sni

Option section: persistence

Configuration object: ORS Application object

Default value:

Valid values:

Value changes: take effect during startup. Changes to this option are not applied dynamically.

If the option **tls-enabled** is set to `true`, you can specify (optional) in this option, a Server Name Indicator (SNI) value.

cassandra-connect-attempt-timeout

Option section: `persistence`

Configuration object: `ORS Application object`

Default value: `2000`

Valid values: Any integer greater than or equal to 1

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the maximum time, in milliseconds, allowed for a connection to the Cassandra cluster to complete.

For example: `persistence/cassandra-connect-attempt-timeout = 5000`

Configuring a value for `cassandra-connect-attempt-timeout` above the default value could cause excessive delays in Orchestration startup if the Cassandra store is not available.

cassandra-keyspace-name

Option section: `persistence`

Configuration object: `ORS Application object`

Default value: `Orchestration`

Valid values: Strings of alpha-numeric characters and underscores. Must begin with a letter.

Value changes: take effect during startup. Changes to this option are not applied dynamically.

This option specifies the name of the Cassandra keyspace to use for this ORS cluster. This option would be used if you plan to run multiple distinct ORS clusters using the same Cassandra cluster. This option must be unique for an Orchestration cluster. If nodes in different Orchestration clusters inadvertently define the same keyspace, cross-contamination of persisted data will lead to unpredictable operation in a failover scenario.

For example: `persistence/cassandra-keyspace-name = ORS_Cluster_west`

For the most current information on using ORS with Cassandra, see the [Cassandra Installation/Configuration Guide](#).

cassandra-listenport

Option section: persistence

Configuration object: ORS Application object

Default value: 9160

Valid values: Any valid socket port.

Value changes: take effect during startup. Changes to this option are not applied dynamically.

This option provides the Cassandra client connection port when using the Cassandra-based persistence method. This option must be supplied with an appropriate value for correct Cassandra-based persistence operation to occur.

For example: persistence/cassandra-listenport = 9160

cassandra-max-latency

Option section: persistence

Configuration object: ORS Application object

Default value: 400

Valid values: Any non-negative integer greater than 100 and less than 5000

Value changes: take effect during startup. Changes to this option are not applied dynamically.

Maximum request latency allowed before messages are logged warning of excessive delays in response to Cassandra requests. If subsequent requests have a latency less than this value a message is logged indicating the excessive latency condition has cleared. This option is not required, and the default is 400 msec.

cassandra-nodes

Option section: persistence

Configuration object: ORS Application object

Default value: unknown

Valid values: A list of host names for Cassandra nodes in the Cassandra cluster.

Value changes: take effect during startup. Changes to this option are not applied dynamically.

This option provides the Cassandra client connection port when using the Cassandra-based persistence method. This is a semi-colon separated list of host names or IP addresses.

For example: persistence/cassandra-nodes = DWS;mpswin;test47

cassandra-read-timeout

Option section: persistence

Configuration object: ORS Application object

Default value: 500

Valid values: Any integer value from 201 to 4999

Value changes: Takes effect after restart.

This option, introduced in ORS Release 8.1.300.47, defines the maximum time, in milliseconds, that ORS will wait for a Cassandra connection to become ready to read. If this time is exceeded, ORS considers the connection to be lost.

cassandra-schema-version

Option section: persistence

Configuration object: ORS Application object

Default value: ORS8130000

Valid values: This is a required parameter, which may be left at the default value or set to a previous schema version, but it must agree with the Column Family name for the cassandra cluster to which ORS is connecting. This is a string and must be of the form ORSddddddd, where ddddddd must be numeric

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the schema version that either exists in the Cassandra cluster, or is to be loaded automatically by Orchestration upon the first successful connection to the Cassandra cluster.

On startup and connection to Cassandra, Orchestration verifies that the requested schema agrees with the existing Cassandra schema, if the schema has been previously loaded. Or, it loads the schema automatically at the schema version defined by the default.

If the schema disagrees or is not defined, Orchestration will continue to operate, but without persistence, and therefore, without the ability to restore sessions or scheduled events.

For example: persistence/cassandra-schema-version = ORS8130000

cassandra-strategy-class

Option section: persistence

Configuration object: ORS Application object

Default value: SimpleStrategy

Valid values: SimpleStrategy or NetworkTopologyStrategy.

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the strategy class that Cassandra uses for the cluster. `SimpleStrategy` defines a single cluster, without multiple Data Centers.

The `NetworkTopologyStrategy`, which is network strategy in conjunction with the Cassandra-topology properties file (located in each Cassandra instance install configuration directory), defines the Data Centers for the Cassandra cluster. Multiple Data Centers typically are geographically dispersed.

For the most current information on Cassandra, see the [Cassandra Installation/Configuration Guide](#).

For example: `persistence/cassandra-strategy-class = SimpleStrategy`

`cassandra-strategy-options`

Option section: `persistence`

Configuration object: ORS Application object

Default value: Unknown

Valid values: `replication_factor:N`, where N is the replication factor desired, or `DC1:K;DC2:J...` Where DC1, etc. are the Data Center names defined in the topology properties file, and K, J, etc. are the replication factors for each Data Center.

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the `replication_factor` to be configured for the given Orchestration keyspace. The two variations apply to `SimpleStrategy` or `NetworkTopologyStrategy` respectively. See the [Cassandra Installation and Configuration Guide](#).

Examples:

- If `SimpleStrategy`, `cassandra-strategy-options = replication_factor:2`
- If `NetworkTopologyStrategy`, `cassandra-strategy-options = DC1:2;DC2:3`

`cassandra-thread-count`

Option section: `persistence`

Configuration object: ORS Application object

Default value: 8

Valid values: Any integer greater than or equal to 1

Value changes: During startup. Changes to this option are not applied dynamically.

This option sets the number of threads in the persistence worker thread pool. Recommended value: 2x the number of cores.

For example: `persistence/cassandra-thread-count = 4`

cassandra-write-timeout

Option section: persistence

Configuration object: ORS Application object

Default value: 500

Valid values: Any integer value from 201 to 4999

Value changes: Takes effect after restart.

This option, introduced in ORS Release 8.1.300.47, defines the maximum time, in milliseconds, that ORS will wait for a Cassandra connection to become ready to write. If this time is exceeded, ORS considers the connection to be lost.

max-cache-count

Option section: persistence

Configuration object: ORS Application object

Default value: 100000

Valid values: Any integer greater than or equal to 10000

Value changes: During startup. Changes to this option are not applied dynamically.

Defines the number of entries allowed within the internal Orchestration persistence cache.

For example: persistence/max-cache-count = 150000

For example: persistence/max-cache-count =

When connection to persistent storage is disabled or persistent storage is not available, Orchestration Server removes the cached data of terminated sessions. This enables the processing of long-lived sessions, if the number of current active sessions does not exceed the number specified in the max-cache-count option.

max-cache-size

Option section: persistence

Configuration object: ORS Application object

Default value: 100,000,000

Valid values: Any integer greater than or equal to 0

Value changes: During startup. Changes to this option are not applied dynamically.

This option sets the maximum size of the amount of memory (in bytes) that the internal Orchestration persistence cache can use.

For example: `persistence/max-cache-size = 100000`

Overload Protection Logic for Large SCXML Strategies

Beginning with version 8.1.401.02, ORS offers overload protection when an assembled SCXML document's size exceeds the assembled or compiled cache entry size. This is applicable only for digital interactions and enhanced pulling.

This feature is used to slowly start loading strategies that have not been executed before (or not executed after any changes related to strategy configuration).

- This feature is applicable for all strategies (which are represented by Enhanced Routing Script objects) that are used for enhanced pulling.
- ORS creates a *pulling* set where strategies can be marked as confirmed or non-confirmed
- When a pulling set is created – all strategies are non-confirmed
- A strategy is confirmed if ORS successfully creates a session for that strategy.
- When a session is started for a strategy, the strategy is confirmed.
- When session creation fails for a strategy, the strategy is non-confirmed.
- For confirmed strategies, the ORS pull behavior is unchanged, that is, it pulls according to the cycle quota.
- For non-confirmed strategies, ORS pulls only one interaction per strategy per pull cycle.
- When an Enhanced Routing Script object representing a strategy is changed, the strategy changes to non-confirmed
- When ORS is switched from the backup mode, all strategies are in the non-confirmed state.

The following new option is introduced to control the overload protection logic:

`mcr-pull-gradual-start`

Option section: `orchestration` in an ORS Application object

Configuration object: ORS Application object

Default value: `true`

Valid values: `true`, `false`

Value changes: Immediately upon notification from the Configuration Server

When set to `true`, ORS starts pulling with only 1 interaction per cycle for the strategy, until sessions for that strategy are successfully started (that is, the strategy is confirmed). Then, ORS starts pulling for that strategy according to the cycle quota. If creation of session fails (strategy is non-confirmed), the pulling rate for this strategy drops to 1 until the strategy is confirmed again. Any changes in strategy configuration (Enhanced Routing Script) will change the status of the strategy to non-confirmed.

Controlling session lifetime functionality for deployments without Cassandra

persistence

Beginning with version 8.1.401.08, controlling session lifetime functionality is extended for deployments without Cassandra persistence. Previously, a connection to Cassandra was required for this functionality. The maximum session age in seconds is configured in the `max-session-age` option or in the `_maxtime` attribute in the `<scxml>` action element.

The following new ORS options are introduced to support this enhancement:

extend-max-session-age-action

Option section: `orchestration` in an ORS Application object

Configuration object: ORS Application object

Default value: `false`

Valid values: `true`, `false`

Value changes: Dynamically

This option is used to enable or disable session termination. It is semi-dynamic for sessions started before changing the value of the option and fully dynamic for new sessions. That is, changes take effect immediately for sessions that are started after the value of this option is updated.

max-session-age-notification-timeout

Option section: `orchestration` in an ORS Application object

Configuration object: ORS Application object

Default value: `15`

Valid values: Any integer from 0 to 86400 (seconds)

Value changes: Dynamically (for new sessions when `extend-max-session-age-action` is set to `true`) This option controls when ORS notifies a session in advance that it is about to be terminated using the `session.terminating.forcedly` event by default. This option allows to specify the time in seconds before reaching max session age at which to send the event.

This option value should be less than the `max-session-age` or `_maxtime` values. Otherwise, the option is ignored (set to 0), and no notification event will be sent.

override-notification-event-name

Option section: `orchestration` in an ORS Application object

Configuration object: ORS Application object

Default value: An empty string (the default event name is used if no value is provided)

Valid values: Any notification event name

Value changes: Dynamically (for new sessions when `extend-max-session-age-action` is set to `true`)

This option is used to specify any notification event name instead of the default event. It is not recommended to override the notification event with something that starts with `error.`, because this may potentially break backward compatibility. For some specific strategies generated by recent

versions of Designer for Azure, you can override this recommendation.

If you rely on `_genesys_terminate_event` processing when Cassandra is in use, set this option to `_genesys_terminate_event`.

scxml Section

This section describes `scxml` section options.

assembled-cache-reload-threshold

Option section: `scxml`

Configuration object: ORS Application object

Default value: 45

Valid values: Any non-negative integer

Value changes: Takes effect after restart

Options `max-assembled-cache-age` and `assembled-cache-reload-threshold` allow you to fine-tune the behavior of the Assembled Document Cache. This option determines the period of time, in seconds, after which the pre-emptive re-validation of a cached entry occurs. When the age of a cached document in the Assembled Document Cache exceeds the `assembled-cache-reload-threshold`, the next request re-validates the document and updates the cache.

debug-enabled

Option section: `scxml`

Configuration object: ORS Application object

Default value: `false`

Valid values: Boolean

Value changes: During startup. Changes to this option are not applied dynamically.

This option allows you to specify whether debugging with **Genesys Composer** is enabled in the system. If it is enabled, you must also specify the `debug-port`.

debug-port

Option section: `scxml`

Configuration object: ORS Application object

Default value: 7999

Valid values: any available port

Value changes: During startup. Changes to this option are not applied dynamically.

This option allows you to specify and enable or disable the port to be used by the debug client **Genesys Composer** to interact with the SCXML Engine during a debug session.

default-encoding

Option section: `scxml`

Configuration object: ORS Application object

Default value: UTF-8

Valid values: Any valid converter name supported by ICU.

Value changes: After restart

This option defines the source encoding that ORS will use to encode all input String data into Unicode before it is passed into the SCXML engine. Input String data includes:

1. SCXML and JavaScript String literals
2. variables
3. properties of String type
4. values of String type message attributes from the incoming API such as TLib and Ixn
5. values of String type properties of Configuration Objects retrieved from Configuration Server, and so on.

This option also defines the destination encoding that ORS will use to encode all SCXML Actions, function attributes, and parameters of String type from Unicode before those values will be used as attribute values in all ORS external APIs like TLib, Ixn, HTTP, and so on.

fetch-timeout

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

Configuration object: ORS Application object; Enhanced Routing Script object

Default value: 5000

Valid values: Any integer greater than 0

Value changes: After restart

This option specifies the time (in milliseconds) before a document fetch operation is abandoned. If the SCXML document cannot be retrieved within this timeout value, the fetch operation is abandoned and a fetch operation for the `alternate-url` is attempted.

fips-enabled

Option section: `scxml`

Configuration object: ORS Application object

Default value: false

Valid values: true or false

Value changes: During startup. Changes to this option are not applied dynamically.

Enables FIPS compliance in the SCXML library.

For example: `scxml/fips-enabled = false`

Note: With the newer OpenSSL 3.x that comes supported with ORS 8.1.401.11+, in addition to setting the option `scxml\fips-enabled`, additional configuration steps are necessary to enable FIPS. For more information, see [Configuring FIPS for ORS on Windows](#).

http-client-side-port-range

Option section: `scxml`

Configuration object: ORS Application object

Default value: <empty string>

Valid values: A number between 1 and 65535 followed by a "-" and then another number between 1 and 65535; with the second number larger than the first number.

Value changes: During startup. Changes to this option are not applied dynamically.

Specifies the port range of the socket used for an HTTP client side connection. An empty value, or lack of option, indicates that the default should be used.

For example: `scxml/http-client-side-port-range = 1000-2000`

http-curl-dns-cache-timeout

Introduced in version 8.1.400.80, this option defines the TTL value for entries in the internal DNS cache (in seconds). If set as 0, internal DNS caching is disabled. If set to -1, cached DNS entries will have an infinite TTL.

Option section: `scxml`

Configuration object: ORS Application object

Default value: 60

Valid values: Any positive integer or 0 or -1

Value changes: After restart

http-curl-ip-family

Introduced in version 8.1.400.49, this option is used to explicitly define the IPv4 for connections used by the ORS SCXML engine to retrieve SCXML documents from a web server.

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `<empty string>`

Valid values: `ipv4`, `ipv6`

Value changes: During startup. Changes to this option are not applied dynamically.

Example: `scxml/http-curl-ip-family = ipv4`.

This option specifies the type of IP address to be used in the SCXML library for host name resolution upon SCXML document retrieval via HTTP protocol or execution of the SCXML action of the HTTP method (GET, POST, and so on). Use this option to eliminate any possible delays that may occur upon opening a connection to a web server if the host has enabled more than one version of IP protocol for both interfaces; such as both IPv4 and IPv6.

Starting with ORS 8.1.400.50, IPv6 is supported as a valid value.

`http-enable-continue-header`

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `false`

Valid values: `true` or `false`

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies whether to enable or disable the `100-continue` header in the `HTTP 1.1 post` request.

For example: `scxml/http-enable-continue-header = true`

`http-enable-keepalive`

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `false`

Valid values: `true` or `false`

Value changes: Changes take effect after restart.

This option, introduced in Release 8.1.300.35, enables keepalive transmissions on TCP sockets used by Orchestration Server. If set to `true`, enables keepalive transmissions on TCP sockets for fetching documents from an application server.

http-keepalive-idle

Option section: scxml

Configuration object: ORS Application object

Default value: 7200

Valid values: Any valid integer

Value changes: Take effect after restart.

This option, introduced in Release 8.1.400.39, defines the time between keepalive transmissions in idle condition for TCP sockets for fetching documents from an application server.

http-keepalive-intvl

Option section: scxml

Configuration object: ORS Application object

Default value: 75

Valid values: Any valid integer

Value changes: Take effect after restart.

This option, introduced in Release 8.1.400.39, defines the time interval between keepalive re-transmissions, if an acknowledgement to the previous keepalive transmission was not received.

http-max-age

Option section: scxml

Configuration object: ORS Application object

Default value: 60

Valid values: Any integer greater than or equal to 0

Value changes: During startup. Changes to this option are not applied dynamically.

Defines the value of the max-age directive of the cache-control header in HTTP requests for documents that the SCXML library retrieves from the web server upon SCXML application assembly. Also, if the response from the server does not change it, the value of that option defines the time interval for how long a cached copy of a document will be used without re-validation. However, since an SCXML application usually consists of multiple documents, that option cannot be used to change the interval until the next re-validation of the entire SCXML application. For that purpose, please use options assembled-cache-reload-threshold and max-assembled-cache-age.

Note: If max-age is configured for an **Enhanced Routing Script** object, the value from max-age takes precedence over the http-max-age Application level option.

http-max-age-local-file

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `60000`

Valid values: Any integer greater than or equal to 0

Value changes: During startup. Changes to this option are not applied dynamically.

This option sets the maximum age of the local file in milliseconds.

For example: `scxml/http-max-age-local-file = 65000`

http-max-cache-entry-count

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `1000`

Valid values: Any integer greater than or equal to 0

Value changes: During startup. Changes to this option are not applied dynamically.

This option sets the maximum number of entries that can be stored in the cache.

For example: `scxml/http-max-cache-entry-count = 1250`

http-max-cache-entry-size

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `100000`

Valid values: Any integer greater than or equal to 0

Value changes: During startup. Changes to this option are not applied dynamically.

This option sets the maximum size of each cache entry in bytes.

For example: `scxml/http-max-cache-entry-size = 125000`

http-max-cache-size

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: 10000000

Valid values: Any integer greater than or equal to 0

Value changes: During startup. Changes to this option are not applied dynamically.

This option sets the maximum size of the HTTP cache in bytes.

For example: `scxml/http-max-cache-size = 12500000`

http-max-redirects

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: 5

Valid values: 0 - 100

Value changes: During startup. Changes to this option are not applied dynamically.

This option sets the maximum number of times to follow the `Location:` header in the HTTP response. Set to 0 to disable HTTP redirection.

For example: `scxml/http-max-redirects = 10`

http-max-stale

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: 60

Valid values: Any integer greater than or equal to 0

Value changes: During startup. Changes to this option are not applied dynamically.

Defines time in seconds to extend the lifetime of cached SCXML application.

For example, if the value of the `http-max-age` option is set to 60 seconds and value of the `max-stale` option is set to 30 seconds, then the fetched document will be cached for the timeframe of 90 seconds.

- If value is set to 0, the lifetime of a cached file will not be extended.
- To disable fetching SCXML applications from the cache, set `http-max-age` and `http-max-stale` to 0.
- If `max-stale` is configured for an Enhanced Routing Script object, the value from `max-stale` takes precedence over the `http-max-stale` Application level option.

http-no-cache-urls

Option section: `scxml`

Configuration object: ORS Application object

Default value: `<empty string>`

Valid values: Comma-delimited set of strings

Value changes: During startup. Changes to this option are not applied dynamically.

This option sets a comma-delimited list of substrings that would prevent a response from being cached, if the URL contains any of the substrings.

For example: `scxml/http-no-cache-urls=myserver.com, yourserver.com`

http-proxy

Option section: `scxml`

Configuration object: ORS Application object

Default value: `<empty string>`

Valid values: Valid IP Address : Port or URL: Port

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the HTTP proxy server (if applicable). If specified, all HTTP fetches done by the ORS will be done via this proxy server.

For example: `scxml/http-proxy = 127.0.0.1:3128`

`scxml/http-proxy = myproxy:3128`

http-ssl-ca-info

Option section: `scxml`

Configuration object: ORS Application object

Default value: `<empty string>`

Valid values: Valid file name

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the file name holding one or more CA certificates with which to verify the peer. This is only useful when `ssl-verify-peer=true`.

For example: `scxml/http-ssl-ca-info = myca.cer`

http-ssl-ca-path

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `" "`

Valid values: Valid path

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the path holding one or more CA certificates with which to verify the peer. The certificate directory must be prepared using the `openssl c_rehash` utility.

For example: `scxml/http-ssl-ca-path = c:\ssl\ca`

http-ssl-cert

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `<empty string>`

Valid values: Valid file name

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the file name of the SSL certificate.

For example: `scxml/http-ssl-cert = mycert.crt`

Spaces are not valid in the directory name that is specified in the file path for this option. For example `C:\Program Files\ Certificates\my_cert.pem` is not a valid path.

http-ssl-cert-type

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `PEM`

Valid values: `PEM`, `DER`

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the SSL Certificate Type: `PEM – PEM encoded certificate` `DER – DER encoded certificate`

For example: `scxml/http-ssl-cert-type = DER`

http-ssl-cipher-list

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `<empty string>`

Valid values: String value

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the cipher list as defined by OpenSSL. Paste the following url into your web browser to see valid cipher list formats:

http://www.openssl.org/docs/apps/ciphers.html#CIPHER_LIST_FORMAT

For example: `scxml/http-ssl-cipher-list=RC4-SHA`

http-ssl-key

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `<empty string>`

Valid values: Valid path or file name

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the path or file name of the SSL private key. For example: `scxml/http-ssl-key = c:\ssl\mykey.key`

Spaces are not valid in the directory name that is specified in the file path for this option.

For example `C:\Program Files\Certificates\my_cert.pem` is not a valid path.

http-ssl-key-type

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `PEM`

Valid values: `PEM`, `DER`

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the SSL Key Type:

`PEM` – PEM encoded key

DER – DER encoded key

For example: `scxml/http-ssl-key-type = DER`

`http-ssl-random-file`

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `<empty string>`

Valid values: Valid file name

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the file which is read from in order to see the random engine for SSL. The more random the specified file is, the more secure the SSL connection will become.

For example: `scxml/http-ssl-random-file=c:\ssl\random-seed`

`http-ssl-verify-host`

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `disable`

Valid values: `disable, common, match`

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies how the common name from the peer certificate should be verified during the SSL handshake.

- `disable` – the connection succeeds regardless of the names in the certificate
- `common` – the certificate must contain a “Common Name” field, but the field’s contents are not validated
- `match` – the certificate must indicate correct server name, or the connection will fail

For example: `scxml/http-ssl-verify-host = match`

`http-ssl-verify-peer`

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `false`

Valid values: `true or false`

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies whether the system should verify the peer's certificate. When this is true, either `http-ssl-ca-path` or `http-ssl-ca-info` would be set.

For example: `scxml/http-ssl-verify-peer = true`

`http-ssl-version`

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `default`

Valid values: `String (default, TLSv1, SSLv2 or SSLv3)`

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the SSL version to be used. By default, the system will determine the correct version to use. However, this option may be useful when some servers make it difficult to determine the correct SSL version.

For example: `scxml/http-ssl-version = SSLv2`

`http-verbose`

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `true`

Valid values: `true, false`

Value changes: Take effect after restart.

The option, released in 8.1.400.20, allows you to configure the amount of information written to the log when ORS reuses the connection for a new HTTP request.

`https-proxy`

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `<empty string>`

Valid values: `Valid IP Address: Port or URL: Port`

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the HTTPS proxy server (if applicable). If specified, all HTTPS fetches done by the ORS will be done via this proxy server.

For example: `scxml/https-proxy = 127.0.0.1:3128`

`js-preload-files`

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `ors.js`

Valid values: A semi-colon delimited list of JavaScript files

Value changes: During startup. Changes to this option are not applied dynamically.

This option provides a list of JavaScript files that are pre-loaded at ORS startup. If the full file path is not provided, ORS will look for files in the ORS working directory. This allows users to specify functions or variables that are available to all sessions.

The 8.1.400.21 release changed the default value for the `js-preload-files` option to `ors.js`. The `ors.js` file must always be defined to insure successful execution of SCXML applications if the `ixnid` parameter is not explicitly provided in the request.

`max-assembled-cache-age`

Option section: `scxml`

Configuration object: `ORS Application. object`

Default value: 60

Valid values: Any non-negative integer

Value changes: Takes effect after restart

Options `max-assembled-cache-age` and `assembled-cache-reload-threshold` allow you to fine-tune the behavior of the Assembled Document Cache. This option determines the expiration time, in seconds, of an entry in the Assembled Document Cache.

`max-assembled-cached-docs`

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: 1000

Valid values: Minimum = 0, no maximum value

Valid changes: Take effect after restart.

This option, introduced in ORS Release 8.1.300.48, determines the maximum number of documents to be stored in the assembled document cache. If this number is exceeded, the least referenced cached items will be removed from the cache to make room for new entries.

max-assembled-cached-doc-size

Option section: `scxml`

Configuration object: ORS Application object

Default value: 10000000

Valid values: Minimum = 0, no maximum value

Valid changes: Take effect after restart.

This option, introduced in ORS Release 8.1.300.48, determines the maximum allowed size (in bytes) of each individual entry in the cache. If a fully-assembled SCXML strategy exceeds this size, it cannot be stored in the cache.

max-assembled-cache-size

Option section: `scxml`

Configuration object: ORS Application object

Default value: 10000000

Valid values: Minimum = 0, no maximum value

Valid changes: Take effect after restart.

This option was introduced in ORS Release 8.1.300.48. Use `max-assembled-cache-size`, `max-assembled-cached-docs`, and `max-assembled-cached-doc-size` to configure the document cache by storing fully-assembled SCXML documents, eliminating the need for document fetches for `xincludes` in subsequent sessions.

This option determines the maximum allowed size (in bytes) of the assembled document cache. The size of the cache is determined by the total size of all cached documents stored in the cache.

max-cache-entry-count

Option section: `scxml`

Configuration object: ORS Application object

Default value: 1000

Valid values: Any integer greater than or equal to 0

Value changes: During startup. Changes to this option are not applied dynamically.

This option sets the maximum number of entries that can be stored in the cache. For example: `scxml/http-max-cache-entry-count = 1250`

max-cache-entry-size

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `100000`

Valid values: Any integer greater than or equal to 0

Value changes: During startup. Changes to this option are not applied dynamically.

This option sets the maximum size of each cache entry in bytes.

For example: `scxml/http-max-cache-entry-size = 125000`

max-compiler-cache-size

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `10000000`

Valid values: non-negative integer

Value changes: Takes effect after restart. Changes to this option are not applied dynamically.

The maximum amount of memory (in bytes) allowed for the `<xi:include>` compiler cache.

max-compiler-cached-docs

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `1000`

Valid values: non-negative integer

Value changes: Takes effect after restart. Changes to this option are not applied dynamically.

The maximum number of items that the `<xi:include>` compiler cache can have.

max-compiler-cached-doc-size

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `10000000`

Valid values: non-negative integer

Value changes: Takes effect after restart. Changes to this option are not applied dynamically.

The maximum size, in bytes, allowed to cache the results of the compiled `<xi:include>` document.

max-debug-sessions

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: 0

Valid values: an integer

Value changes: During startup. Changes to this option are not applied dynamically.

When debugging with Genesys Composer, this option allows you to set the maximum number of simultaneous debug sessions within the current SCXML Engine instance. Note that this value must be at least one digit less than the value specified in the `session-processing-threads` option to prevent session thread starvation. If it is not, it will default to 0.

max-includes

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: 500

Valid values: 0 to 10000

Value changes: During startup. Changes to this option are not applied dynamically.

This option sets the maximum number of documents that may be included using `<xi:include>`.

For example: `scxml/max-includes = 200`

max-microstep-count

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: 1000

Valid values: any integer (no maximum)

Value changes: During startup. Changes to this option are not applied dynamically.

For infinite loop protection, this option is used along with `max-state-entry-count`. Both options may be configured globally in ORS configuration, or may be configured per session by setting the attribute in the SCXML document:

```
<scxml _microStepLimit="1000" . />
```

```
<scxml _stateEntryLimit=100" ./>
```

For example, `max-microstep-count = 500`

This option specifies the maximum number of times in which transitions may be taken during a session without the processing of a new event. Once this value has been exceeded, the session is exited gracefully before being terminated by the system. A value of 0 indicates that the session is not to be forcibly terminated.

max-pending-events

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `100`

Valid values: positive integer between 30 and 100000, inclusive.

Value changes: Changes take effect During startup. Changes to this option are not applied dynamically.

This option specifies the maximum number of events allowed to be queued to a session (inclusive of internal, external, delayed and undelivered events). If this number is reached, ORS shall attempt to exit the session. This feature cannot be disabled.

For example: `scxml/max-pending-events = 1000`

max-preprocessor-cache-size

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `10000000`

Valid values: Any integer greater than or equal to 0

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the amount of memory, in bytes, that the `<xi:include>` pre-processor cache can use.

For example: `scxml/max-preprocessor-cache-size = 20000000`

max-preprocessor-cached-docs

Option section: `scxml`

Configuration object: `ORS Application object`

Default value: `1000`

Valid values: Any integer greater than or equal to 0

Value changes: During startup. Changes to this option are not applied dynamically.

This option sets the maximum number of items that the `<xi:include>` pre-processor cache can have.

For example: `scxml/max-preprocessor-cache-docs = 2000`

`max-preprocessor-cached-doc-size`

Option section: `scxml`

Configuration object: ORS Application object

Default value: 10000000

Valid values: non-negative integer

Value changes: Takes effect after restart. Changes to this option are not applied dynamically.

The maximum size, in bytes, of the cached results of the preprocessed `<xi:include>` documents.

`max-script-duration`

Option section: `scxml` (in the ORS Application object) or `Application` (in an Enhanced Routing Script object)

Configuration object: ORS Application object and Enhanced Routing Script object

Default value: 2,000 milliseconds

Valid values: 500 - 10,000 milliseconds

Value changes: When configured on an ORS Application object, takes effect during startup. When configured on an Enhanced Routing Script object, takes effect for sessions that started after that update. When configured in an Enhanced Routing Script object, that option takes precedence over the option configured in the ORS Application object.

If the execution time for JavaScript within a `<Script>` element exceeds the configured limit, execution will be interrupted, an Alarm-level message will be printed in ORS log, and the `error.script` Script event will be raised for the corresponding session.

`max-session-age`

Option section: `scxml`

Configuration object: ORS Application object

Default value: 604800

Valid values: A positive integer

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the maximum age in seconds that an ORS session should exist. If this age is reached, ORS shall attempt to exit the session.

To disable this feature, set the `max-session-age` to 0. The SCXML Engine supports `_maxtime` as an attribute on the `<scxml>` element, and also supports the option `max-session-age`, which is intended to queue a terminate event for the session after the time expired.

The value specified in the `_maxtime` attribute overrides the value in the `max-session-age` configuration option.

A connection to Cassandra is required for this functionality.

Orchestration Server sets the time to live in Cassandra for information required to support proactive session recovery. The time to live for the information in Cassandra is the configured `scxml/max-session-age` plus 3600 seconds. After this period, the data within Cassandra will be marked as deleted.

Note: Beginning with version 8.1.401.08, controlling session lifetime functionality is extended for deployments without Cassandra persistence. Previously, a connection to Cassandra was required for this functionality. Additional options, `extend-max-session-age-action`, `max-session-age-notification-timeout`, and `override-notification-event-name`, are available to support this ability. See [Controlling session lifetime functionality for deployments without Cassandra persistence](#) for option descriptions.

`max-state-entry-count`

Option section: `scxml`

Configuration object: ORS Application object

Default value: 500

Valid values: any integer (no maximum)

Value changes: During startup. Changes to this option are not applied dynamically.

For example, `max-state-entry-count = 250`

Starting with release 8.1.400.43, the default value of this option is changed from 100 to 500.

For infinite loop protection, this option is used along with `max-microstep-count`. Both options may be configured globally in ORS configuration, or may be configured per session by setting the attribute in the SCXML document:

```
<scxml _microStepLimit="1000" . />
```

```
<scxml _stateEntryLimit="100" . />
```

This option specifies the maximum number of times that a transition may be taken to a target state. Every state element within an SCXML document keeps an individual count of the number of times entered via a targeted transition. Once this value has been exceeded, the session is exited gracefully before being terminated by the system.

A value of 0 indicates that the session is not to be forcibly terminated.

password

Option section: `scxml`

Configuration object: ORS Application object

Default value: <empty string>

Valid values: String value

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the SSL key password.

For example: `scxml/password = agent007`

persistence-default

Option section: `scxml`

Configuration object: ORS Application object

Default value: `false`

Valid values: `true` or `false`

Value changes: Takes effect after restart. Changes to this option are not applied dynamically.

This option defines the default session **persistence**. If set to `true`, then Orchestration will persist the session if Cassandra is available. If set to `false`, Orchestration will not attempt to persist the session.

Note: In your routing strategy, you can set the `_persist` attribute in the `<scxml>` element, which will take precedence over the value set in the `persistence-default` Application level option.

persistence-max-active

Option section: `scxml`

Configuration object: ORS Application object

Default value: `10000`

Valid values: `100` - `1000000`

Value changes: During startup. Changes to this option are not applied dynamically.

This option allows setup of a maximum number of active sessions that the SCXML engine will keep in memory.

For example: `scxml/persistence-max-active = 5000`

process-event-timeout

Option section: `scxml`

Configuration object: ORS Application object

Default value: 10000

Valid values: A positive integer

Value changes: During startup. Changes to this option are not applied dynamically.

This option specifies the maximum time (in milliseconds) allotted for the processing of the event queue. The processing of one event may lead to additional events being queued. Processing of the event queue does not complete until the event queue is empty. This feature sets an upper bound to the amount of time dedicated to processing these events. If the timeout is reached, ORS attempts to exit the session. To disable this feature, set the `process-event-timeout` to 0.

For example: `scxml/process-event-timeout = 60000`

session-processing-threads

Option section: `scxml`

Configuration object: ORS Application object

Default value: 8

Valid values: Integer from 1 to 128 (inclusive)

Value changes: During startup. Changes to this option are not applied dynamically.

This option sets the number of threads in the thread pool. The recommended value is two times the number of cores.

For example: `scxml/session-processing-threads = 16`

system-id

Option section: `scxml`

Configuration object: ORS Application object

Default value: -1

Valid values: Any integer greater than or equal to -1

Value changes: During startup. Changes to this option are not applied dynamically.

This option allows setup of the ORS system ID in order to have unique session IDs across ORS instances. If -1 is specified, ORS creates a session ID based on the IP address of the host running ORS.

For example: `scxml/system-id = 11`

Optimized Initial Loading of Large SCXML Strategies

Beginning with version 8.1.401.02, ORS is improved to optimize initial loading of large SCXML strategies and thereby enhance performance. The functionality optimizes initial loading strategies by:

- Providing an option to increase performance of retrieving SCXML strategies via HTTP, for socket and CURL buffers.
- **Implementing overload protection** logic during initial start of SCXML strategy.
- Providing an **option** that causes session creation to fail if the SCXML strategy's size exceeds the size of compiled cache.

The following three new configuration options are introduced to increase performance of retrieving SCXML strategies via HTTP. Use these options to increase the send buffer for sockets used by CURL, and the internal CURL receive buffer.

http-buffer-size

Option section: `scxml` in an ORS Application object

Configuration object: ORS Application object

Default value: 131072

Valid values: Any integer between 1024 and 524288 (`CURL_MAX_READ_SIZE`)

Value changes: After restart

This option sets the `CURLOPT_BUFFERSIZE` option that represents the read buffer size, in bytes.

curl-socket-rcv-buffer-size

Option section: `scxml` in an ORS Application object

Configuration object: ORS Application object

Default value: None

Valid values: Determined by OS; 0 indicates that the option will not be set

Value changes: After restart

This option sets the `S0_RCVBUF` option (receive buffer size) for the socket used by CURL. Option will be set only if the new value is bigger than the current socket buffer size.

curl-socket-snd-buffer-size

Option section: `scxml` in an ORS Application object

Configuration object: ORS Application object

Default value: None

Valid values: Determined by OS; 0 indicates that the option will not be set

Value changes: After restart

This option sets the `S0_SNDBUF` option (send buffer size) for the socket used by CURL. Option will be set only if the new value is bigger than current the socket buffer size.

fail-session-if-not-cached

Beginning with version 8.1.401.02, you can also now configure ORS to fail session creation when the size of an SCXML strategy exceeds the compiled cache size. The following option is introduced to facilitate this feature:

Option section: `scxml` in an ORS Application object

Configuration object: ORS Application object

Default value: `false`

Valid values: `true`, `false`

Value changes: After restart

When this option is set to `true`, session creation fails if an SCXML document cannot be placed into the compiled document cache.

log Section

This section describes log options specific to ORS

x-server-trace-level

Option section: `log`

Configuration object: ORS Application object

Default value: `0`

Valid values: `0` - `3`

Value changes: as soon as committed to Configuration Server

This option specifies the level of tracing to be enabled for the ORS.

For example: `log/x-server-trace-level = 2`

See [log-trace-segments](#) for information on segmenting log output.

x-server-gcti-trace-level

Option section: `log`

Configuration object: ORS Application object

Default value: `0`

Valid values: `0` - `3`

Value changes: as soon as committed to Configuration Server

This option specifies the level of GCTI tracing to be enabled for the ORS. It controls how much detail should be in the logs for GCTI-related events, such as those from T-Server.

For example: `log/x-server-gcti-trace-level = 2`

`x-server-config-trace-level`

Option section: `log`

Configuration object: `ORS Application object`

Default value: `0`

Valid values: `0 - 3`

Value changes: as soon as committed to Configuration Server

This option specifies the level of configuration tracing to be enabled for the ORS. It controls how much detail should be in the logs for Configuration-related events, such as reading from Configuration Server and reacting to dynamic changes.

For example: `log/x-server-config-trace-level = 2`

`x-print-attached-data`

Option section: `log`

Configuration object: `ORS Application object`

Default value: `0`

Valid values: `0, 1`

Value changes: as soon as committed to Configuration Server

This option specifies whether or not attached data should be formatted and printed in the logs.

`0—Suppress printing attached data`

`1—Format and print attached data`

For example: `log/x-print-attached-data = 1`

`x-print-treatment-attr`

Option section: `log`

Configuration object: `ORS Application object`

Default value: `1`

Valid values: `0, 1`

Value changes: Immediately

Beginning with release 8.1.400.93, this option enhances hiding of sensitive data in TEvent attributes. The option is used to specify if the T-lib attributes, `AttributeCollectedDigits`, `AttributeLastDigit`, `AttributeDTMFDigits`, and `AttributeTreatmentParms`, are printed in the log.

- 1 - enables printing treatment attributes to the log.
- 0 - Suppresses printing treatment attributes to the log.

log-filter-data Section

Starting with ORS Release 8.1.400.30, ORS can now selectively hide sensitive data in logs when the printing of such sensitive data is not explicitly requested by an SCXML strategy. A new option supports the hiding of sensitive data in ORSURS request and Web request logging.

`ors-regex-<name>[;<filter-type>]`

For more information on this option, see [hiding sensitive data](#).

mcr Section

This section describes mcr section options.

om-memory-optimization

Option section: `mcr`

Configuration object: `ORS Application object`

Default value: `true`

Valid values: `true` or `false`

Value changes: take effect immediately.

This option specifies whether memory optimization will be in effect for the current ORS. When the value is set to `true`, ORS will remove passive multi-media interactions from memory cache.

For example: `mcr/om-memory-optimization = true`

om-max-in-memory

Option section: `mcr`

Configuration object: `ORS Application object`

Default value: `50`

Valid values: `1` to `2000`

Value changes: take effect immediately.

This option specifies a maximum number of interactions to store in memory cache before interactions

will begin to be removed from the cache (oldest first), when `om-memory-optimization` is set to true. The value is in thousands, so that the default value of 100 represents 100,000 interactions that are to be stored in memory cache. The maximum value for this option is 2000, which represent 2,000,000 interactions.

For example: `mcr/om-max-in-memory = 100`

Note: For high volume loading, Genesys recommends the default value of 100.

`om-delete-from-memory`

Option section: `mcr`

Configuration object: `ORS Application object`

Default value: 1

Valid values: 1 to 2000000

Value changes: take effect immediately.

This option specifies how many interactions should be deleted when the value of `om-max-in-memory` has been reached.

For example: `mcr/om-delete-from-memory = 1`

performance-alarms Section

This section describes `performance-alarms` section options.

`alarm-name`

The name of this option is the alarm name.

Option section: `performance-alarms`

Configuration object: `ORS Application object`

Default value: None

Valid values: Specify an alarm definition string.

Value changes: Immediately upon notification.

Starting with Release 8.1.400.21, ORS supports Performance Monitoring. You can use a set of performance-related counters and configure conditions that will trigger Management Layer alarms. This option defines the alarm name and the performance monitoring parameters. For more information, see [Performance Monitoring](#).

alarm-check-interval

Option section: performance-alarms

Configuration object: ORS Application object

Default value: 60

Valid values: Any non-negative integer from 1 to 600

Value changes: Immediately upon notification.

datastream-update-interval

Configuration object: Orchestration Server Application

Option section: performance-alarms

Default value: 1

Valid values: A number between 1 and 120

Value changes: Immediately upon notification

Introduced in Release 8.1.400.30 to support [displaying ORS Performance Data in a RTME client](#) such as Pulse. Defines the time interval in seconds between performance counters submissions from ORS to Stat Servers.

Starting with Release 8.1.400.21, ORS supports Performance Monitoring. You can use a set of performance-related counters and configure conditions that will trigger Management Layer alarms. This option defines the time interval, in seconds, to use when checking for alarm conditions. For more information, see [Performance Monitoring](#).

elasticsearch Section

See [Elasticsearch Connector](#) for more information.

ors-es-bulk-max-size

Option section: elasticsearch

Configuration object: ORS Application object

Default value: 10000

Valid values: Integer value in range from 1000 to 100000000

Value changes: Take effect immediately upon notification.

Introduced in ORS 8.1.400.40.

The word "bulk" refers to the possibility of having ORS accumulate a configurable amount reporting data and then sending the data to Elasticsearch instead of sending one reporting request at a time.

Used for both Elasticsearch session and performance reporting. This option specifies the maximum

number of actions inside a single bulk request. If reporting is enabled, ORS stores all requests in its internal bulk requests cache. This option can be set to control cache overgrowth. If the maximum number of actions in the cache is reached, ORS removes the oldest actions and inserts the newest.

Note: Genesys recommends clustering Elasticsearch nodes with several master nodes to prevent a potential loss of reporting data should an extended node disconnect occur (unless all available Master nodes in an Elasticsearch cluster are excluded).

ors-es-bulk-write-period

Option section: `elasticsearch`

Configuration object: ORS Application object

Default value: 5 seconds

Valid values: An Integer value in the range of 5 to 120 seconds.

Value changes: Take effect immediately upon notification.

Introduced in ORS 8.1.400.40.

Used for both Elasticsearch session and performance reporting.

This option specifies the maximum time interval between the sending of two bulk requests to Elasticsearch. The bulk mechanism is based on an internal buffer that stores all requests that should be sent to Elasticsearch. In cases where there are no buffered requests, the actual time between two bulk requests could exceed the `ors-es-bulk-write-period`. However, when there is a steady stream of the requests to be handled by single bulk operation, the actual time between two bulk requests will be equal to the time defined by this option.

ors-es-node-exclude-timeout

Option section: `elasticsearch`

Configuration object: ORS Application object

Default value: 600 seconds

Valid values: Integer value in range from 5 to 86400 seconds.

Valid changes: Take effect immediately upon notification.

Introduced in ORS 8.1.400.40.

After successfully connecting to an Elasticsearch node, ORS checks the node status. In case of a red status, ORS disconnects from the node and excludes the node from the reconnection procedure for the time period defined by the option `ors-es-node-exclude-timeout`.

ors-es-node-info-report

Option section: `elasticsearch`

Configuration object: ORS Application object

Default value: false

Valid values: true, false

Value changes: Take effect immediately upon notification.

Introduced in ORS 8.1.400.40.

This option specifies if node reporting via the Elasticsearch engine is enabled. If, upon ORS startup, `ors-es-node-info-report` has a value of false, and the value is subsequently changed to true, the report information will not be sent.

`ors-es-nodes`

Option section: `elasticsearch`

Configuration object: ORS Application object

Default value: <empty string>

Valid values: String containing a semicolon-separated list of `http(s)://host:port` of ElasticSearch servers.

Value changes: Take effect after restart.

Introduced in ORS 8.1.400.40.

This option is used for connectivity to Elasticsearch. It specifies the addresses of Elasticsearch nodes. Configure each ORS to work with an Elasticsearch node. For valid values:

- If the node is specified as `http://<ES host>:<port>`, then a non-encrypted connection will be used for communication with Elasticsearch.
- If the node is specified as `https://<ES host>:<port>`, then an encrypted connection will be used for communication with Elasticsearch.

`ors-es-perfsnapshot-report`

Option section: `elasticsearch`

Configuration object: ORS Application object

Default value: false

Valid values: true, false

Value changes: Take effect immediately upon notification.

Introduced in ORS 8.1.400.40.

This option specifies if performance reporting is enabled via Elasticsearch.

- If set to true, performance reporting via the Elasticsearch is enabled.
- If set to false, performance reporting via Elasticsearch is disabled.

ors-es-reconnect-timeout

Option section: elasticsearch

Configuration object: ORS Application object

Default value: 5 seconds

Valid value: 1 to 20 seconds.

Valid changes: Take effect immediately upon notification.

Introduced in ORS 8.1.400.40.

This option specifies the timeout period that ORS uses when trying to reconnect to an Elasticsearch node. If a connection to Elasticsearch is lost, then ORS will try to reconnect to the next Elasticsearch node in the ors-es-nodes list.

ors-es-session-report

Option section: elasticsearch

Configuration object: ORS Application object

Default value: false

Valid values: true, false

Value changes: Take effect immediately upon notification.

Introduced in ORS 8.1.400.40.

This option specifies if Elasticsearch session reporting is enabled or disabled. When set to true, session reporting is enabled. When set to false, session reporting is disabled.

If an ors-es-session-report value is changed from false to true in the middle of a session, ORS will send an update request for a non-existing session document. Elasticsearch will respond with a "document missing" exception, which ORS will ignore.

password

Option section: elasticsearch

Configuration object: ORS Application object

Default value: <empty string>

Valid value: String value

Valid changes: Take effect immediately upon notification.

Introduced in ORS 8.1.400.40.

This option specifies the password that should be used for basic authentication on the Elasticsearch

server.

username

Option section: `elasticsearch`

Configuration object: ORS Application object

Default value: `<empty string>`

Valid value: String value

Valid changes: Take effect immediately upon notification.

Introduced in ORS 8.1.400.40.

This option specifies the username that should be used for basic authentication on the Elasticsearch server.

scxml-doc-request-headers Section

Starting with release 8.1.400.72, this new section is used for defining arbitrary headers for the HTTP requests generated by the SCXML engine to retrieve SCXML strategy documents. Users can define additional headers in this section using the format specified below:

`<header name>`

Option section: `scxml-doc-request-headers`

Configuration object: ORS Application object or annex of Enhanced Routing Script object (if the same header is configured in both objects, the header value in the Enhanced Routing Script object takes precedence).

Default value: None

Valid values: `<header value>`

Value changes: Immediately

Important

The option name is the `<header name>` and must not contain a ':' symbol. You cannot define multiple headers with the same name. ORS does not validate the consistency of the additional header definitions; they are added to the HTTP requests as is.

gts Section

Starting with release 8.1.400.73, the following two new options have been added to support configurable DN registrations:

dn-reg-chunk-size

Option section: `gts`

Configuration object: `ORS Application` object or `Switch` object (if both objects are configured, the `Switch` object takes precedence)

Default value: 500

Valid values: 1 to 2147483647

Value changes: On next registration attempt

This option is used to define the maximum number of consecutive Register Address requests to one T-Server.

dn-reg-chunk-timeout

Option section: `gts`

Configuration object: `ORS Application` object or `Switch` object (if both objects are configured, the `Switch` object takes precedence)

Default value: 1

Valid values: 1 to 2147483647

Value changes: On next registration attempt

This option is used to define the timeout in seconds between chains of consecutive Register Address requests.

elasticsearch-request-headers Section

Starting with release 8.1.400.89, this new section is introduced to define arbitrary headers for HTTP requests to Elasticsearch (excluding those created via the `<session:fetch>` action).

Users can define the additional headers as shown below:

`<header name>`

Option section: `elasticsearch-request-headers`

Configuration object: `ORS Application` object

Default value: None

Valid values: <header value>

Value changes: Immediately

Important

- The <header name> option must not contain a : symbol.
- You cannot define multiple headers with the same name.
- ORS does not validate the consistency of the additional header definitions; they are added to the HTTP requests as is.

If the arbitrary header has the same name as a header that ORS adds by itself (for example, Connection or User-Agent), the arbitrary value takes precedence except for the following headers:

- X-Opaque-Id
- Authorization (if Elasticsearch/username configured)
- Content-Type
- Content-Length