



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

4/10/2025

Contents

- 1 Application Level Options
 - 1.1 orchestration Section
 - 1.2 persistence Section
 - 1.3 scxml Section
 - 1.4 log Section
 - 1.5 mcr Section

Application Level Options

orchestration Section

This section describes orchestration section options.

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`

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.

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.

`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.

log-trace-segments

Option section: `orchestration`

Default value: `all/All`.

Valid values: Any combination of comma-separated segment names, case sensitive: `ORSInternal`, `ORSInit`, `ORSSynch`, `Cluster`, `ItxLink`, `ItxManager`, `CallMonitor`, `ORSURS`, `Persistence`, `Schedule`, `SessionManager`, `ThreadSync`, `ScxmlCommon`, `ScxmlLog`, `ScxmlMetric`, `ScxmlQOS`, `ORSScxml`, `ScxmlConfig`, `ScxmlIxnAction`, `ScxmlOrsAction`, `DFM`,

ScxmlEvent, FMClassification, FMDialog, FMInteraction, FMQueue, FMResource, FMSession, FMStat, FMWeb, ScxmlProperty, ScxmlLIO

Valid changes: Take effect immediately

This option, introduced in 8.1.300.52, provides more precise control of debug log out by allowing you to define the specific type of debug messages that will be printed into the log files. When set to all or ALL (the default), all debug messages are printed with the log level defined by the log/x-server-trace-level option. To control the log levels for each segment, use syntax:

<segment_name>:<log_level>. If <log_level> is not specified or invalid, ORS applies the default value (defined in x-server-trace-level). If an unknown segment is specified, ORS ignores it without generating an error.

The debug segment header {<segment_name>:<log_level>} is now added right after the timestamp:

```
10:31:02.249 {ScxmlMetric:3} METRIC <event_queued
sid='ML06K6519D5UPAGSHNE41VJBSS00000M' name='interaction.added' type='external'
thread='15800' />
```

Examples:

SETTING	DESCRIPTION
all, ORSInternal:0, ORSURS:0	Everything is enabled except ORSInternal and ORSURS
ScxmlLIO,ThreadSync	Only ScxmlLIO and ThreadSync are enabled
all,ORSURS	Everything is enabled (the same as all)
ORSURS:0	Everything is disabled (the same as empty)
CallMonitor,all:0	Everything is disabled (the same as empty)
all,ORSURS:1	Everything is enabled with the default trace level, but the trace level for the ORSURS segment will be changed to 1
(empty)	Everything is disabled (the same as all:0, or x-server-trace-level=0)
all:3	The same as all with x-server-trace-level=3

Initial Setup:

Run ORS 8.1.300.53 or higher with full debug logging enabled for at least one session processed. Then look at the log messages you need – they all contain a prefix with the segment name and message level, so you will immediately know to configure that particular segment in log-trace-segments. For example, if you determine that you need only messages from the ScxmlMetric segment with level 1 or 2, then set that option as ScxmlMetric:2. If then you determine that you also need messages from the Persistence segment with level 1 only, add it to that option ScxmlMetric:2, Persistence:1, and so on.

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 `sxml-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-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.

Application Level Options

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-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 5000

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.

For example: `orchestration/mcr-pull-limit = 5000`

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`

`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.

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 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.

Note: The ORS `switch-multi-links-enabled` configuration option and the `switch-multi-links-enabled` configuration option specified on the load sharing switch object must both be enabled to support this functionality.

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+.

`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, introduced in ORS Release 8.1.300.45, 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`.

persistence Section

This section describes persistence section options.

`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: No default value (empty string).

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 `ORSdddddd`, where `dddddd` 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`

Application Level Options

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: `100000000`

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`

scxml Section

This section describes scxml section options.

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.

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`

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-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 ORS 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-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 time in seconds for how long a fetched SCXML application is cached. When the same SCXML application is to be used for the new session within the configured timeout, the cached version of the document will be used instead of fetching it from the application server.

To disable fetching of SCXML applications from cache, set options `http-max-age` and `http-max-stale` to `0`.

If the URL of the SCXML application contains parameters that are unique for each invocation, then the application will be fetched each time regardless of option value.

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-redirections

Option section: `scxml`

Configuration object: `ORS Application objectt`

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-redirections = 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`

`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: `No value`

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.

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-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-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: Immediately

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-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.

max-state-entry-count

Option section: `scxml`

Configuration object: ORS Application object

Default value: 100

Valid values: any integer (no maximum)

Value changes: Immediately

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

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`

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`

Also see [log-trace-segments](#).

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`

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: 100

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.

Application Level Options

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`