Orchestration Server Deployment Guide

Direct Statistic Subscription

4/3/2025

Contents

Direct Statistic Subscription

Starting with Release 8.1.400.17, an architectural change streamlines the way that Orchestration Server obtains statistical data when executing routing strategies. Instead of requesting statistics from Stat Server by going through Universal Routing Server (URS), ORS can now request data directly from Stat Server.

This capability is only applicable for the `statistic.sData`, `statistic.getAvgData`, `statistic.getMinData`, `statistic.getMaxData` functions described in the Orchestration Server Developer's Guide and the `statistic.subscribe` action when used in strategies. It is not related to any other ORS capabilities that indirectly use statistical data (for example, `queue.submit` action). There are no changes to the ORS `statistic.sData`, `statistic.getAvgData`, `statistic.getMinData`, and `statistic.getMaxData` functions and the `statistic:subscribe` action so no changes are required in existing SCXML applications.

Note: This feature does not completely replace URS for obtaining statistic information since there are statistics calculated by URS itself. Specifically, the following statistics must be requested from URS only: `RStatCallsInQueue` and `StatAgentLoading`.

Configuration Summary

How to configure the ORS Stat Server Direct Access feature is summarized in the table below. Each step is then detailed in the sections that follow.

Objective	Key Procedure and Action
Configure the connection to Stat Servers in the ORS Application	Under ORS Application > **Connections** Add/configure each Stat Server that ORS could query for statistics
Configure Transaction Objects of Statistic Type	Under the Tenants > Routing/eServices > Transactions directory > Transaction objects of Statistic Type > orchestration section Set the **statserver-source** option to **True**
Configure the default Stat Server in the ORS Application	Under ORS Application > orchestration section Set the **def-statserver-name** option to <**name_of_Stat_Server_application**>
Configure the default object type in the ORS Application	Under ORS Application > orchestration section Set the **def-stat-object-type** option to <**name_of_default_object_type**>

Configure Connection to Stat Server

In the ORS `Application` object, add/configure each Stat Server that ORS could query for statistics.

- In `Connections`, enter the name of the Stat Server `Application`.
- If needed, configure additional capabilities for connection to Stat Server (ADDP protocol, security, and so on). For more information, see the Framework 8.1 Configuration Manager Help, topic Applications:Connections General Tab. All capabilities, provided by Management Framework for connection to server are supported.

Configure Transaction Objects of Type Statistic

In the Configuration Database, `Transaction` objects of `Statistic` Type contain statistic definitions. For each Tenant, define the objects of type `Statistic` that should use ORS direct Stat Server access by configuring the `orchestration/statserver-source` option with a value of `True`.

statserver-source

Object: `Transaction` of `Statistic` type

Option section: `orchestration`

Default value: `False`

Valid values: `False`, `True`

Value changes: Upon ORS restart

This option defines the source for the statistical data in the `Statistic` configuration. If this option is set to `True`, then Orchestration Server direct access to Stat Server is used for this statistic. If the option is `False` (or the option is not configured, Orchestration Server will obtain the statistic data by requesting it from Universal Routing Server.

Configure Default Stat Server

Use the `def-statserver-name option` to specify the default Stat Server. If name of `StatServer` is not explicitly specified in the object name (see description of the `._genesys.statistic.sData` function in the Orchestration Server Developer's Guide), ORS will use default Stat Server for statistic subscription.

def-statserver-name

Object: Name of an object 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 the following order:

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

Configure Default Object Type

Use this option to configure the default object for which statistics are being collected.

def-stat-object-type

Object: ORS Application

Option section: `orchestration`

Default value: `agent`.

Valid values:

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

Value changes: Take effect immediately

This value will be used by default in case an object type is skipped in the object description in the strategy.

Same Statistic for Multiple Tenants

In the case where a `Transaction` object of the same Statistic belongs to more than one Tenant, ORS selects the proper statistic definition for the action/function called from SCXML using option `def-statserver-name`.

- On startup, ORS reads all `Transaction` objects of `Statistic` type that belong to the Tenants associated with the ORS Application. In Genesys Administrator, the Tenants are listed in Configuration > Server Info > Tenants.
- If ORS is executing an SCXML application without a Tenant context, the statistical value cannot be obtained.

Disconnects/Reconnects

Since Stat Server supports only warm-standby High Availability (HA), there is no difference between scenarios with a standalone Stat Server and a Stat Server HA pair. In both cases, ORS will drop the statistic subscription and create it from scratch upon reconnect. While the connection to Stat Server is not yet established upon reconnect, the result of the `sData` function will be undefined, if ORS is requesting the statistic from that Stat Server. This result is applicable for a Stat Server HA pair as well.

Targeting a Stat Server

As described above, ORS supports simultaneous connections to many Stat Servers, which must be listed in the Connections tab of the ORS `Application` object. Any of these servers could used to receive statistical data. You can also explicitly and implicitly subscribe to a Stat Server for exact statistical data.

Explicitly Targeting a Stat Server

You can explicitly target a Stat Server in the statistic action/function call as part of the object name. In this case, the component parses as in following code snippet, which contains the target Stat Server and statistic name.

```
<statistic:subscribe object="'SipGr_2@StatSrvName.GA'"
statistic="‘StatSrvStatName'"/>
```

If the Statistic name pointing to a specific Stat Server should be used, then ORS uses the cache of the connected Stat Servers to find the corresponding server. If there is no such Stat Server with that name in the connection cache, then ORS will not subscribe for this statistic and prints out the following error message into the log:

```
TFMStatSubscription: Server StatSrvName is not in the connection list
```

If the server with name `StatSrvName` is available, then ORS communicates with it and subscribes to statistic with parameters, configured in Transaction/statistic object, named as `StatSrvStatName`.

Implicitly Targeting a Stat Server

Implicitly targeting a Stat Server requires that one Stat Server from the list of connected ones is set as default. To configure a default Stat Server, option `orchestration/def-statserver-name` must point to its name. If this option does not contain the server name or contains an incorrect server name (i.e., the name is not from the list of connected Stat Servers in the ORS Connection list), then ORS uses the default value of the `def-staserver-name` option. An incorrect value of this option is logged via Standard-level message ID=3010. - If the name server of Stat Server is defined and it is the correct name, then ORS can communicate with the targeted Stat Server.

Debug-level logging

The log file sample below contains typical log output for the initial subscription phase: request to open statistic, event 22 (`SEventStatOpened`), event 2 (`SEventInfo`) – actual statistic value.

```
13:43:21.897 {FMStat:2} ExecuteSubscribe: Object 'AG_20_AlexK.GA', statistic
'CurrCallsInbound'
13:43:21.897 {FMStat:3} ORSStatServer::SubscribeToStatistic: <<
        RequestID        '1'
        StatServer name        'Stat_Server_812'
        tenant        'sip80'
        statistic        'CurrCallsInbound'
        target        'AG_20_AlexK.GA'
        interval        0
 <<
13:43:21.902 {FMStat:3} ORSStS::HandleREvent: <<
        RequestID        '1'
        event        22
        value        0
 <<
13:43:21.902 {FMStat:3} ORSStS::HandleREvent: <<
        RequestID        '1'
        event        2
        value        0
 <<
```