



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Orchestration Server Developer's Guide

Statistic Interface

Contents

- 1 Statistic Interface
 - 1.1 Functions
 - 1.2 Action Elements
 - 1.3 Events

Statistic Interface

This interface provides statistical information to the Orchestration logic.

Functions

`_genesys.statistic.sData`

This function returns the value of a statistic for a specified object. For example, this function can be used to get the number of interactions waiting in a queue (that is, number of callers ahead of this caller), so that it can be announced to the caller via an IVR application while waiting for a target.

value `_genesys.statistic.sData(object, statistic)`

Before you can query a statistic using `_genesys.statistic.sData` within your SCXML file, you should subscribe to the statistic. Please see [subscribe](#) for more details.

However, the following list of predefined statistics can be used without subscribing to it within your SCXML file:

- `StatTimeInReadyState`
- `StatAgentsAvailable`
- `StatAgentsTotal`
- `StatAgentsBusy`
- `StatCallsAnswered`
- `StatCallsCompleted`
- `StatExpectedWaitingTime`
- `StatLoadBalance`
- `StatAgentsInQueueLogin`
- `StatAgentsInQueueReady`

Parameters:

- **JavaScript object.** Starting with ORS 8.1.400.45, the `_genesys.statistic.sData` function allows you to directly define statistic parameters in the SCXML routing strategy code via a JavaScript object passed as an input parameter. For information on this enhancement, see [Direct Statistic Definition](#) in the *Orchestration 8.1.4 Deployment Guide*.
- **object:** STRING which can be a variable or a constant - This is the name of the object for which the statistic value is requested. The format of this parameter value must use the target formats (for details see the [Queue Interface Target Formats](#) section), but there are some exceptions based on the type of statistic (that is, the value of the statistic parameter) being requested. The following describes those exceptions:

- If the statistic parameter value is *CallsDistributed*, *CallsAnswered*, *DistributedPercentage*, *DistributedWaitingTime*, *NotDistributedPercentage*, or *NotDistributedWaitingTime*, then the object parameter value can only be one of the following:
 - **"R"** - This represents the Queue functional module system as a whole (router).
 - **"RP"** - This is the route point associated with this session (that is for example, `_genesys.ixn.interactions[ixnid].voice.dnis`)
 - In all other cases, it is the name of a virtual queue.
- **statistic:** STRING which can be a variable or a constant - This is the name of the statistic being requested. A value of zero (0) will returned if:
 - The connection to the functional module's statistical system (Stat Server) is not available.
 - If the specified statistic is neither on the list of predefined statistics nor defined in the strategy.
 - If the string defining the object is not of a valid format or the type of object does not support that requested statistic.

Returns: value: NUMBER (FLOAT) - This is the current value of the statistic.

The following is an example of using a predefined statistic without subscribing:

```
<scxml version="1.0" xmlns="http://www.w3.org/2005/07/scxml"
  xmlns:queue="http://www.genesyslab.com/modules/queue"
  xmlns:statistic="http://www.genesyslab.com/modules/statistic"
  initial="initial" _persist="false">
  <datamodel>
    <data ID="reqid"/>
  </datamodel>
  <state id="initial" >
    <transition event="interaction.added" target="check" />
  </state>
  <state id="check">
    <transition
cond="_genesys.statistic.sData('SipGr_1@StatServer.GA','StatAgentsAvailable')==1"
target="routing"/>
  </state>

  <state id="routing">
    <onentry>
      <queue:submit requestid="_data.reqid" queue="'802_SipSwitch@.Q'" priority="5"
timeout="20">
        <queue:targets type="dn">
          <queue:target name="'702'" />
        </queue:targets>
      </queue:submit>
    </onentry>

    <transition event="queue.submit.done" target="exit">
      <log expr="'Queue Submit DONE'"/>
      <log expr="'SDATA for SipGr_1 =
'+_genesys.statistic.sData('SipGr_1@StatServer.GA','StatAgentsAvailable')"/>
      <log expr="_event.data.targetselected"/>
    </transition>
    <transition event="error.queue.submit" target="error">
      <log expr="uneval( _event )" />
    </transition>
  </state>
  <final id="exit" />
  <final id="error" />
```

```
</scxml>
```

This is an example of a custom statistic which requires subscription:

```
<scxml version="1.0" xmlns="http://www.w3.org/2005/07/scxml"
  xmlns:queue="http://www.genesyslab.com/modules/queue"
  xmlns:ixn="http://www.genesyslab.com/modules/interaction"
  xmlns:statistic="http://www.genesyslab.com/modules/statistic"
  initial="initial" _persist="false">
  <datamodel>
    <data ID="reqid"/>
    <data ID="ixnid"/>
    <data ID="q1"/>
    <data ID="q2"/>
    <data ID="q3"/>
    <data ID="time_delay" expr="'3s'" />
  </datamodel>
  <state id="initial" >
    <transition event="interaction.added" target="subscribe" />
  </state>

  <state id="subscribe">
    <onentry>
      <log expr="'===== Inside Subscribe ====='"/>
      <statistic:subscribe object="'801_SipSwitch@StatServer.Q'" statistic="'AvgWaitingTime'"
interval="0"/>
      <statistic:subscribe object="'802_SipSwitch@StatServer.Q'" statistic="'AvgWaitingTime'"
interval="0"/>
      <statistic:subscribe object="'803_SipSwitch@StatServer.Q'" statistic="'AvgWaitingTime'"
interval="0"/>
    </onentry>

    <transition event="statistic.subscribe.done" target="delay"/>
    <transition event="error.statistic.subscribe" target="error"/>
  </state>

  <state id="delay">
    <onentry>
      <log expr="'===== Inside Delay ====='"/>
      <send event="'SynchroEvent'" delay="_data.time_delay"/>
    </onentry>
    <transition event="SynchroEvent" target="check" />
  </state>

  <state id="check">
    <onentry>
      <log expr="'Script Start===== '"/>
      <script>
        _data.q1 = _genesys.statistic.sData('801_SipSwitch@StatServer.Q','AvgWaitingTime');
        _data.q2 = _genesys.statistic.sData('802_SipSwitch@StatServer.Q','AvgWaitingTime');
        _data.q3 = _genesys.statistic.sData('803_SipSwitch@StatServer.Q','AvgWaitingTime');
        __Log('AvgWaitingTime for q1 = '+_data.q1);
        __Log('AvgWaitingTime for q2 = '+_data.q2);
        __Log('AvgWaitingTime for q3 = '+_data.q3);
      </script>
      <log expr="'Script End===== '"/>
    </onentry>
    <transition
cond="_genesys.statistic.sData('803_SipSwitch@StatServer.Q','AvgWaitingTime') == 0"
target="routing" />
  </state>

  <state id="routing">
```

```
<onentry>
  <queue:submit requestid="_data.reqid" queue="'802_SipSwitch@.Q'" priority="5"
timeout="20">
    <queue:targets type="dn">
      <queue:target name="'702'" />
    </queue:targets>
  </queue:submit>
</onentry>

<transition event="queue.submit.done" target="exit">
  <log expr="'Queue Submit DONE'"/>
  <log expr="_event.data.targetselected"/>
</transition>
<transition event="error.queue.submit" target="error">
  <log expr="uneval( _event )" />
</transition>
</state>
<final id="exit" />
<final id="error" />
</scxml>
```

`_genesys.statistic.getAvgData`

This function calculates the specified statistic for all listed targets and returns the average value of this statistic.

```
value _genesys.FMname.getAvgData(objects, statistic)
```

Parameters:

- **objects:** STRING which can be a variable or a constant - This parameter is the list of comma-separated objects (targets, in the case of target selection functionality) which this calculation is to be done against.
- **statistic:** STRING which can be a variable or a constant - This parameter defines the statistic that is to be used in this calculation.

Returns: value: NUMBER (FLOAT) - The result of the function is the average value for the requested statistic, based on the list of objects and their statistical values.

`_genesys.statistic.getMinData`

This function calculates the specified statistic for all listed targets and returns the minimum value of this statistic.

```
value _genesys.FMname.getMinData(objects, statistic)
```

Parameters:

- **objects:** STRING which can be a variable or a constant - This parameter is the list of comma-separated objects (targets, in the case of target selection functionality) which this calculation is to be done against.
- **statistic:** STRING which can be a variable or a constant - This parameter defines the statistic that is to be used in this calculation.

Returns: value: NUMBER (FLOAT) - The result of the function is the minimum value for the requested

statistic, based on the list of objects and their statistical values.

`_genesys.statistic.getMaxData`

This function calculates the specified statistic for all listed targets and returns the maximum value of this statistic.

```
value _genesys.FMname.getMaxData(objects, statistic)
```

Parameters:

- **objects:** STRING which can be a variable or a constant - This parameter is the list of comma-separated objects (targets, in the case of target selection functionality) which this calculation is to be done against.
- **statistic:** STRING which can be a variable or a constant - This parameter defines the statistic that is to be used in this calculation.

Returns: value: NUMBER (FLOAT) - The result of the function is the maximum value for the requested statistic, based on the list of objects and their statistical values.

Action Elements

The following are the statistic-specific actions.

`<subscribe>`

This action allows an application to dynamically subscribe to a particular statistic and object pair for this session.

Attribute Details

Name	Required	Type	Default Value	Valid Values	Description
requestid	false	location expression	none	Any valid location expression	This is the location for the request ID that is returned as part of this request. Any data model expression evaluating to a data model location. See SCXML Location Expressions for details. The location's value will be set to an internally generated unique string identifier to

Name	Required	Type	Default Value	Valid Values	Description
					be associated with the action being sent. If this attribute is not specified, the event identifier is dropped. This identifier can be tested by the completion event handler to distinguish among several outstanding requests. If this attribute is not specified, the identifier can be acquired from the completion event. Every request must receive a unique identifier.
object	true	value expression	none	Any value expression that returns a valid string that follows the target formats (for details see the Queue interface Target Formats section)	A value expression which returns the name of the object name associated with this subscription request. The following is the set of valid object types: • Agent • Agent Group (virtual or real) • Campaign • Campaign Group • Destination Label • Interaction Queue • Place • Place Group • Queue (virtual or real)

Name	Required	Type	Default Value	Valid Values	Description
					- Queue Group - Routing Point (virtual and real) See SCXML Legal Data Values and Value Expressions for details.
statistic	true	value expression	none	Any value expression that returns a valid string that represents a valid statistic	A value expression which returns the name of the statistic associated with this subscription request. Any statistic name can be specified except for the following: - CallsDistributed - CallsAnswered - DistributedPercentage - DistributedWaitingTime - NotDistributedPercentage - NotDistributedWaitingTime See SCXML Legal Data Values and Value Expressions for details.
interval	false	value expression	30	A value expression which returns an integer that is greater than 5	A value expression which returns an integer that represents the number of seconds to wait for the interval. See SCXML Legal Data Values and Value Expressions for details. The integer returned must be interpreted as a time interval when an updated value of the statistic is sent.

Note: There can only be one active subscription for a given object and statistic pair and the session.

If there is an active subscription for an object and statistic pair and another `<subscribe>` action is invoked, the new request will be rejected with an error event. If the developer wants to change the interval of a given subscription, they will have to `<unsubscribe>` and `<subscribe>` again with the new value.

The following is an example:

```
<state id="do_subscribe">
  <datamodel>
    <data id="reqid"/>
  </datamodel>
  <onentry>
    <statistic:subscribe requestid="_data.reqid" object="'1234.Q'"
      statistic="'InvQWaitTime'"/>
  </onentry>
  <transition event="statistic.subscribe.done" target="statex"/>
  <transition event="error.statistic.subscribe" target="statey"/>
</state>
```

Children

None

Events

The following events can be generated as part of this action:

- `statistic.subscribe.done` - This event is sent when the request has been accepted by the system and the statistic subscription has started for this session.
- `error.statistic.subscribe` - This event is sent when the request has failed for some reason.
- `statistic.update` - This event is sent at the end of each time interval while the subscription is active.

`<unsubscribe>`

This action allows an application to unsubscribe from a particular statistic and object pair for this session.

Attribute Details

Name	Required	Type	Default Value	Valid Values	Description
requestid	false	location expression	none	Any valid location expression	This is the location for the request ID that is returned as part of this request. Any data model expression evaluating to a data model

Name	Required	Type	Default Value	Valid Values	Description
					location. See SCXML Location Expressions for details. The location's value will be set to an internally generated unique string identifier to be associated with the action being sent. If this attribute is not specified, the event identifier is dropped. This identifier can be tested by the completion event handler to distinguish among several outstanding requests. If this attribute is not specified, the identifier can be acquired from the completion event. Every request must receive a unique identifier.
object	true	value expression	none	Any value expression that returns a valid string that follows the target formats (for details see the [[Queue_Interface#Target_formats]] section)	A value expression which returns the name of the object name associated with this subscription request. The following is the set of valid object types: • Agent • Agent Group (virtual or real) • Campaign • Campaign Group • Destination Label • Interaction Queue

Name	Required	Type	Default Value	Valid Values	Description
					- Place - Place Group - Queue (virtual or real) - Queue Group - Routing Point (virtual and real) See SCXML Legal Data Values and Value Expressions for details.
statistic	true	value expression	none	Any value expression that returns a valid string that represents a valid statistic	A value expression which returns the name of the statistic associated with this subscription request. Any statistic name can be specified except for the following: - CallsDistributed - CallsAnswered - DistributedPercentage - DistributedWaitingTime - NotDistributedPercentage - NotDistributedWaitingTime See SCXML Legal Data Values and Value Expressions for details.

The following is an example:

```
<state id="do_unsubscribe">
  <datamodel>
    <data id="reqid"/>
  </datamodel>
  <onentry>
    <statistic:unsubscribe requestid="_data.reqid" object="'1234.Q'"
      statistic="'InVQWaitTime'"/>
  </onentry>
</state>
```

```
<transition event="statistic.unsubscribe.done" target="statex"/>
<transition event="error.statistic.unsubscribe" target="statey"/>
</state>
```

Children

None

Events

The following events can be generated as part of this action:

- `statistic.unsubscribe.done` - This event is sent when the request has been accepted by the system and the statistic subscription has been terminated for this session.
- `error.statistic.unsubscribe` - This event is sent when the request has failed for some reason.

Events

The following are the statistic action result events:

Event	Attributes	Description
statistic.subscribe.done		This event indicates the success of the request and that the subscription has been activated for this session.
	requestid	This is the ID associated with the request.
error.statistic.subscribe		This indicates that an abnormal condition occurred while trying to perform the request. This event will be sent as a result of a timeout of the request as well as due to problems with the request or interaction itself.
	requestid	This is the ID associated with the request.
	error	This is the type of error that occurred: • Request Invalid • Subscription Already Active
	description	This is a more detailed description of the error.

Event	Attributes	Description
statistic.unsubscribe.done		This event indicates the success of the request and that the subscription has been terminated for this session.
	requestid	This is the ID associated with the request.
error.statistic.unsubscribe		This indicates that an abnormal condition occurred while trying to perform the request. This event will be sent as a result of a timeout of the request as well as due to problems with the request or interaction itself.
	requestid	This is the ID associated with the request.
	error	This is the type of error that occurred: • Request Invalid
	description	This is a more detailed description of the error.

The following are the stat asynchronous events:

Event	Attributes	Description
statistic.update		This provides the update value of the statistic.
	object	This is the name of the object that the statistic is associated with. (string)
	statistic	This is the name of statistic. (string)
	value	This is a floating point number which represent the updated value of the statistic that was subscribed to.