



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.

## Composer Help

Target Block

# Target Block

## Contents

- **1 Target Block**
  - 1.1 Name Property
  - 1.2 Block Notes Property
  - 1.3 Activity Property
  - 1.4 Clear Targets Property
  - 1.5 Condition Property
  - 1.6 Cut Off Time Property
  - 1.7 Detach Property
  - 1.8 Detach Timeout Property
  - 1.9 Enable Status Property
  - 1.10 Exceptions Property
  - 1.11 Expected Wait Time Property
  - 1.12 From Property
  - 1.13 Hints Property
  - 1.14 Include Requests from Previous Blocks Property
  - 1.15 Interaction ID Property
  - 1.16 Logging Details Property
  - 1.17 Login Size Property
  - 1.18 ORS Extensions Property
  - 1.19 Log Level Property
  - 1.20 Pass Context Property
  - 1.21 Position in Queue Property
  - 1.22 Priority Property (Target Selection)
  - 1.23 Priority Property (Queue)
  - 1.24 Queue Poll Rate Property
  - 1.25 Request ID Property
  - 1.26 Resource Selected Property
  - 1.27 Route Property

- [1.28 Statistics Property](#)
- [1.29 Statistics Order Property](#)
- [1.30 Targets Property](#)
- [1.31 Target Component Selected Property](#)
- [1.32 Target Deviation From Ideal Agent](#)
- [1.33 Target Object Selected Property](#)
- [1.34 Target Selected Property](#)
- [1.35 Timeout Property](#)
- [1.36 Total Size Property](#)
- [1.37 Type Property](#)
- [1.38 Use Access Code Property](#)
- [1.39 Use Access Code Setter Property](#)
- [1.40 Use Treatments Property](#)
- [1.41 Virtual Queue Property](#)
- [1.42 Virtual Queue Selected Property](#)
- [1.43 Creating Threshold Expressions](#)

Use to route an voice interaction to a target (use [Route Interaction](#) for multimedia interactions). Also use for [percentage routing and conditional routing](#) using the [threshold functions](#).

- See note on [excluding agents](#) before Target blocks.

The Target block has the following properties:

### Name Property

Find this property's details under [Common Properties](#).

### Block Notes Property

Find this property's details under [Common Properties](#).

### Activity Property

Starting with 8.1.410.03, you can specify a Workforce Management [Activity](#) for routing and a related Cut Off Time (see property below) to specify a timeout. Select the variable that contains the WFM Activity.

### Clear Targets Property

Click the down arrow and select true or false (default). If set to true, URS retains the targets listed in the block are after the interaction moves on through the strategy and encounters other Target blocks.

- This option is only applicable if more than one Target block is used along the same path in the strategy. It has nothing to do with how agents specified within the block are evaluated for availability. If no additional Target blocks are included in the path and no agent is available, the interaction is default-routed. If more than one agent is available, URS uses the statistic specified in the Statistics property to determine which agent receives the interaction.
- If Clear Targets = false and the current interaction encounters a Target block later in the strategy, the targets in this block are added to those of the next Target block. If Clear Target = true, the targets are not added to the list of any Target block that the interaction encounters later in the strategy.

For example, assume two Target blocks are used in a strategy with the first Target block configured with three agents and a timeout of 30 seconds and the second Target block configured with four new agents and a timeout of 60 seconds. The Clear Targets setting determines which agents are considered as targets for an interaction.

- Clear Targets = true. In the first Target block, URS waits for 30 seconds for any of the three agents to become available. If none are available before the timeout expires, the strategy resumes. When the second Target block is encountered, URS only considers these four new agents as possible targets during the 60-second timeout. URS does not consider (clears) the first three agents as possible targets.
- Clear Targets = false and none of the agents listed in the first Target block are available. When the interaction encounters the second Target block, URS considers the first three agents and the four new agents as possible targets during the 60-second timeout set in this object. If no more Target blocks follow the second Target block and no agents are available before the timeout expires, the interaction is routed to the error port.

## Condition Property

Find this property's details under [Common Properties](#).

## Cut Off Time Property

Starting with 8.1.430.03, you can specify the timeout number of seconds to wait for retrieving an Workforce Management Activity.

Code generation:

```
<queue:submit interactionid="system.InteractionID" requestid="App_Target_SCXML['requestid']"
route="false" timeout="20" clearontimeout="false">
  <queue:targets>
    <queue:activity cutofftime="10" name="varWFMActivity"/>
  </queue:targets>
</queue:submit>
```

## Detach Property

Controls whether the Orchestration Platform should <detach> an interaction from the current session before routing to reserved targets. When this property is set to true, the interaction is detached from the current session.

**Note:** A Project properties option, **Interaction Detach**, in the Orchestration Options dialog can generate the detach attribute in the <ixn:redirect> tag in the Routing blocks. See [Detaching Interactions from Sessions](#).

## Detach Timeout Property

Use to specify how long to attempt to <detach> if an initial attempt fails with an invalidstate error. Specify the timeout in milliseconds. If set to 0, no further attempt to detach is made. After the timeout, if the <detach> is not successful, no further attempts will be made and the block will attempt to reclaim the interaction back into the current session using <attach>.

## Enable Status Property

Find this property's details under [Common Properties](#).

## Exceptions Property

Find this property's details under [Common Properties](#). Note: Exceptions for Busy treatment blocks should be handled in the Target block to which they are connected and not in the Busy treatment blocks themselves. Busy treatment exceptions are raised as the `error.queue.submit` exception and not as exceptions listed in individual treatment blocks.

## Expected Wait Time Property

To use this optional property, click the down arrow under Value and select a variable to contain the expected wait time that the interaction will be given in the queue. The selected variable will be populated with the respective output value obtained from the operations described next.

Starting with 8.1.4, the Target block and Route Interaction block provide an integrated mechanism to playback the estimated waiting time to customers, including defining the playback repeat interval. To implement this mechanism, the Target and Route Interaction blocks have new properties (variables) to capture the result of `queue.query.done`. There is also a property (Advanced properties) to configure the poll period (1 second by default).

- [Login Size](#)
- [Position in Queue](#)
- [Request ID](#)
- [Total Size](#)
- [Queue Poll Rate](#)

If any of the above five properties are set, the application does a `queue:query` after the `queue.submit.requestid`.

The `queue:query` is repeated every x secs until `queue.submit.done` is received. There is no additional output.

The variables used for the five properties can be used in a Busy Treatment block, such as [Play Application](#). Or you may put the Target block in a parallel leg and use the variables (for example, to speak the estimate wait time) in another leg.



If the call is distributed right away, then the application is presented an `error.queue.query` event (the `queue:query` is processed after the `queue.submit.done`). If nothing is done, that unexpected error will terminate the application. As a result:

- When assigning a variable to any of the above five properties, you will be prompted to automatically add a target-less exception for `error.queue.query`.

- Diagram validation will show a warning if one of the five properties is set and no handler is defined in the Target block (or one of its parents, // or the **Entry block**).

## From Property

A value expression, which returns the address that this interaction is to be redirected from. Set this property to the variable DNIS for voice interactions, or to the variable InteractionID for multimedia interactions. Composer will automatically set this property to DNIS or to InteractionID when the Destination property is set (respectively) to a Target Block or to a Route Interaction block. When the Destination property is not assigned a Block Reference value, you must select the appropriate **From** value.

1. Click under **Value** to display the  button.
2. Click the  button to open the **From** dialog box.
3. Select one of the following:
  - **Literal.** For Value, you can specify:
    - An agent: <agent id>
    - A place: <place id>
    - A DN: <number>
    - An e-mail address: <username>@<host> or \_origin or \_origin.all or \_udata
    - A customer number: <dn number>
    - A target format addresses: <Target DN>

See the [Orchestration Server Documentation Wiki](#) for those literals that apply to multimedia interactions only.

- **Variable.** If the variable contains a string, see Literal above. If the value is a JSON object, Value can refer to:
  - An agent: {agent: <agent id>, type:A}
  - An agent group: {agent: <name>, type:AG}
  - A place: {place: <place id>, type:AP}
  - A place group: {place: <name>, type:PG}
  - A DN: {dn: <number>, type:Q or RP or DN, switch:<switch name>}
  - An interaction queue: {id: <q name>, type:iq }
  - A workbin: {id: <wb name>, type:wb<owner>}
  - A customer number: {dn: <number>}
  - A target format addresses: Resource Object from the **queue.submit.done** event (the Target Block **Resource Selected property**).

See the [Orchestration Server Documentation Wiki](#) for those literals that apply to multimedia interactions only.

- **Configuration Server** to select the from Switch//DN if **connected**.
- **Resource** to select a resource using properties that will form a JSON object.

See the [Orchestration Server Documentation Wiki](#).

4. Click **OK** to close the From dialog box.

## Hints Property

This property is for use by Orchestration Server. Its use is described in various action elements referenced in the [Orchestration Developer's Guide](#).

## Include Requests from Previous Blocks Property

This property controls whether the block should transition if a target from previous Target block is selected even though it may not be specified in the current block. Set it to true for cascaded target lookups. If set to false, the block will wait until a target specified in the current block is selected for routing.

## Interaction ID Property

Set to a meaningful value or keep the default value, which is the system variable InteractionId. Set to a meaningful value or keep the default value, which is the system variable InteractionId. Find more details under [Common Properties](#).

## Logging Details Property

Find this property's details under [Common Properties](#).

## Login Size Property

Select the variable to contain the number of agents logged in that can be targeted for the request (you must have Universal Routing Server 8.1.3+). The selected variable will be populated with the respective output value obtained from the operations described in the [Expected Wait Time](#) property.



## ORS Extensions Property

Starting with 8.1.4, Composer blocks used to build routing applications (with the exception of the Disconnect and EndParallel blocks) add a new **ORS Extensions** property.

## Log Level Property

Find this property's details under **Common Properties**.

## Pass Context Property

This property accepts true/false values. When set to true and Detach is also true:

- URL built with the block name is stored into this interaction's user data (user data key name is '\_composer\_originating\_session') just before detaching the interaction. That URL will be used by the orchestration destination session (the new orchestration session started to handle the interaction after it was redirected to an other routing point) to request the context of the originating session. After the processing for this block is over, the originating session is blocked until the destination session actually reads the context. The context consists of the system and user **variables**.

## Position in Queue Property

Click the down arrow under Value and select the variable to contain the position that the interaction will be given in the queue. The selected variable will be populated with the respective output value obtained from the operations described in the **Expected Wait Time** property.

## Priority Property (Target Selection)

To use this optional property, click the down arrow under Value and select the variable, which contains a value expression returning the priority that the interaction will be given during target selection.

## Priority Property (Queue)

To use this optional property, click the down arrow under Value and select the variable, which contains a value expression returning the priority that the interaction will be given in the queue. For more information, see [<http://www.w3.org/TR/scxml#ValueExpressions> <http://www.w3.org/TR/scxml/#ValueExpressions>].

## Queue Poll Rate Property

Click the **Show Advanced Properties** button to view. Select the variable to contain how frequently the block will poll for latest values and populate selected variables.

## Request ID Property



Set to a meaningful value or keep the default value, which is the system variable system.ANI. The selected variable will be populated with the respective output value obtained from the operations described in the **Expected Wait Time** property.

## Resource Selected Property

Click the down arrow and select the variable that specifies the target resource for the interaction.

## Route Property



Use this property to set the SCXML <queue:submit> route attribute. Also see the Use Access Code property below.

1. Click under **Value** to display the  button.
2. Click the  button to open the **Route** dialog box.
3. Select one of the following:
  - **Variable.** Then for **Value**, select the variable that contains either true (default) or false.
  - **Literal.** Then for **Value**, enter either true or false.

When set to false, the functional module will not attempt to route the associated interaction.

## Statistics Property

Optional. If you wish to route based on the value of a statistic:

1. Click under **Value** to display the  button.
2. Click the  button to open the **Statistic** dialog box.
3. Select one of the following:
  - **Literal.** For Value, you can write in a URS-predefined statistic or a custom statistic

created with [Statistics Builder](#).

- **Variable.**
- **Configuration Server** to select a statistic. In order to select a statistic, you must be [connected to Configuration Server](#) and have set the option to use [URS Predefined statistics](#).

4. Click **OK** when through in the dialog box.

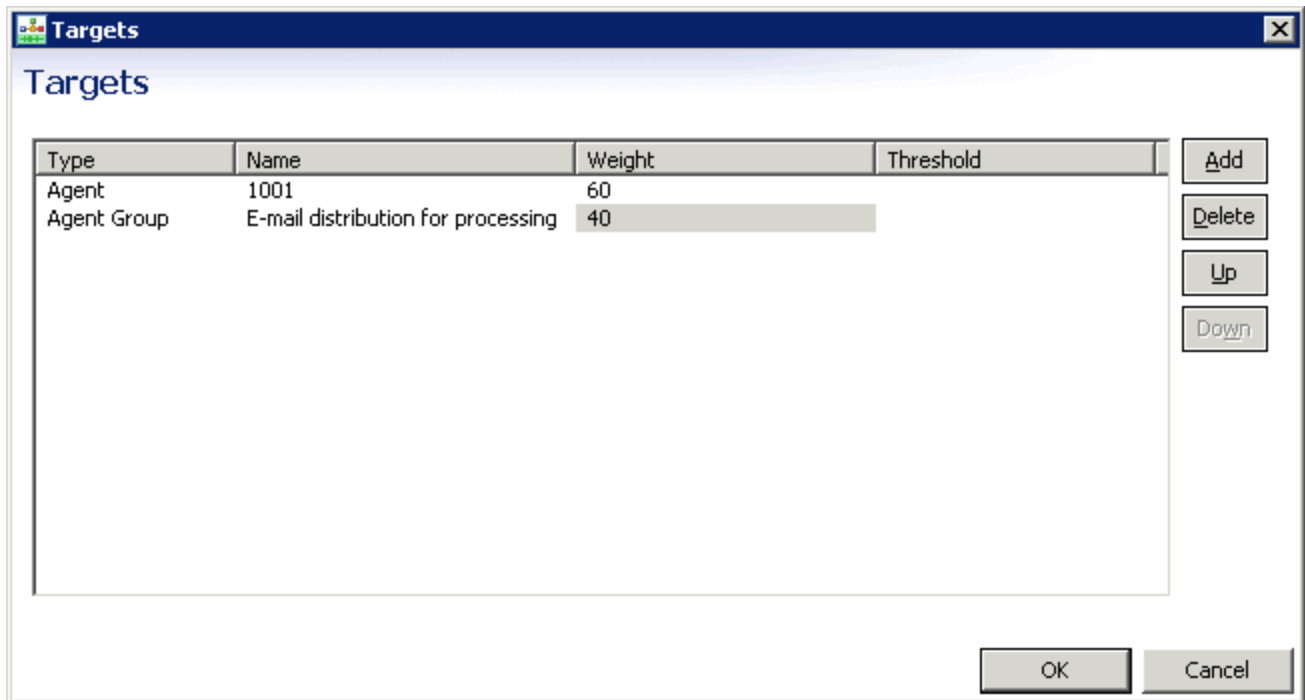
The URS predefined statistics are described in the [Universal Routing 8.1 Reference Manual](#) .

The statistic you select is used by Universal Routing Server to determine which target to route the interaction to if more than one target is available. After defining a complete set of available agents (taking agent capacity rules into consideration, if configured), URS applies the selection criteria specified in the Target block, which can include using the minimum or maximum value of the statistic (see Statistics Order property).

## Statistics Order Property



This property can work with the Statistics property.

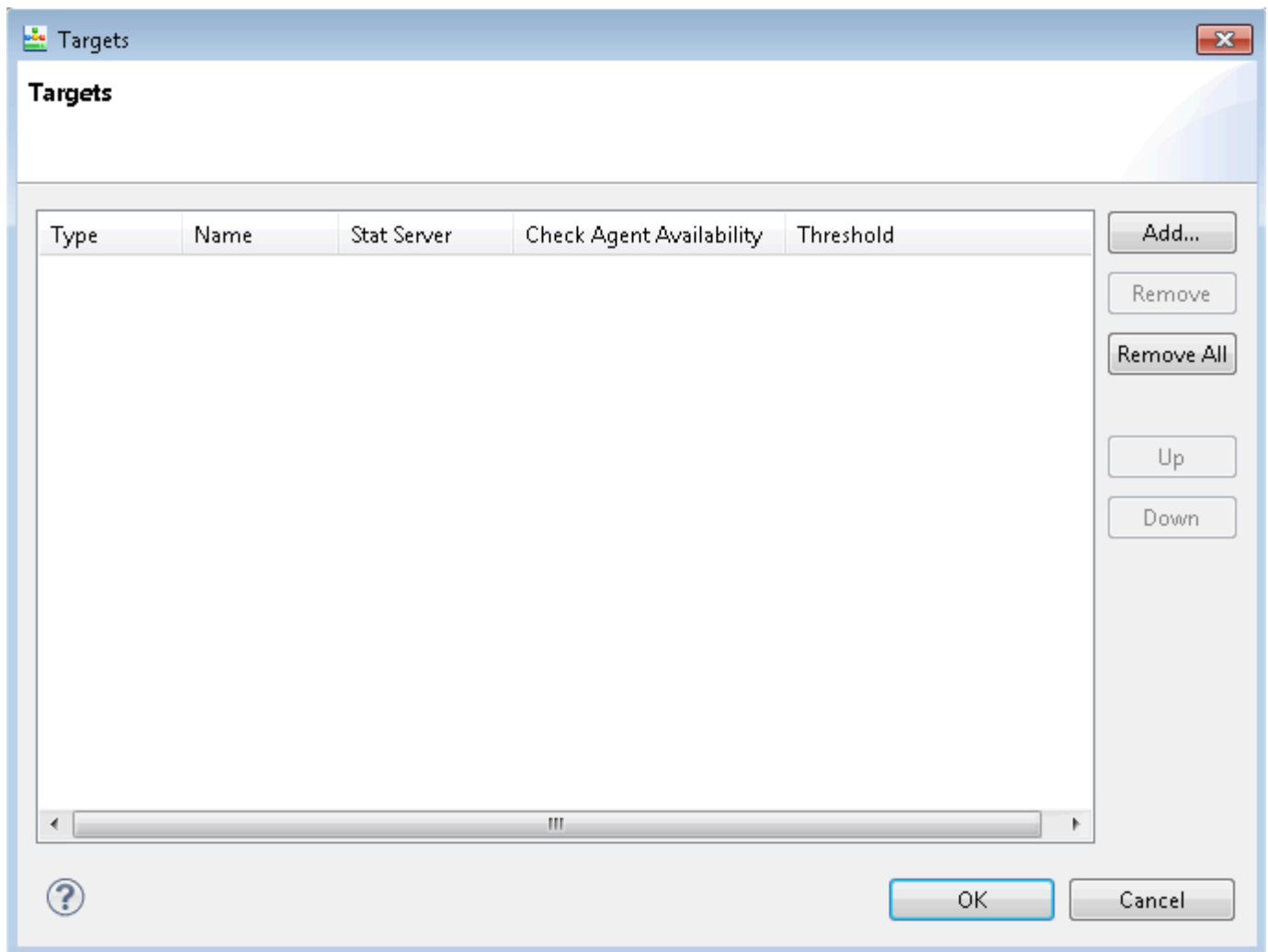
- Select Max or Min to specify whether the interaction should be routed to the target with the maximum or the minimum value of the statistic.
- Select to route interactions to targets based on a [percentage allocation](#). This selection causes the dialog box that opens from the Targets property to display a column where you can specify a percentage for each target.



## Targets Property

Use this property to specify routing targets and for percentage or conditional routing.

1. If you have not already done so, [connect to Configuration Server](#).
2. Opposite the Targets property, click under Value to display the  button.
3. Click the  button. The Targets dialog box opens.



The figure above shows the dialog box when **Min** or **Max** is selected for the **Statistics Order** property. In this case, a Stat Server column appears. If you select **Percentage** for the Statistics Order property, a **Weight** column appears instead of a Stat Server column (see Statistics Order property above).

4. Click **Add** in the Targets dialog box.
5. Click under **Type** to display a down arrow.
6. You can route based on various criteria. Click the down arrow and select the target Type (defined in [Genesys Administrator](#)): **Agent**, **AgentGroup**, **ACDQueue**, **Campaign Group**, **Destination Label**, **LCA**, **Place**, **PlaceGroup**, **Queue Group**, **Route Point**, **Skill** or Variable. At runtime, the application reads the key-value pairs in the targets section. The name part of the key-value pair can be used at your convenience. Only the value part of the key-value pair is read by the application and must be defined as indicated under <targetid> in the *Orchestration Developer's Guide*, [Parameter Elements](#). For example, a valid value is: ksippo@.A
7. Click under **Name** to display to bring up a dialog box. Targets of the selected **Type** appear for selection.
8. Select the name of routing target and click **OK**. You have the option to add another target.
9. If applicable, select a **Stat Server**. Or, if you selected **Percentage** for the Statistics Order property,

enter the **Weight** column.

10. Starting with 8.1.410.14, you can check agent availability in the Target block before executing `<queue:submit>` by selecting true for **Check Agent Availability**. This "skip targets" feature can improve the efficiency of finding an agent by enhancing or relaxing the target criteria without waiting until the routing timeout is reached in the Target block. If multiple/mixed target types are used in the dialog, the skip target-enabled target types will take precedence and will cause Target block execution to be skipped if no agent is available. In this case, a warning message appears stating that the Targets block uses different target types with skip targets enabled.
11. The Target block adds a new "Skip Target" output in the workflow diagram if **Check Agent availability** is enabled for one of the target types in the block. **Note:** This feature is not supported for the following target types: List Objects, Routing Points, Destination Labels, Campaign Groups, Skill expressions, and Variables.
12. Starting with 8.1.440.18, you also have the option of using the **threshold functions** for conditional routing. You can use Threshold to define additional conditions the target must meet to be considered as a valid routing target. Click the button under Threshold to open **Expression Builder** where you can create a threshold expression. See **Creating Threshold Expressions** below. Threshold is not supported for the following target types: List Objects and Variables.
13. Click OK when through in the Targets dialog box.

### Important

The option of routing an interaction to the last called agent (LCA) is available only from release 8.1.510.12

## Target Component Selected Property

Select a variable containing the agent-level target to which the interaction was routed or should be routed to definitively.

- If the target specified in `<submit>` and selected for routing is of type Agent, Place, Queue, or Routing Point, this contains the target itself.
- If the desired target type is Agent Group, Place Group, or Queue Group, the function returns the agent, place, or queue from the corresponding group the interaction was sent to.

The target format is `isName@StatServerName.Type`. The selected variable will be updated with the target information after receiving a `queue.submit.done`.

## Target Deviation From Ideal Agent

New with 8.1.410.14. Select a variable to hold the result of calling the Universal Routing Server (URS) `TargetListSelected` function to calculate a value indicating an agent target's deviation from **ideal agent** to handle an interaction. When the property is assigned to a variable, Composer makes a call

## Target Block

---

to URS right after getting `queue.submit.done`:

```
'urs/call/@' + system.InteractionID + '/func/TargetListSelected?params=[]'
```

URS returns a string similar to the sample below:

```
return:ok|dn:7001|rdn:7001|switch:SIP_Switch|agent:KSippola|place:SIP_Server_Place1|
target_location:Stat_Server|stat_value:0|cost.type:X
```

If the **Set Ideal Agent block** is used before the Target block, the URS reply includes a mismatch key-value pair. The below sample was obtained with:

```
SetIdealAgent[Billing > 6</tt>] and the target with Skill <tt>Billing == 5</tt>).<br>
<pre>return:ok|dn:7001|rdn:7001|switch:SIP_Switch|agent:KSippola|place:SIP_Server_Place1|
target_location:Stat_Server|stat_value:0|mismatch:2|cost.type:X
```

In this sample, the Target Deviation From Ideal Agent variable is assigned the (integer) value (i.e., 2) of the mismatch key-value pair when available, undefined otherwise.

## Target Object Selected Property

Select a variable containing the high-level target (one that you specify in a `<submit>`) to which the interaction was routed or should be routed to definitively. If a skill expression is used, the function returns: `?:SkillExpression@statserver.GA` or even `?GroupName:SkillExpression@statserver.GA`. The target format is `Name@StatServerName.Type`. The selected variable will be updated with the target information after receiving a `queue.submit.done`.

## Target Selected Property

Select a variable containing the DN and the switch name of the target to which the interaction was routed or should be routed to definitively. The target format is `Name@SwitchName.Type`. The selected variable will be updated with the target information after receiving a `queue.submit.done`.

## Timeout Property

If not specified, validation generates an error.

If not specified, validation generates an error. Enter an integer to specify the time in seconds an interaction waits for an available target or keep the default of 10 (added starting with 8.1.440.18). If

the timeout expires before one of the targets is available, the interaction is routed to the error port (if the exception property is configured for the block).

Upon entering any Target block, if a target was not selected and at least one statistic is not open and waiting time is 0, URS adjusts waiting time to the maximum specified in URS options reservation\_pulling\_time and treatment\_delay\_time values. This eliminates the possibility of not routing the first call after URS is started.

## Total Size Property

Select the variable to contain the total number of agents that can be targeted for the request. The selected variable will be populated with the respective output value obtained from the operations described in the [Expected Wait Time\\_Property](#). You must have installed Universal Routing Server 8.1.3+.

## Type Property

Use to define the type of redirection processing that is to be done. For more information and individual values, see the *Orchestration Server Developer's Guide* on the [Orchestration Server Wiki](#).

## Use Access Code Property

Set to true to update the DN property of the resource reserved based on routing criteria to include the access code returned by URS. If true and the Route property is set to false, the resource object will still be modified even though the Target block will not route the call to the resource.

## Use Access Code Setter Property

This property specifies the name of the function to use to update the resource with the returned access code. By default, this property points to the updateDN(accesscode, dn) function from include/RouteFeature.js. You can update the provided function so that all Target blocks that use this function benefit from the update. Or you can provide your own function to implement specific needs for a specific block.

## Use Treatments Property

You can specify Busy treatments that will be executed while waiting for targets to become available. To enable Busy treatments, set this property to **true**. Busy treatments are similar to Mandatory

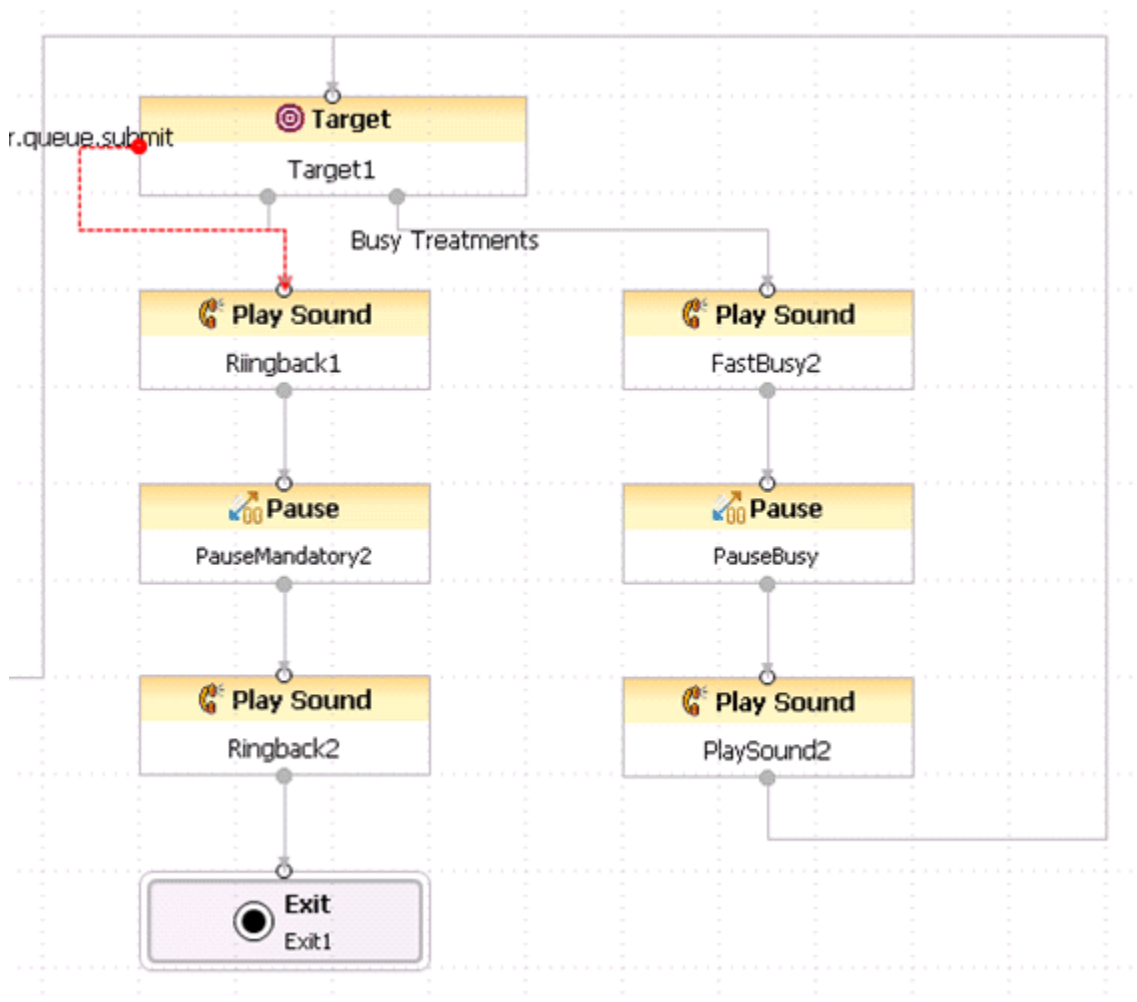


treatments that you can specify using individual **Voice Treatment** blocks.

This property provides instructions to Universal Routing Server for handling the interaction before an eligible target is available to receive it. Treatments (or Busys treatments) are played to callers while the call is waiting in a queue. You can specify any number of treatments for an interaction. The waiting interaction receives each treatment for the specified duration, in the specified order.

Any time that URS finds an available target for the call, the previously-started and currently-played busy treatment (and the entire Busy treatments chain) will be interrupted in favor of routing the call to a target. Target availability is checked upon starting a treatment, on updates from Stat Server, and usually every 2 seconds (the value of this time interval is configurable by the URS option `pulse_time`).

In order to specify treatments, this property must be set to **true**. Setting to true displays a new Busy treatments outport on the Target block. As shown below, **you can connect treatment blocks to this outport** that should be used for Busy treatments. Connect the last Busy treatment block back to its Target block. This indicates the completion of the Busy treatments loop.





### Important

Busy treatment blocks cannot handle exceptions or be used for exception handling. Diagram validation will check that the loop is completely correct and that only appropriate blocks are used for Busy treatments.

## Virtual Queue Property

Use this property to define the enabled ACD virtual queue for the tenant. The virtual queue is not a physical queue but rather a logical queue to which interactions are queued if the specified targets are not available. Also see the **Virtual Queue Selected** property, which lets you select a variable. To define the virtual queue:

1. Opposite the Virtual Queue property, click under **Value** to display the  button.
2. Click the  button. The Virtual Queue dialog box opens.
3. Configuration Database are listed for selection. Select a queue.
4. Click **OK**.

## Virtual Queue Selected Property

Select a variable containing a logical queue (Alias) to which interactions are queued if the specified targets are not available. Also see the **Virtual Queue** property below, which lets you specify the queue.

## Creating Threshold Expressions

When using the Target block, Targets property, to define a threshold expression, there are two different cases:

- The threshold expression uses the function(s) sdata, acfgdata, lcfgdata. Composer converts the expression into a string. At runtime, ORS passes this string to URS. The expression must not contain variables as URS will **not** be able to evaluate them. For example:  
`Threshold=sdata('agtgrp1.GA', 'StatAgentsAvailable') > 12`
- The threshold does not contain any occurrence of the sdata, acfgdata, lcfgdata. Composer included the expression as is in the generated SCXML and ORS will evaluate the expression before passing the result of that expression evaluation to URS. The expression may use variables. For example:

Threshold=myvar

The variable value may contain occurrences of sdata, acfgdata, lcfgdata functions. For example:

```
myvar="sdata('agtgrp1.GA', 'StatAgentsAvailable') > 12"
```

Or Threshold=var1 + var2 where var1="sdata('agtgrp1.GA', 'StatAgentsAvailable') > "  
and var2="12"

See [Queue Interface Action Elements](#) for more detail.