



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.

Designer User's Guide

Route Agent Block

Contents

- 1 Agent Routing tab
 - 1.1 Agent Routing
 - 1.2 Other Routing Settings
- 2 Treatments tab
 - 2.1 Audio
 - 2.2 Shared Module
- 3 Advanced tab
 - 3.1 Targeting
 - 3.2 Extensions
- 4 Results tab



- Administrator

This block enables you to route to an agent based on the specified Agent ID and Virtual Queue, Agent Login and SIP Switch, or Last Called Agent.

Related documentation:

-

You can use the **Route Agent** block in the **Assisted Service** phase to route calls to an agent based solely on:

- a specified **Agent ID** and **Virtual Queue**
- a specified **Agent Login** and **SIP Switch**, or
- the **Last Called Agent** that the caller spoke to

To route calls based on skills and other criteria, use the Route Call block.

You can sequentially place multiple **Route Agent** blocks with different settings, so that if routing fails in one block, your application proceeds to the next block. When a **Route Agent** block successfully routes the call to an agent, the application moves to the **Finalize** phase, ignoring any subsequent blocks in the **Assisted Service** phase.

Important

The **Route Agent** block can only be used to route voice calls. It does not support the routing of chat interactions.

Agent Routing tab

Agent Routing

Select a Routing type:

- **Agent ID** - Select the **Agent ID** (specified as a variable) and **Virtual Queue** that the call will be routed to.
- **Agent Login** - Select the **Agent Login** (specified as a variable) and **SIP Switch** (where the selected agent is logged on, also specified as a variable) that the call will be routed to.

- **Last Called Agent** - Select this option to route the call to the agent that the caller last spoke with.

Important

If you select the **Last Called Agent** option, you must also select the **Enable Customer Profile** option in the Contextual Data tab of the Application Settings.

Other Routing Settings

- **Overall timeout** - If enabled, you can specify how long the application must wait before moving on to the next block. Optionally, you can enable the check box to specify a variable.
- **Route only if the Agent is ready** - If enabled, a call is routed to an agent only if his status is set to Ready. If disabled, the call is routed to an agent regardless of his status.

Treatments tab

Specify a busy treatment to execute while waiting for an agent to become available. You can choose to play audio and/or execute a shared module.

- Learn more about busy treatments

Important

After a busy treatment has been executed at least 10 times, Designer exits the **Route Agent** block and moves to the next block if the average duration of the treatment is less than 1000 ms (for example, due to a missing audio file).

Audio

Click **Add Audio** to add a Play Message child block underneath this **Route Agent** block. The collection of audio plays repeatedly until the call is successfully routed or times out.

Shared Module

Click **Add Module** to add a Shared Module child block underneath this **Route Agent** block. In the child block, you can select a shared module to execute.

A potential use case is to execute a shared module based on a specified set of conditions that can change over time and respond to external factors. For example, you might use a shared module that can play one announcement for callers if the estimated wait time (EWT) is beyond a certain threshold, and another announcement for when they are the next caller in the queue. To set up this feature:

1. In the application, create a user variable, `ewt`, and set its default value to 0.
2. Create a **Self Service** type shared module.
3. In the shared module, create a user variable, `ewt`, and set its default value to 0.

Properties - Initialize



This block or phase is typically used to setup variables for the application and initialize them. Assign blocks can be used to calculate expressions and assign their results to variables in this phase.

User Variables

System Variables

Specify User Variables. String values must be surrounded by single quotes.

[+ Add Variable](#)

Name	In	Out	Default Value	Private	Delete
ewt	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0	<input type="checkbox"/>	

4. In the **Self Service** phase of the shared module, add a **Segmentation** block. Add the conditions as shown below:

Properties - Segmentation



This block is used to evaluate expressions and take different paths in the application based on the outcome. E.g `varZipCode==94014` can be used to take a different path vs `varZipCode==95125`.

Conditions

Milestone

[+ Add Condition](#)

Segment Label	Condition Expression	Delete
EWT less than or equal to	<code>EstimatedWaitTime <= ewt</code>	
EWT greater than last wai	<code>EstimatedWaitTime > ewt</code>	

5. Add two **Play Message** blocks as child blocks of the condition blocks, and add an **Assign Variables** block at the end. Your shared module should appear as shown below:

The screenshot shows a configuration interface for a 'Self Service' process. At the top, there is a 'Segmentation' block with two conditions: 'EWT less than or equal to last wait' and 'EWT greater than last wait', each with a 'Play Message' sub-block. Below these is an 'Assign Variables' block.

6. Configure the first **Play Message** block. An example is below:

Properties - Play Message



This block is used to play audio messages. These messages can be TTS (Text to Speech), Audio Files (previously uploaded in Audio Resources page, or variables played as TTS).

Specify prompts to be played

+ Add Prompt

Type	Var?	Value	Play as	Actions
TTS	<input type="button" value="▼"/>	<input type="checkbox"/> Transferring. Please be patient. Your estim	<input type="button" value="text"/> <input type="button" value="▼"/>	<input type="button" value="▲"/> <input type="button" value="▼"/> <input type="button" value="✖"/>
TTS	<input type="button" value="▼"/>	<input checked="" type="checkbox"/> EstimatedWaitTime	<input type="button" value="text"/> <input type="button" value="▼"/>	<input type="button" value="▲"/> <input type="button" value="▼"/> <input type="button" value="✖"/>
TTS	<input type="button" value="▼"/>	<input type="checkbox"/> minutes.	<input type="button" value="text"/> <input type="button" value="▼"/>	<input type="button" value="▲"/> <input type="button" value="▼"/> <input type="button" value="✖"/>

7. Configure the second **Play Message** block. An example is below:

Properties - Play Message



This block is used to play audio messages. These messages can be TTS (Text to Speech), Audio Files (previously uploaded in Audio Resources page, or variables played as TTS).

Specify prompts to be played

[+ Add Prompt](#)

Type	Var?	Value	Play as	Actions
TTS	<input type="button" value="▼"/>	<input type="checkbox"/>	We are sorry for the delay, the next agent w	<input type="button" value="text"/> <input type="button" value="▼"/>
TTS	<input type="button" value="▼"/>	<input checked="" type="checkbox"/>	EstimatedWaitTime	<input type="button" value="text"/> <input type="button" value="▼"/>
TTS	<input type="button" value="▼"/>	<input type="checkbox"/>	minutes.	<input type="button" value="text"/> <input type="button" value="▼"/>

8. Configure the **Assign Variables** block as shown below:

Properties - Assign Variables



This block can assign values of expressions to variables. Define a variable in the Initialize phase or block and select it in this block to assign it values or results of ECMAScript expressions. You can also call ECMAScript utility functions, such as sorting an array, and provide an input to be run through the function.

[Assignments](#)

[Sort Function](#)

String values must be surrounded by single quotes.

[+ Add Assignment](#)

Variable	Expression	Delete
ewt	EstimatedWaitTime	

9. In your application, select the **Route Agent** block and click the **Treatments** tab.
10. Click **Add Module**. A child **Shared Module** block appears beneath the **Route Agent** block.
11. In the child **Shared Module** block, select the shared module that you created in Step 2.

The application passes **ewt** to the shared module, along with the system variables, which includes **EWT**. The shared module compares **ewt** and **EWT** in the **Segmentation** block and executes a **Play Message** block, depending on which variable is larger. At the end, the shared module sets **ewt** to **EWT** before returning to the application.

Advanced tab

Targeting

Use Agent Status

If checked, the routing engine uses the **status** flag to route the call to an agent. If not checked, the routing engine uses the **loggedin** flag to route the call to an agent.

Important

This check box has no effect if you enabled the **Route only if the Agent is ready** check box in the **Agent Routing** tab.

Threshold Expression

This option enables you to use an ECMAScript (or JavaScript) expression to further refine a routing threshold for the specified target(s). Threshold expressions for the **Route Agent** block can be used for the following routing types:

- Agent ID
- Agent Login
- Last Called Agent

Threshold expressions can contain variables or reference queue-specific values, such as `sdata(target, statistic)` or `callage()`. Strings must be enclosed in single quotes. For example:

Threshold Expression `'callage() < ' + myvar`

For more information about using ECMAScript in Designer, see [ECMAScript Expressions](#).

Important

For routing types that have multiple targets, the script defined in **Threshold Expression** applies to all targets.

Extensions

Use this section to add an extension as a key-value pair to this block.

Use JSON format

This option allows certain special characters to be used in extensions, such as full stops (.), single quotes ('), double-quotes ("), and At signs (@).

Add Extension Data

Click **Add Extension Data** to add an extension as a key-value pair to this block. The value type can be a string or integer.

If you want to use a variable for the **Key** or **Value**, select the **Variable** checkbox and then select a variable from the drop-down menu. If the **Value** is an integer, select the **Integer** checkbox.

You do not need to enclose extension values in quotes. However, if the quote is part of the value, you must escape the quote character by using a preceding backslash. For example:

- Incorrect: Joe's Pizza
- Correct: Joe\'s Pizza

Important

Designer displays an error message if Extension Data is added, but the **Key** and **Value** settings are not defined.

This example shows a few different ways that key-value pairs can be added as extensions:

Extensions

Specify the key/value pairs to be added as extensions

+ Add Extension Data

#	Variable?	Integer?	Value	Delete
1	Key	<input type="checkbox"/>	ExtenString	
	Value	<input type="checkbox"/>	welcome	
2	Key	<input checked="" type="checkbox"/>	varExampleKey	
	Value	<input checked="" type="checkbox"/>	varExampleValue	
3	Key	<input type="checkbox"/>	ExtenInteg	
	Value	<input type="checkbox"/>	123	

Results tab

Select a variable to store the result of this **Route Agent** block execution.