# GENESYS™

# Composer Help

## Skill Expression Builder

4/10/2025

# Skill Expression Builder

## Contents

You can route interactions to the most appropriately skilled agent using a skill expression or a statistical expression. Skill Expression Builder lets you create both types of expressions, which produce a result of true or false.

- For a video tutorial on defining Skills and other objects that can be used in Skill Expression Builder, see Defining Agents, Agent Groups, and Skills.

- For a video tutorial on using Skills as routing targets, see Skills-Based Routing.

## Using Skill Expression Builder

Open Skill Expression Builder from the Targets property in the Target block as follows:

1. If you have not already done so, connect to Configuration Server. Otherwise, when selecting a Target of Type **Skill**, Skill and other Configuration Database objects will not be available for selection.

2. Set the Validate Skill Expressions preference.

3. Opposite the Targets property, click under Value to display the ⬚ button.

4. Click the ⬚ button. The Targets dialog box opens. An example completed dialog box is shown below.
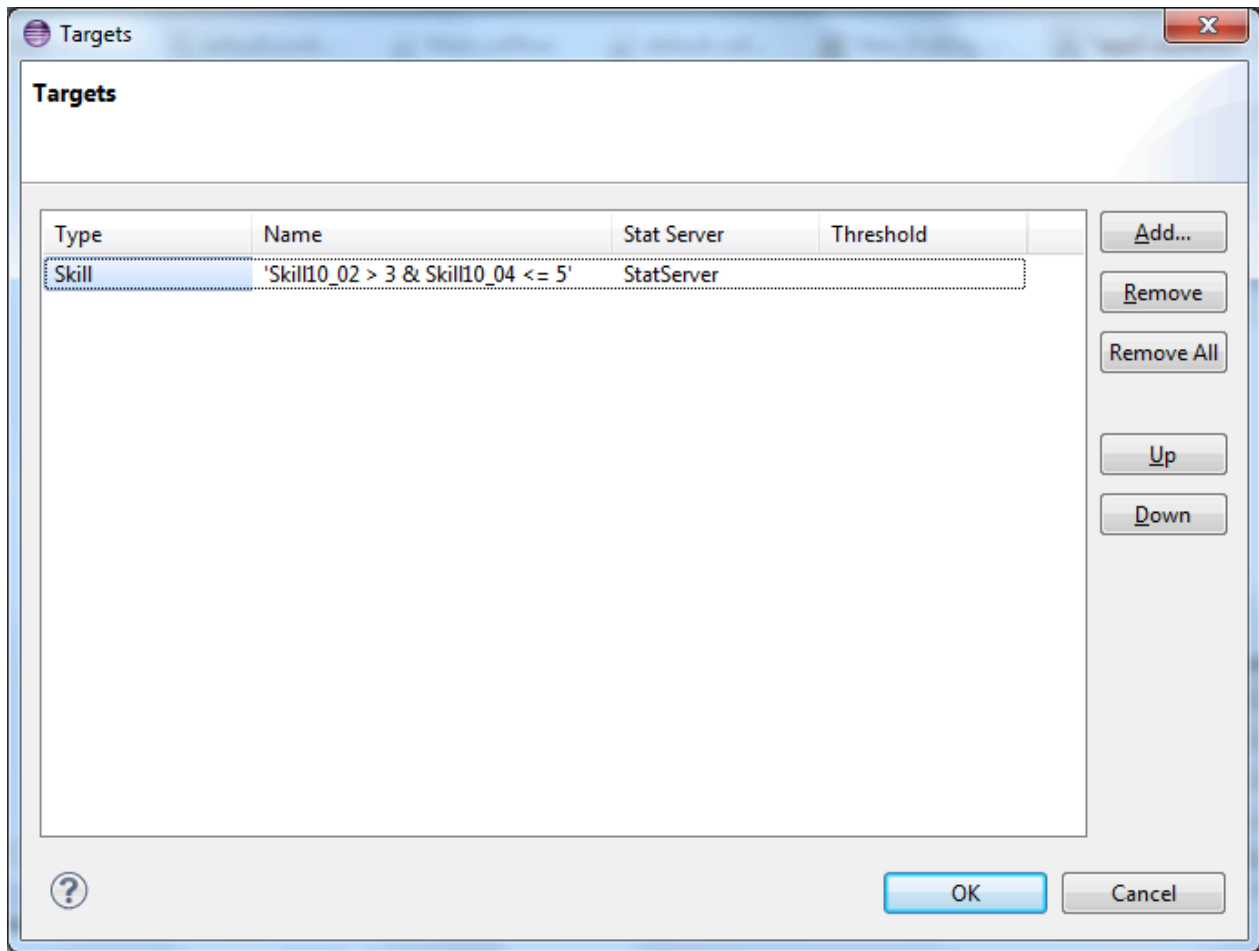
5. Click **Add** in the Targets dialog box.

6. Click under **Type** and select Skill.

7. Click under **Name** to open the Skill Expression Builder.

8. Expand the Skill folder, select a **Skill**. An example completed dialog box is shown below.
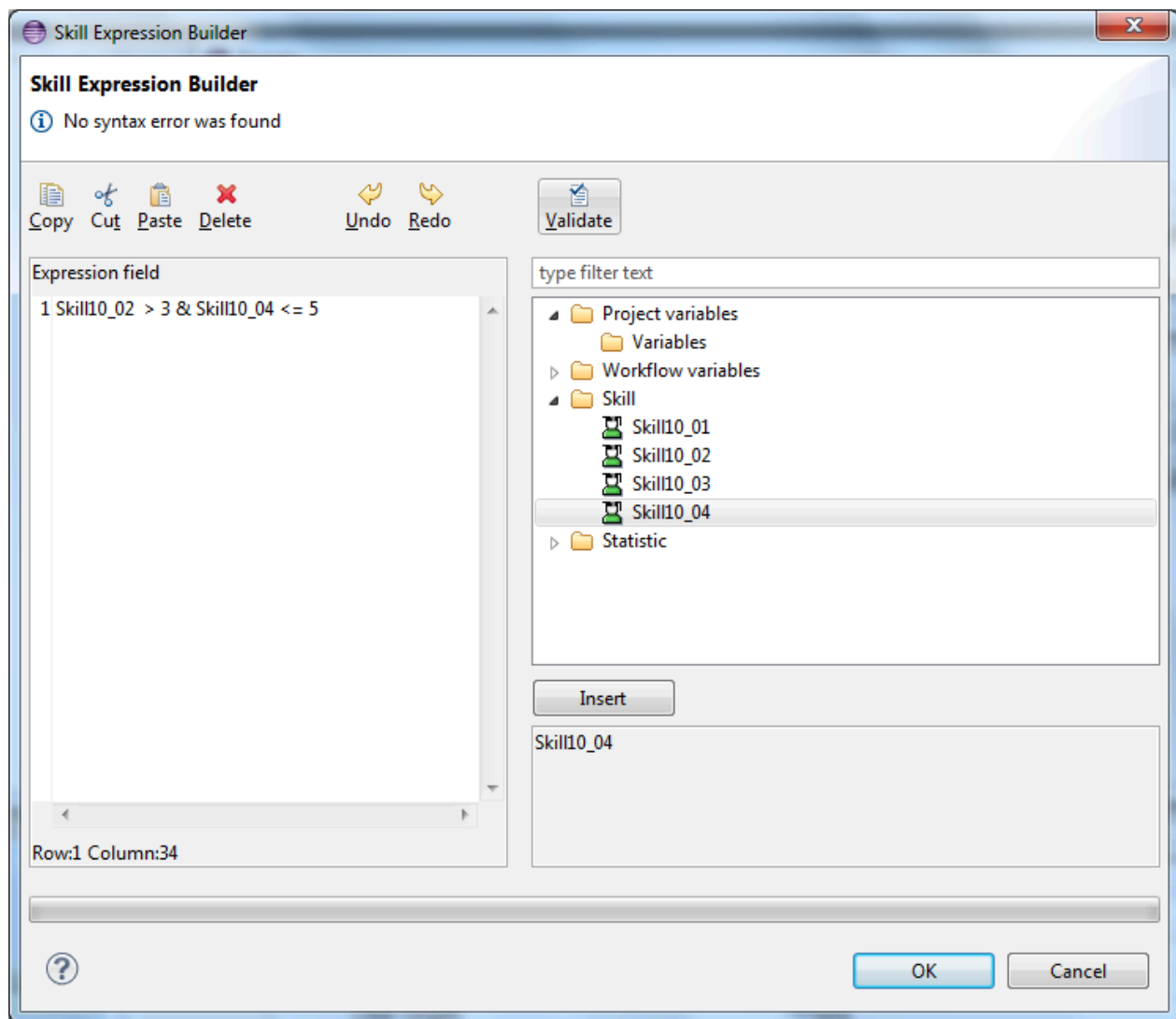
## Important

Starting with version 8.1.450.04, single quotes are not added automatically for multi-byte characters, if not provided by the user in the expression.

You can also use variables and/or Statistics.

## Variables

Valid and invalid examples are shown below.

```
site = 1 & english = 1 & account = 1 is not a valid expression
'site = 1 & english = 1 & account = 1' is a valid expression
Because Composer is able to validate 'site = 1 & english = 1 & account = 1', it
```

```
automatically adds the missing quotes
site = vsite & english = vlanguage & account = vaccount is not a valid expression
'site = ' + vsite + ' & english = ' + vlanguage + ' & account = ' + vaccount is a valid
expression
```

## Statistics

The statistic name in a skill expression can be any agent statistic used in URS function SData, which returns the current value of the statistic for a given target. For example, you may wish to have URS return the number of interactions waiting, so that if a target is not available, the caller will hear the IVR announce the number of interactions ahead of him. The selected elements appear under Expression. The statistic must be written in the format: `$(statisticname)`. For example: `$(StatAgentLoggedIn)=1`

9. Under Skill Expression on the left, define the Skill Expression. Skill Expressions are limited to alphanumeric characters and underscores, cannot begin with a digit, and cannot exceed 126 characters.

10. Click **OK**.

11. Back in the Targets dialog box, you have the option of entering a threshold function for conditional routing.

## Skill Function

Beginning with release 8.1.530.17, a new category folder, Skill Function, is added to the Skill Expression Builder. Currently, it contains the `sitex()` function.
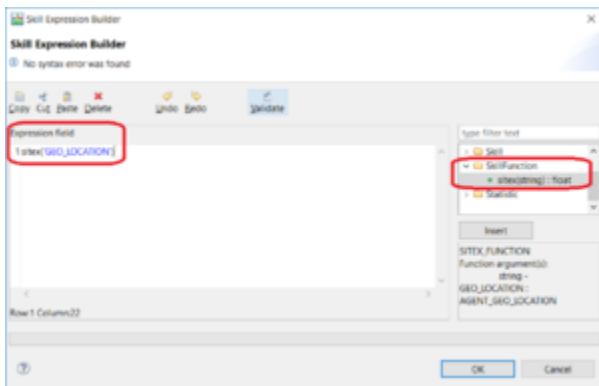


You can use the `sitex()` function to route interactions to targets based on their location. For example, `sitex(GEO_LOCATION) & English > 0`. For more information on the `sitex()` function, refer to the sitex() function description in the URS documentation.

> ## Important
> `GEO_LOCATION` must be specified at the Application level in SIP Server. Routing based on location using the `sitex()` function will work only in environments with SIP Cluster.

## Using Statistic

1. Expand **Statistic**. URS predefined statistics appear for selection.

2. Select a statistic.

3. Click a comparison symbol.

4. Create the expression using a combination of the comparison operators, functions, values, and variables. an example is shown below:

`$(RStatCallsInQueue) = 3 & $(RStatCost) = 2 | $(PositionInQueue)) >=6.`

Note: Use of the `RStatCost` statistic requires that you have cost-based routing implemented at your site. For a description of each statistic, consult the *Universal Routing 8.1 Reference Manual*.

5. Click the button to validate on the toolbar.

6. Click **OK** when through creating the expression to return to the Targets dialog box.

## Comparison Symbols

The table below describes the comparison symbols used to evaluate a skill condition.

| Symbol | Interpretation |
| --- | --- |
| != | Differs depending on the Data type: Skill–not equal to the indicated level value. Statistic–not equal to the indicated statistic value. |
| < | Differs depending on the Data type: Skill–less than the indicated level value. Note: depending on how you use this operator, it may result in including agents that do not have the skill at all (skill name = 0). For example, with English < 8, the queue functional module includes all agents with the English skill less than 8, and also agents with no English skill at all.<br><br>Statistic–less than the indicated statistic value |
| <= | Differs depending on the Data type:Skill–less than or equal to the indicated level value. Statistic–less than or equal to the indicated statistic value |
| = | Differs depending on the Data type: Skill–equal to the indicated level value. Statistic–equal to the indicated statistic value. |
| > | Differs depending on the data type: Skill–greater than the indicated level value. Statistic–greater than the indicated statistic value |
| >= | Differs depending on the Data type: Skill–greater than or equal to the indicated level value. Statistic–greater than or equal to the indicated |

| |
|---|
| statistic value. |

This is a value of the same data type as the Data name element. The value must already evaluate to an integer. Float numbers are not supported. There are different limitations depending on the data type:

- **Skill value**–This value represents the level of skill. For example; an agent could have an English skill level greater than 3 (English > 3). An agent can be excluded from a skill by setting that agent's skill level for that skill to zero in the configuration layer (English=0).

- **Statistic**–This value represents the value of the statistic/metric. For example; an agent could be in Ready state  longer than 20 seconds (`$(StatTimeInReadyState) > 20`).

## Logic Operators

Use the logic operators to evaluate multiple conditional expressions together. The following logic operators are supported:

- **AND** (&)
- **OR** (|).

The AND and OR logic operators have the same priority. For example:

`English >3 & $(StatAgentLoggedIn)=1`

Variables and Literals

Starting with 8.1:

- Composer supports variables in skill expressions (they appear in the Skill Expression builder tree).
- You must enclose literal expressions in single quotes.

Background: Previously, Composer automatically added single quotes around the expression entered by the user. Now that variables are supported, Composer must distinguish literal strings and variables. As a result, you must enclose literal strings in single quotes.

Also see: Variables Project and Workflow and the Chat Transcript block.