



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.

# Genesys App Automation Platform Help

Script Block

12/14/2025

# Script Block

## Important

This page is only applicable to users with the role **Application Designer**.



You can use **Script** blocks to perform complex operations, such as loops and *if* clauses, and define their own methods.

## Adding this block to the callflow

To add and configure **Script** blocks in a callflow:

1. Drag and drop a **Script** block onto the appropriate position in the callflow.
2. Click the **Script** block to view its properties.
3. In the **Script** tab, select a script type and enter the script in the text box. Both script types are based on Groovy Script.
  - **Easy script** - Click **Add Entry** to add an entry to the script in a simplified interface. This script always returns **success**.
    - **Variable** - Specify a variable name.
    - **Function** - Select whether to specify a value for this variable, or determine the value based on another variable.
    - **Value** - If you selected the **Set to Value** function, specify a value. If you selected the **Set to Variable** function, the value of the variable you set in the **Variable** field is used.
    - **Attach to call** - Enable this option to attach this variable data to the call, which also means it becomes available to agents in CTI Viewer.
    - **Remember** - Store this variable data in the database.
  - **Complex script** - Allows advanced users to use Groovy Script to perform more complex operations, such as loops, *if* clauses, and define their own methods. The value returned by the script is used to select the next path. In most cases, the block returns **success**, but you can also use **Script** blocks for callflow branching or to trigger a global event handler such as **agent** or **recognition failure**.
4. (Optional) In the **Unit Test** tab, you can run tests on **Complex scripts** you have configured. For example, if your script defines a method to perform certain operations, and you want to ensure the results are correct, you can define a unit test for the specific method as follows:

- a. Call a defined method into the script with known set values.
  - b. Get the results from the method.
  - c. Compare the expected results with the results calculated in the method.
5. Click **Update**.

### Important

You can download a JavaDoc version of the scripting commands [here](#).