Genesys Intelligent Automation Help

Creating Applications and Modules

4/9/2025

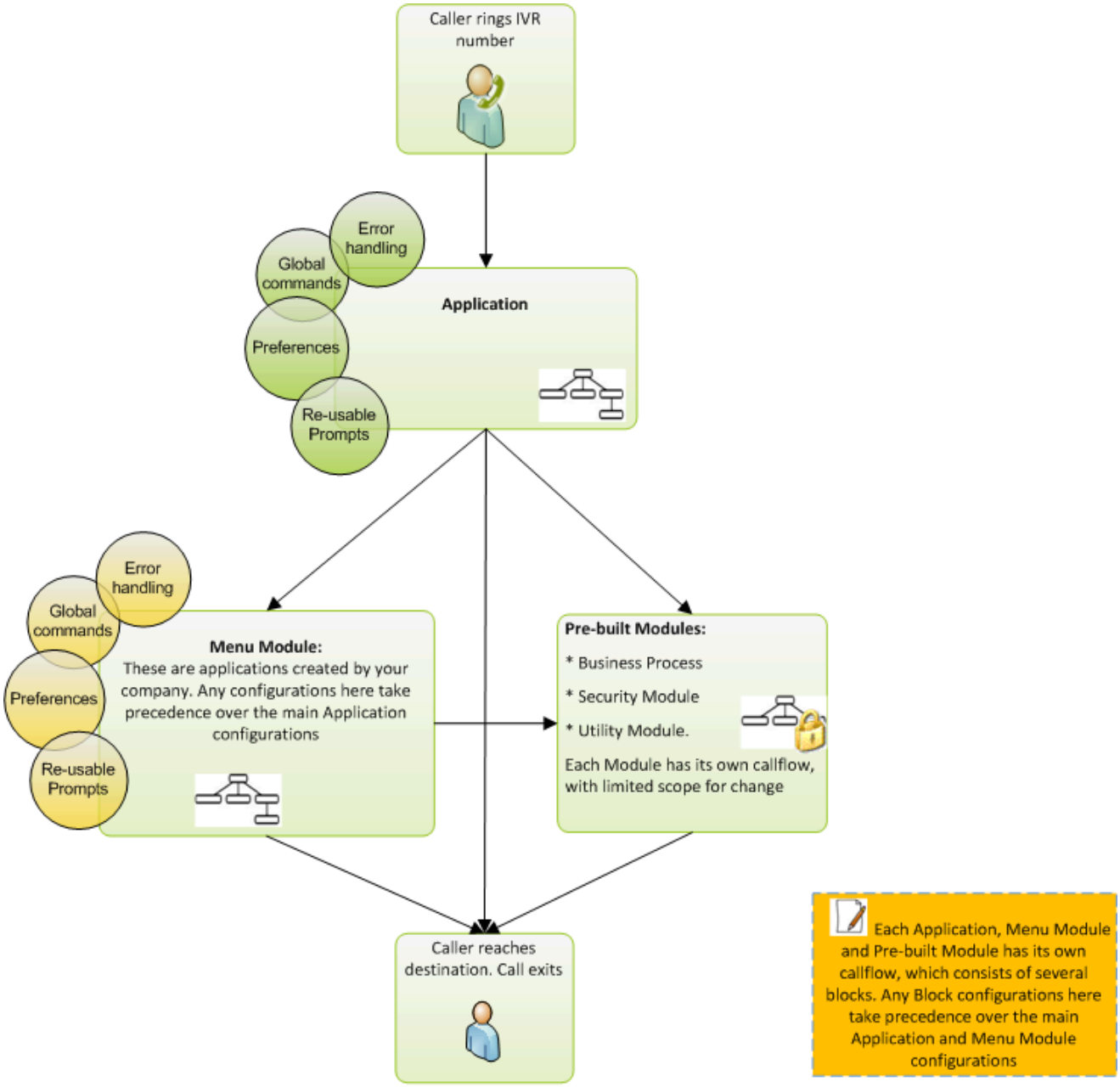
Contents

Creating Applications and Modules

> **Important**
>
> Information in this chapter is dependent on your user role. The following restrictions apply:
>
> - **Application Designer** can create applications and modules and add all types of blocks to a callflow.
> - **Application Maintainer** can only add Message, Menu, Phone, Link, Interceptor, and End blocks to a callflow.

Applications, menu modules and prebuilt modules all allow callers to perform specific self-service tasks over the phone. Callers dial into the application, rather than into individual modules. The application holds all the defaults, global commands, reusable prompts and error-handling paths (as well as the **agent** path).

This main application is based on the Standard Application Template. This template allows you to call into a menu module or a prebuilt module. Menu modules can also call into prebuilt modules, as illustrated in the graphic below:



Applications, menu modules and prebuilt modules can all have their own error-handling, global commands, callflow defaults and reuseable prompts. However, except from callflow defaults, you cannot change these settings for pre-built modules.

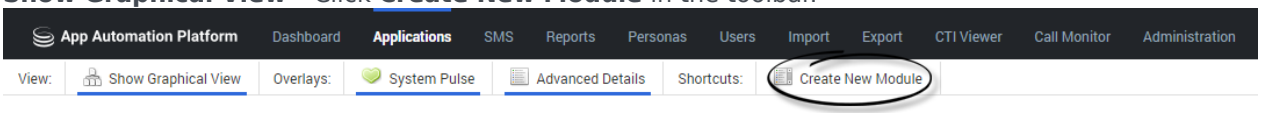
> ## Important
>
> Refer to Understanding settings inheritance for information on the order of

> precedence for each of these aspects.

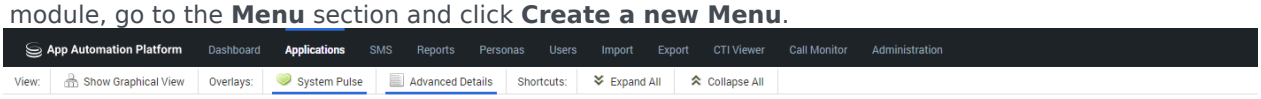
Creating applications and modules

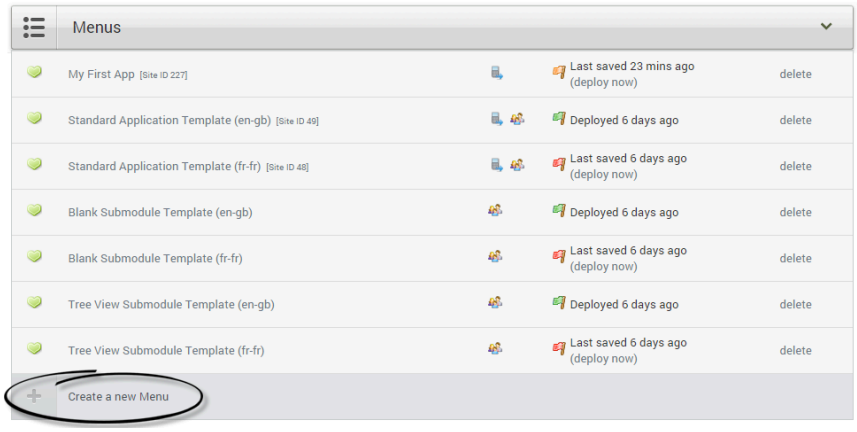
You can create an application or module from the following locations:

- **Show Graphical View** – Click **Create New Module** in the toolbar.



- **Applications** view – Click **Create a new ...** in the desired section. For example, to create a new menu module, go to the **Menu** section and click **Create a new Menu**.





Selecting a template

Next, you must select a template on which to base the module. Each template provides the framework that brings together common elements, such as Start blocks, links to other modules, and more.

After you select a template, a panel appears on the right in which you can enter a name and description for your module.

- **Menus** - These allow you to set up your own callflow using the Callflow Editor.

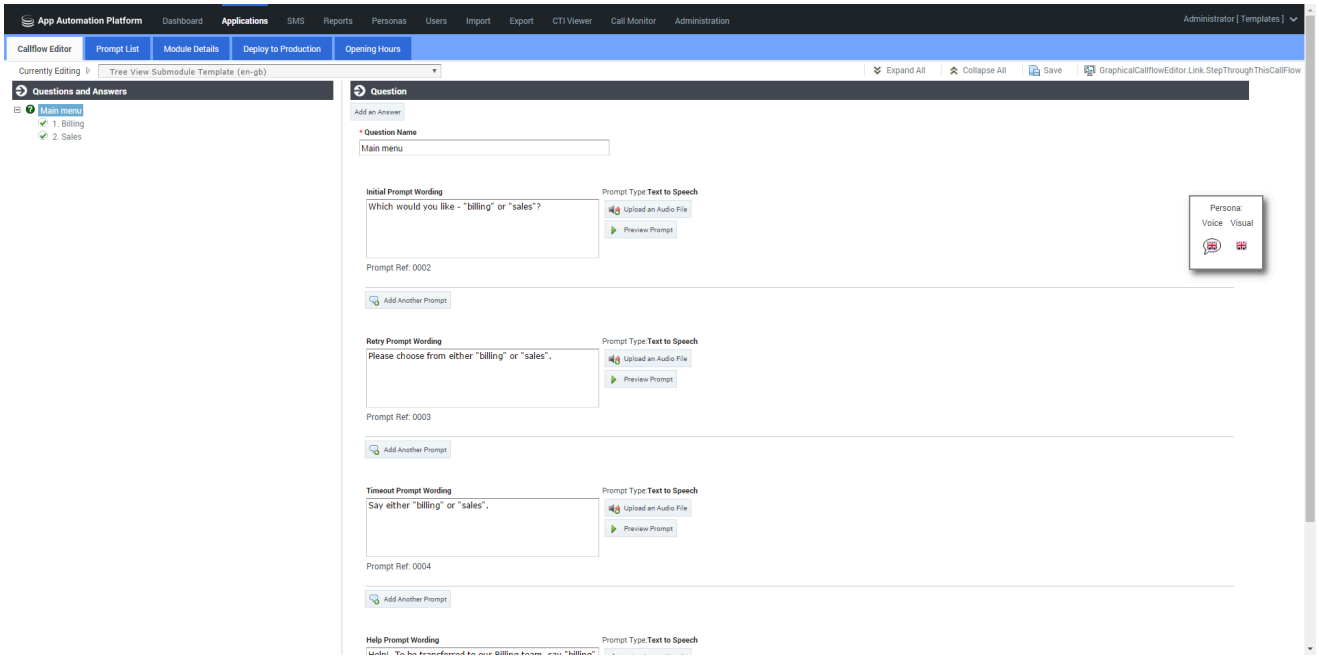
Important

You cannot edit the callflow of the **Tree View Submodule Template**. This template is designed to deal with large numbers of menu options. Refer to the Using the Tree View Submodule Template section below.

- **Security Modules**, **Business Processes**, or **Utility Modules** - You cannot edit the callflow of these modules. However, when you view these modules in the Callflow Editor you might see additional tabs that allow you to configure the default behavior of the callflow.

Using the Tree View Submodule Template

The Tree View Submodule Template differs from other menu templates in that it does not use the drag-and-drop method described in the Callflow Editor section. Instead, the Callflow Editor displays in a tree-view structure, seen below.



Important

You cannot change default behavior and preferences for the Tree View Submodule Template. If your business needs require these defaults to be changed, contact your Genesys representative.

The Tree View Submodule Template allows you to quickly add multiple questions and answers to a callflow. For example, your company might have 30 different events running throughout the year, and

you expect customers to call in to ask for information and make bookings for each of these events. You can use this template and its streamlined interface to quickly set up 30 different menu options to account for each event.

To add questions and answers using the Tree View Submodule Template:

1. Go to the Applications view.

2. In the **Menus** section, click the **Tree View Submodule Template** to open it. The **Question** section appears first:

   a. In the **Question Name** field, enter the name of the first question you want to ask callers.

   b. Update the **Initial**, **Retry**, **Timeout** and **Help** prompts for the first question.

   c. Click **Update**.

> ### Important
> If you make any changes to the Question or Answer prompts and configurations, you must click **Update** to save the changes. Likewise, in the Callflow Editor in which the Tree View Submodule Template is used, you must click **Save** to save the whole callflow.

3. Click **Add an Answer**. The answer appears below the first question in the tree-view list on the left; on the right, the **Answer** section appears:

   a. In the **Answer Name** field, enter the name of this answer.

   b. Depending on how you asked the caller to respond to the question, you must complete one or both of the following:

      - **Recognition Phrases** – Add any phrases you anticipate callers might say to select this answer. Add a carriage return after each phrase.
      - **DTMF** – Enter the digit you want callers to press to select this answer.

   c. Specify what you want to happen when the caller selects this answer. Select one of the following options from the **Action When This Answer is Chosen** list:

      - **No Action** – The module asks the caller to confirm this is the correct answer. If necessary, you can add click **Add a Question** to add a follow-up question.

> ### Important
> If you previously selected **Transfer to a Phone Number** or **Link to Another Module** for this question, you must first select **No Action** and click **Update** before the **Add a Question** button becomes available.

      - **Transfer to a Phone Number** – Enter the phone number to which you want to connect the caller. For example, you can configure this option to dial a sales agent to complete a transaction.

- **Link to Another Module** – Select the module to which you want to link the caller. For example, you might select the **Payment with Full Balance** module if you had asked if the caller wanted to buy a product or service.

d. In the **Prompts** section, update **Confirmation Prompt Wording** and **Information Prompt Wording**.

4. Click **Update**.
5. (Optional) At this stage, you can:
   - Click the first question in the tree view and repeat the steps in the Answer section to add a new answer.
   - Click **Add a Question** to add a subquestion for the first answer.
   - Remove the answer by clicking **Delete this Answer**.
   - Create a new question by following the steps in the Question section.
6. Click **Save**.

Using the Callflow Editor

After you create an application, it opens in the Callflow Editor so you can design it to suit your business purposes.

Callflows consist of various blocks and paths that outline the steps the application must follow when interacting with a caller. See the Using the Callflow Editor page for more information.

Prompts

Prompts can be found in Menu, Message, or Phone blocks. You can play prompts using either TTS (Text To Speech) or prerecorded audio files. Refer to the Prompts page for more information.

Understanding settings inheritance

Understanding the inheritance order enables you to set useful defaults in your main application but override them in specific situations for a particular submodule or block.

For example, you might set the **Maximum retry count** at 1 in the main application but use a higher value for a particular Question block in a submodule that asks a complex question that often takes callers a few attempts to answer. You might also have a particular "Yes/No" question within that submodule that says, "If you don't know the answer just stay silent." To do so, set the **Maximum no input count** value to `0` for that specific Menu block.

Applications and modules inherit settings in the following order:

1. Callflow preferences.
2. Path and menu options.

Callflow preference inheritance

Callflow preferences are inherited in the following order:

1. Current block.
2. Current module defaults.
3. Calling module defaults.
4. Main application defaults.
5. Current server settings page.

Path and menu option inheritance

Paths and menu option settings are inherited in the following order:

1. Current block.
2. Current module defaults.
3. Calling module defaults.
4. Main application defaults.

Setting callflow preferences

The **Preferences** tab in each block dialog box allows you to apply specific rules as to how a call is handled. Refer to the Preferences page for more information.

Setting Opening Hours Rules

Genesys Intelligent Automation uses Opening Hours Rules throughout an application. These rules allow you to specify at which time and on which days a call can be transferred to a specific number. For example, you can apply an Opening Hours Rule to a Phone block to specify what happens if your company is closed:

1. Click the **Opening Hours** tab.
2. Click **Create a New Rule**.
3. Enter a name in the **Rule Name** field.
4. In the **Weekday Opening Hours** section, specify which days of the week that your office is open. For each day, select either **Open**, **Closed**, or **Timed**. If you select **Timed**, specify the opening hours on

that day.

5. In the **Special Dates** section, specify special dates when the usual opening hours do not apply. For example, you can add New Year's Day and select **Closed** for status.
6. In the **Actions** section, specify what you want to happen if the call occurs outside of the opening hours. Select one of the following options from the **Suggested Action If Closed** list:
   - **Transfer** – Transfers the call to another telephone number. You can add several numbers, using a comma-separated list. The system moves on to the next number in the list until the call is answered, or until a **no answer** event.
   - **End the call** – Ends the call and returns a **system hangup** result.
   - **Other** – Specifies another event to trigger (for example, **main menu**).

   In **Out of hours prompt**, enter TTS text or upload an audio file to play to callers if they ring outside of opening hours.
7. Click **Save**.

Testing your application

You can test your application within the Callflow Editor by using the Virtual Call or WebIVR feature.

In the **Test your App** menu, select whether you want to use the **Test** or **Production** version of your application, then click **Virtual Call** or **Web IVR**.

Refer to the Virtual Call or WebIVR pages for more information on using this feature.

Deploying to Production

After you make changes to a callflow, you can use Intelligent Automation to simultaneously deploy a complete IVR application and its associated submodules to your production environment, with the new callflows being applied to the very next call. You do not need to restart Intelligent Automation to deploy changes. Any calls already in progress will be completed using the original callflow.

You can test changes before you deploy them to production. Intelligent Automation provides you with both a test IVR number and a production IVR number. The latter is used to handle live customer calls, while the test IVR number allows you to test applications before deploying them to production. You can test changes made to an application by dialing into the test IVR number, enabling you to experience exactly how the application will perform in live operation.

After deployment, you can roll back the changes to a previous configuration. Intelligent Automation maintains a record of each new configuration with the option to retrieve a previous deployed configuration and use it as the basis for further configuration changes. For example, you might apply specific changes to the IVR to cope with changes in demand during a holiday period. After this period ends, the preholiday version of the IVR can be redeployed as the basis for further changes.

To deploy an application:

1. Perform one of the following options:
   - In the Callflow Editor, click the **Deploy to Production** tab.
   - In the **Applications** view, ensure **Advanced Details** is active and then click **deploy now** beside the application you want to deploy.
2. Enter a description in the **Reason for Deploying** field. This description identifies the main change in this version of the application (for example, `Updated welcome prompt wording`).
3. (Optional) Enable the **Deploy this application's submodules as well** to deploy the submodules linked to this Application.

   > **Important**
   > If you roll back an application, its associated submodules are not rolled back. You must roll back each submodule individually.

4. Click **Deploy to Production Now**.

The table at the bottom of the **Deploy to Production** tab lists all previous deployments, including the version currently in production. In the **Actions** column, you can choose:

- **Delete** - Delete a previous version of the application.
- **Copy to Test Number** - Copy this version to the test number so you can conduct tests. This action overwrites the current test version.

To roll back to a previous version:

1. Click **Copy to Test Number** beside the version you want to restore.
2. Click **OK** when asked if you want to overwrite the current test module.
3. Place some test calls to ensure you are happy with the test version to which you have just rolled back.
4. Repeat the steps to deploy an application.

Updating application or module details

> **Important**
> This section only applies to users with the role **Application Designer**.

To update application or module details:

1. In the Callflow Editor, click the **Application Details** or **Module Details** tab.
2. Update the application or module name and description as needed.

3. Click **Save**.

Deleting applications and modules

> **Important**
> This section only applies to users with the role **Application Designer**.

To delete an application or module:

> **Important**
> Ensure that the **Show Graphical View** option is not selected.

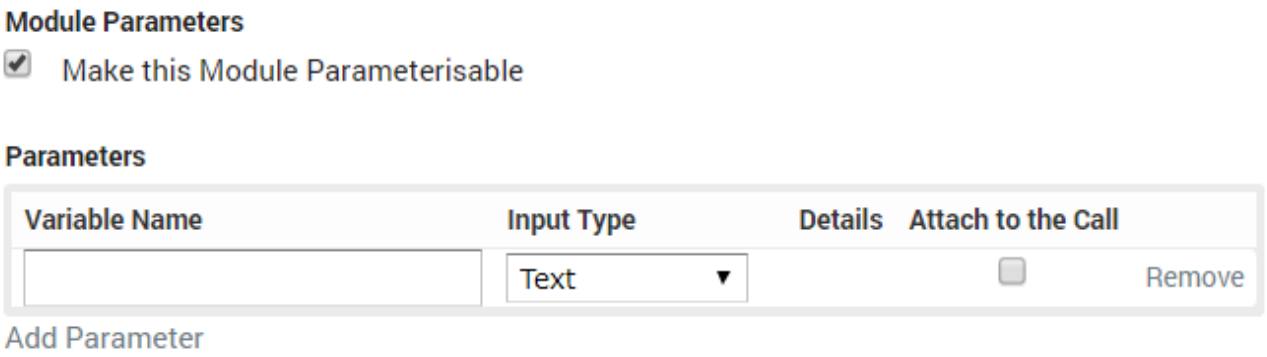
1. Click **Applications** in the navigation bar.
2. Click **Advanced Details**.
3. Click **delete** beside the application or module that you want to delete.

When you delete an application or module, all accompanying data and prompts are also deleted.

Working with module parameters

You can declare variables for an application or module for use whenever the application or module is called. The variables and their values will then be available to the calling block. You can even declare different values for different blocks, depending on which block calls them.

To declare new variables, select **Make this Module Parameterisable** in the **Application Details** or **Module Details** window. This opens an empty list in which you add new variables, like this:



When a call reaches the Link block in the call flow, the call is transferred to the module while setting its transfer parameter to the value set in the Link block. So you can specify different values depending on which Link block the call enters a module from.

There are three types of variables:

- Text variables
- Web Service variables
- Genesys CME variables

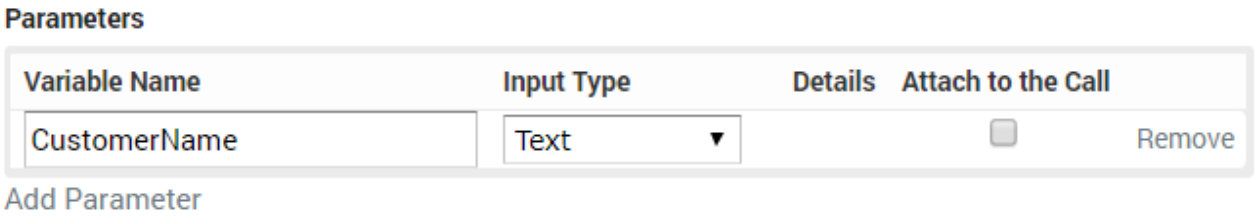
The following sections describe these variables, and how to declare them and specify from where the values are obtained.

Text Variables

Text variables are the simplest type of variables to create. Their value is determined when the Link block is configured with a **Transfer Method**.

To declare a text variable:

1. In the **Parameters List** section of the **Application Details** or **Module Details** tab, click **Add Parameter**.
2. Enter a **Variable Name**.
3. Check that **Text** is selected in the **Input Type** field. **Text** is the default value.
4. Click **Attach to the Call** if you want the variable to be accessible when the call is transferred to a routing strategy.



When you configure a Link block with a Transfer Method, you specify the value to assign to the linked variable.

Web Service

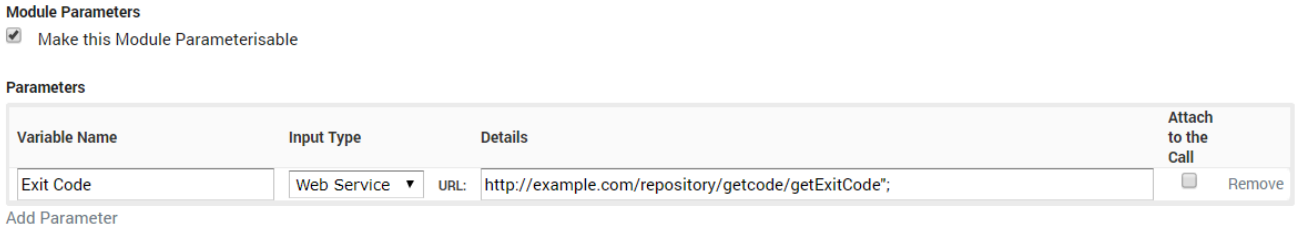
The value of a Web Service variable is derived from a set of values that are returned by a Web service. For example, declare a variable named `exitCode`, and specify the URL of a Web service that returns the following XML:

```
<response>
   <values>
      <value name="EXIT_CODE 1"/>
      <value name="EXIT_CODE 2"/>
      <value name="EXIT_CODE 3"/>
      <value name="EXIT_CODE 4"/>
      <value name="EXIT_CODE 5"/>
   </values>
</response>
```

When the Link block is opened and the Web service is called, the five values (`EXIT_CODE 1`, `EXIT_CODE 2`, `EXIT_CODE 3`, `EXIT_CODE 4`, and `EXIT_CODE 5`) populate a drop-down menu in the Link block, from which you choose the required value for this scenario.

To create a Web Service variable:

1. In the **Parameters List** section of the **Application Details** or **Module Details** tab, click **Add Parameter**.
2. Enter a **Variable Name**.
3. In the **Input Type** field, select **Web Service** from the drop-down menu.
4. in the **Details** field, enter the URL of the Web service from which you will obtain the possible option values.
5. Click **Attach to the Call** if you want the variable to be accessible when the call is transferred to a routing strategy.



Genesys CME

A Genesys CME variable is similar to a Web Service variable, except that it derives its value from a Transaction object of type List in the Genesys Configuration Environment instead of a Web service. This object contains a list of key-value pairs; the values of the keys are assigned to Genesys CME variables based on the key value.

To enable this type of variable, set `FeatureEnablement.SiteParameters.CME` to *true* in the server's default settings. You must use the Web service provided in the **fish-services** Web application. For this Web application to access the Configuration Server, you must provide the access information in its **genesys_cme.properties** file, using a simple text editor to replace the following text in the properties file with the appropriate values:

- **CME.HostName**
- **CME.HostPortNumber**
- **CME.ClientName**
- **CME.UserName**
- **CME.Password**
- **CME.Timeout**

After connecting to Configuration Server, the Web application returns the key-value pairs from the Transaction list specified in the application call. To specify the list that you want to retrieve, append the following to the URL for the Web application:

```
?list=<NameOfList>
```

where <NameOfList> is the name of the Transaction list object as specified in the Configuration Database.

For example:

```
http://<host>:<port>/fish-services/genesys/GetSiteParameters.jsp?list=<NameOfList>
```

This Web Service is called when a Link block linking to that module is opened in the Callflow Editor. From the list that appears in the Link block, you select the item to be passed into the module. The value selected will be stored with the callflow definition when the callflow is saved (during design, not at run time).

> **Important**
>
> You can reassign the variable values at runtime using the Script block or iHUB process.

To create a Genesys CME variable:

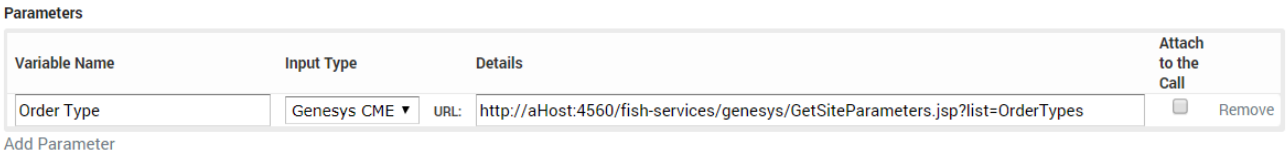
1. In the **Parameters List** section of the **Application Details** or **Module Details** tab, click **Add Parameter**.
2. Enter a **Variable Name**.
3. In the **Input Type** field, select **Genesys CME** from the drop-down menu.
4. In the **Details** field, enter the URL of the Web application from which you will obtain the List.
   - For retrieving a **list of values** into the Link block, use

     ```
     http://<host>:<port>/fish-services/genesys/GetSiteParameters.jsp?list=<NameOfList>
     ```

   - For retrieving **key-value pairs** into the Link block, use

     ```
     http://<host>:<port>/fish-services/genesys/
     GetSiteParametersPair.jsp?list=<NameOfList>
     ```

5. In the Link block, select a key from the list in the Parameter Name field. The corresponding value from the Transaction list will be assigned to the parameter.
6. Click **Attach to the Call** if you want the variable to be accessible when the call is transferred to a routing strategy.



Deleting Variables

To delete a variable, select the variable in the list of parameters, and click **Remove**, located at the end of the row.

> **Warning**
>
> When you remove, or delete, a variable, it is also removed from any parent modules in which it has been saved. Make sure that you really want to remove the variable everywhere, before you continue with the deletion.