Genesys Intelligent Automation Help

Genesys Intelligent Automation Current

10/30/2024

Table of Contents

Genesys Intelligent Automation Help	**4**
Logging In	**5**
Interface Overview	**10**
Dashboard	13
Applications	16
Integration	20
Reports	24
Personas	34
Users	45
Import	47
Export	48
CTI Viewer	49
Call Monitor	50
Administration	61
My Account	76
Using iHub	**78**
Creating Applications and Modules	**98**
Using the Callflow Editor	112
Setting Callflow Preferences	123
Prompts	131
Blocks	139
Start Block	140
End Block	141
Message Block	142
Menu Block	149
Question Block	154
Phone Block	165
Link Block	169
Interceptor Block	171
Script Block	176
Recording Block	179
Rich Media	**181**
Natural Language AI Integration	**188**
Custom Natural Language Menu Microapp	197
Custom Natural Language Menu Dialogflow CX Chatbot Integration	200

Natural Language Audio Streaming Menu 202

MicroApps **206**

Two-factor Authentication 213

Smart Transfer 218

Identification 221

WebIVR Applications **223**

Apple Business Chat **230**

Troubleshooting **235**

Developer Mode 238

Document Change History **240**

Genesys Intelligent Automation Help

This help file introduces you to the Genesys Intelligent Automation user interface and explains concepts and procedures to help you use this software.

Prior to release 9.0, Genesys Intelligent Automation was known as Genesys App Automation Platform (GAAP).

Who should use this document?

The intended audience for this document are users who have been assigned the following roles:

- Maintainers - Individuals responsible for the maintenance of application callflows, such as the update of call prompts.
- Designers - Individuals responsible for the design and configuration of application callflows, such as the addition of extra blocks within a callflow, or the identification and resolution of problem areas in a callflow.
- Administrators - Individuals responsible for administering and configuring contact center settings.

Some sections are only applicable to certain roles and are identified as such.

What is in this document?

This document explains the following topics:

- Logging In - Describes how to log in and out of Intelligent Automation.
- Interface Overview - Describes key terms and concepts, and provides information about the user interface and its views, such as Dashboard, Reports, Administration, and more.
- Integrating Intelligent Automation and Customer Environments - Describes how to install and use the Integration Hub (iHub) to integrate Intelligent Automation with Customer resources.
- Creating Applications and Modules - Explains how to create applications using the Callflow Editor, descriptions of each block, and how to work with prompts.
- Troubleshooting - Provides solutions to common questions you might have about working with Intelligent Automation.

Logging In

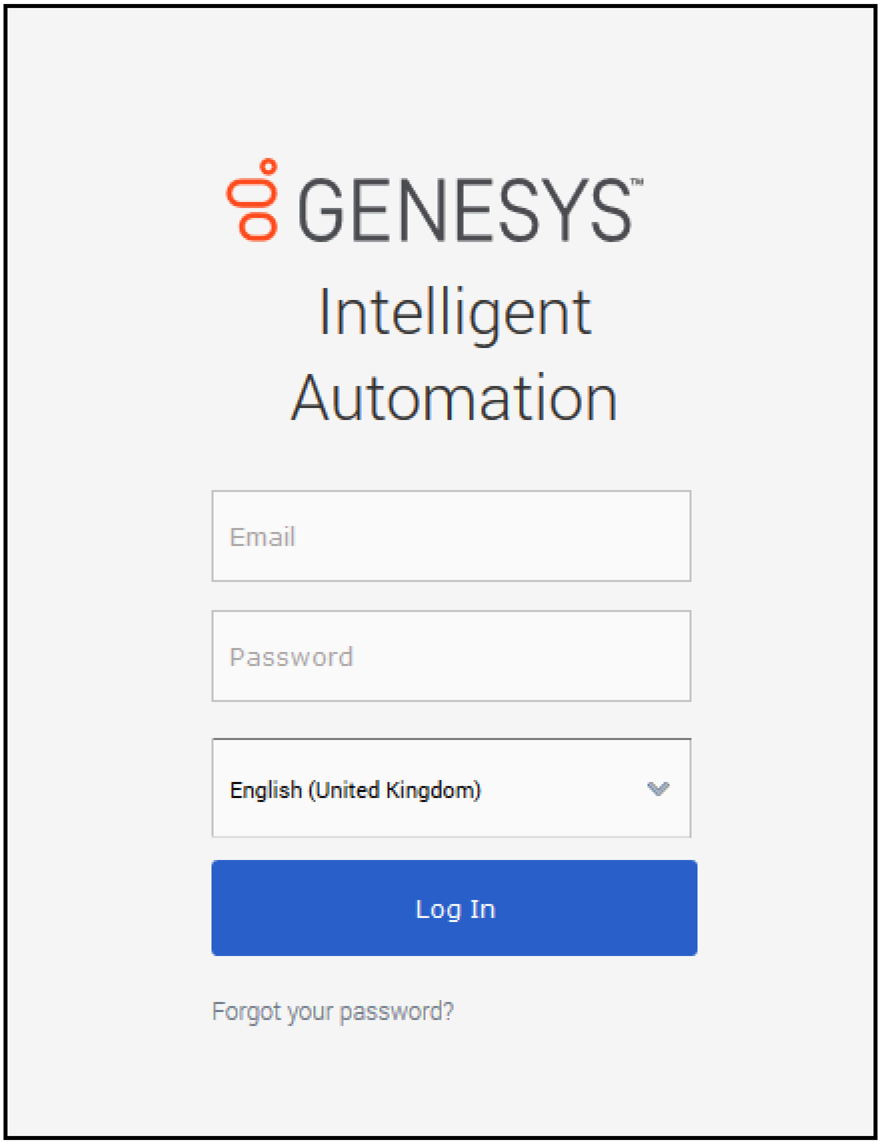
To log into Genesys Intelligent Automation, you must first be created in one of the following two databases by a user with the necessary permissions:

- The Intelligent Automation database
- The Configuration Database, using Genesys Administrator Extension.

Once you are created, you will be notified and given a password.

Log in

To log in, use the URL provided by your administrator to open the login window.



Enter your email address and either your Intelligent Automation or Configuration Server credentials in the appropriate fields. If necessary, select the language in which you want to work from the drop-down list. Then click **Log In**.

What if my login doesn't work?

If your login is not successful, an error message is displayed at the top of the window. Re-try the login. Recheck your credentials and try again. If you can't log in after a few tries, you might want to contact your system administrator.

Warning

For security reasons, you might be limited to the number of retries before you are locked out, that is, you are unable to enter anything in the Login window. In this case, you must wait a while before you can start trying again. Genesys suggests that you use this time to recheck your credentials.

What if I forget my password?

If you forget your password, click **Forgot your password?** in the login window, and the system administrator will email you a link to reset your password. Just click the link in the email, and change your password as described below.

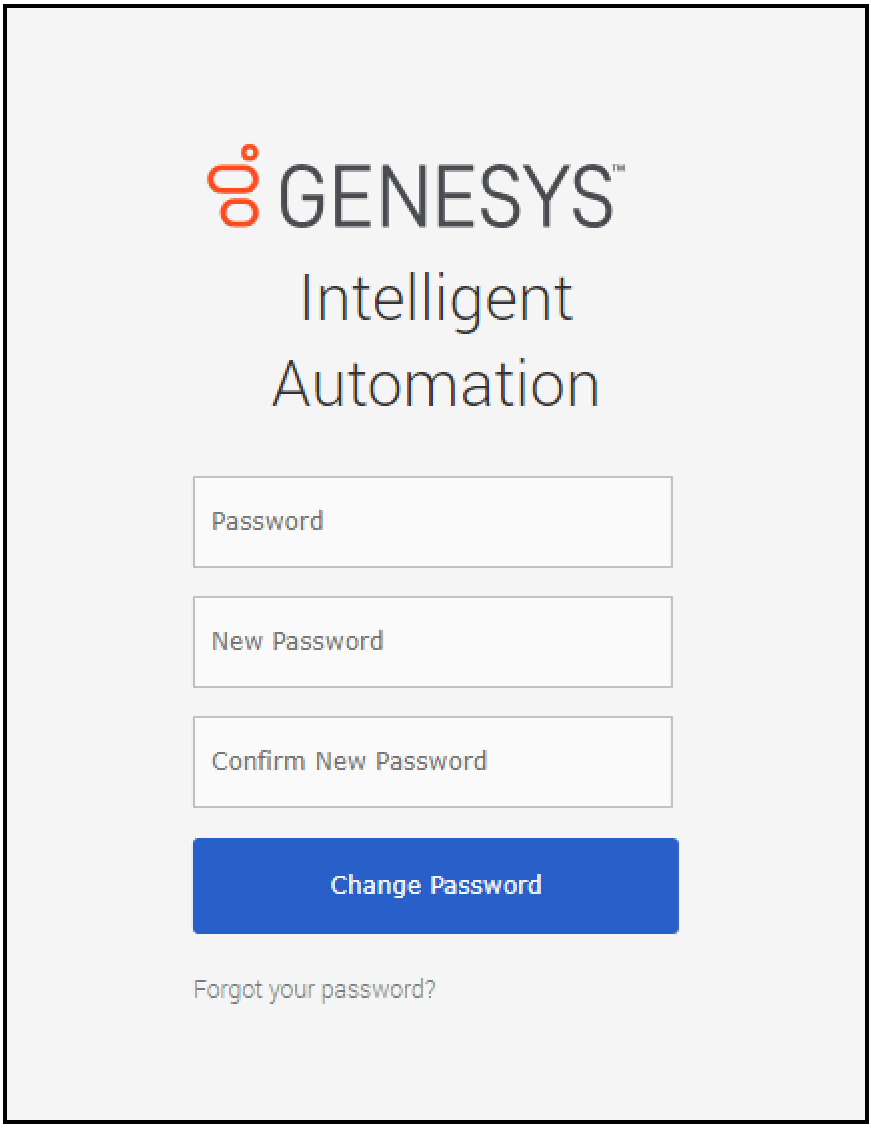
Tip

The email will be sent to the email address you use to log in. If you have multiple email accounts and cannot see the email from the system administrator, check that you are in the same email account as your login.

How do I change my password?

You can change your own password arbitrarily from the My Account item that appears in the drop-down menu when you right click your name in the top right-hand corner of the Intelligent Automation screen.

The system can also force you to change your password by displaying the following window when you click Login.



GENESYS Intelligent Automation

Depending on your organization's policies, you might have to change your password in two situations:

- If you are a new user, you might have to change your password after your first login.
- As decided by your system administrator. For example, policy may state that passwords expire every three months. Or you might have forgotten your password and were emailed a link to reset it.

Create a new password following the rules sent to you by your administrator. In the window, enter your original password, then enter your new password. If the new password does not meet the criteria for a valid password, you will be presented with a warning and a list of the criteria. Change the new password to meet the criteria. After you have entered a valid password, the warning does not appear. Re-enter the new password and click **Log in**. You are now logged in.

Log Out

To log out of Intelligent Automation, click your account details in the top right corner of the display, and click **Logout** in the drop-down menu.

Interface Overview

Genesys Intelligent Automation (also referred to as *Intelligent Automation*), provides non-technical users with a high level of control over the management and configuration of the system using a web-based interface. You can use the system Dashboard to see at a glance how your applications are performing. The Dashboard also proactively highlights areas for improvement, therefore avoiding potential usability issues. To further explore the data, you can use the Reports view to drill-down and view reports and customer journeys to enhance your applications and make your contact center more efficient.

Intelligent Automation gives you the ability to dial into both a test and production version of your IVR application. The test version allows you to call in and test your application as soon as you make changes, without affecting callers in the production environment. Once you are satisfied with your changes, the application can be deployed directly to production, and the new callflows are applied to the very next call.

Terminology

The following are key terms used in the Intelligent Automation software and throughout this document:

- Applications, modules and menus
- Blocks
- Paths
- Company

Applications, modules and menus

Applications, *modules*, and *menus* all refer to programs designed to perform a specific task for the caller. However, each application is configured differently and provides different functionality.

Applications

Applications are the IVR programs that execute when customers call your company. Applications hold all the defaults, global commands, standard prompts, and Callflow Preference settings that are inherited by *modules* and *menus*.

Throughout this document and the Intelligent Automation software, *application* refers to the IVR application as a whole, which has been developed by your company to allow its customers to perform self-service tasks.

Each application is configured using a callflow diagram, which depicts each stage a caller will encounter during the call. This diagram can incorporate modules and menus depending on the functionality required.

Modules

Modules are prebuilt callflows that include out-of-the-box prompt wording, built-in error handling (at both a question and module level), and module-specific configuration screens. These screens give you the flexibility to configure your applications for your specific needs.

Multiple modules are available for different call types within each industry sector (for example, mobile operators may use the *Pay-as-you-go Top Up* module).

These modules are provided in the Intelligent Automation installation if they have been purchased by your company.

Menus

Menu modules provide you with a blank canvas and a set of predefined blocks so that you can build intelligent call-steering applications. They also enable you to piece together prebuilt modules to create a fully-fledged IVR application with the functionality you require.

> Tip
>
> - *Applications* contain global behaviour, such as the handling of the *agent* command from the caller, and they are what callers first encounter after they dial your phone number.
> - *Modules* are prebuilt callflows equipped with handling a particular business process, such as the capturing of credit card details for payment over the phone.
> - *Menus* are the modules that glue everything together to provide flexibility and a seamless customer experience.

Blocks

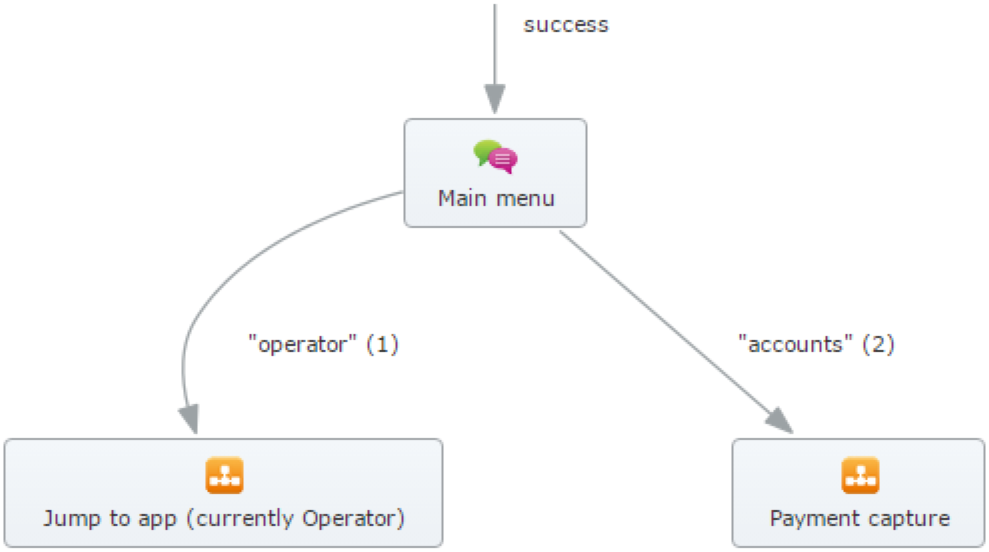
Each individual step in an application callflow is known as a *block*. Each block is represented by a rectangle on the callflow diagram in the Callflow Editor page.

Blocks perform a single basic function. For example, a Phone block allows you to set up the transfer of a caller to a specific number depending on the time of day when the call was placed.

Paths

The term *path* refers to the route that leads from one block to the next. A path is depicted as a line that links the blocks in a callflow together, but it also represents the outcome of the last block (for example, **success**).

In a callflow diagram, each path is accompanied by a label that explains the outcome of the previous block.



In the example above, the caller must select "accounts" in the *Main menu* block to progress to the payment capture module, or the caller must select "operator" to progress to the operator module. Menu options are surrounded by quotation marks because they are presented to the caller as options in the *Main menu* block. Path names that do not use quotation marks denote a result of an action by the caller (for example, *success*).

If a block has no path leading from it, the block's result is returned to the parent application. For example, consider the following scenario:

- *Module A* calls *Module B* via a Link block. This **Link** block has been configured to have two possible paths:
  - **success** - Links to a Message block with the prompt "Success."
  - **failure** - Links to a Message block with the prompt "Failure."
- The last step in *Module B* is a Script block that has no path coming from it. The script performs logic and returns either **success** or **failure**.
- When the **Script** block returns **success**, the caller hears the prompt "Success." When the **Script** block returns **failure**, the caller hears the prompt "Failure."

An error results if a Script block returns a result that does not have a configured path, either in its global defaults or in the parent module.

Company

All Intelligent Automation installations have a default *Templates* company inside which all prebuilt modules are loaded. Administrators can create another company under which they can create their own custom applications and modules. In other words, a company serves as the repository for applications and modules.

Intelligent Automation is set up to support multiple companies in the same installation. Each company can be administered separately but share applications with others.

Dashboard

The **Dashboard** view offers a simple view into the current status of your applications. The main areas of the **Dashboard** view are:

- Application Overview
- Chart View
- Quick Links toolbar
- System Pulse

Application Overview

The **Application Overview** section presents historical and real-time data for your organization’s Key Performance Indicators (KPIs) for all of your applications. It provides an instant view of IVR performance against seven key metrics for the current day, previous seven days, and previous 30 days. You can click each metric to view more details in Chart View.

The first five metrics are:

- **Total Calls** – The total number of calls handled by applications for the period.
- **Average Calls** – The average number of calls handled per day for the period.
- **Planned Transfer Outcome** – The percentage of calls transferred to customer service agents as a normal part of the callflow.
- **Unplanned Transfer Outcome** – The percentage of calls transferred to customer service agents when the customer failed to reach their desired destination (for example, because of recognition failure or error).
- **Hang-up Outcome** – The number of calls in which the caller ended the call.

The remaining two metrics make use of a unique Intelligent Automation feature called a *business task*. A business task executes when a caller starts along a particular path on the callflow to, for example, request a balance, report a fault, or pay a bill. These tasks are typically controlled using one of the prebuilt applications. If the task completes successfully, it returns to the main application with the result **success**; otherwise, it returns the result **failure**. Tracking IVR activity in this way provides meaningful information to the business about the application's performance without the need for extensive analysis of individual responses to prompts. In the Dashboard view, this performance is represented as:

- **Total Tasks Attempted** – The number of business tasks (as defined in the callflow) started by callers.
- **Task Success Rate** – The percentage of business tasks completed.

Chart View

This section is at the bottom of the **Dashboard** view. The chart displayed depends on the metric selected from the Application Overview. The chart provides a snapshot of calls over the previous 30 days, enabling you to identify trends as they emerge. This chart is updated every time you log into Intelligent Automation.

Quick Links toolbar

The Quick Links toolbar has buttons that link to other areas of the system, including:

- Active Modules – View all the applications, modules, and business processes in your environment.
- Deployment Status – View the current deployment status of each module. You can also click Advanced Details in the **Active Modules** view to display the same information.
- Solution Overview – View a graphical representation of the callflow for each application. You can also click Show Graphical View in the **Active Modules** view to display the same graphical view.

System Pulse

This panel displays a performance indicator for each application. This indicator provides a quick insight into how well a particular application is performing in terms of usability and call completion.

System Pulse lists the modules within the application that are currently experiencing problems. It indicates where the issue resides within the module, allowing you to quickly access the area and make the necessary changes.

The health indicator shows the health of the application callflow and each of its linked modules. In other words, application health degrades if the application is performing successfully but one of the modules that it links to is having usability issues.

> ### Tip
> Use the Graphical View to see the health for each module in isolation.

The table below describes the System Pulse indicators:

Indicator	Description
💚	The application, and all modules linked to it, are currently in a healthy state and are performing within predefined thresholds. It does not require any attention.

Indicator	Description
🧡	Indicates that the performance of an application, or modules linked to it, is approaching predefined thresholds. May require attention.
💔	The performance of an application, or modules linked to it, is falling outside predefined thresholds and requires urgent attention. A list of specific areas that require attention is displayed below the application. These issues must be resolved to return the application or module to a healthy state. Click a specific area to access configuration settings.

The status of each application and module is queried for the current company when you log in, using data from the last 30 days. The status refreshes each time you visit the **Dashboard** view.

Applications

The **Applications** view displays the following:

- **Menus** - Applications and menus that you have created.
- **Security Modules** – Prebuilt modules that provide you with control over the security section of your application callflows.
- **Business Processes** – Prebuilt modules that automate business procedures, such as conducting account balance enquiries.
- **Utility Modules** – Prebuilt modules that all users can use, such as selecting accounts.

Prebuilt modules can be added to your main application. Blocks cannot be added or removed from these prebuilt modules, but specific options can be changed (for example, wording of prompts).

> Important
>
> Contact your Genesys representative if you need to update callflows for Security Modules, Business Processes, or Utility Modules.

Click System Pulse in the toolbar to view the current status of each application. You can click a red or amber heart to view more information on where the issue resides within the application callflow. Additionally, you can click an issue to drill down and identify the root cause of the issue (for example, voice-recognition errors, hangups, or unplanned transfers).

The **Business Processes** section uses the following icons:

Icon	Description
⑂	This module calls into one or more other modules.
⚠	A link in the callflow does not link to a valid module.
🔒	Blocks cannot be added or removed from this callflow. Important This icon is only visible when **Advanced Details** is active in the **Applications** view.

Advanced Details

Click **Advanced Details** to view the deployment status of applications and modules.

The following icons denote deployment status:

Icon	Description
🟩	Successfully deployed.
🟧	A change has been made to the application or module recently and has not yet been deployed.
🟥	A change has been made to the application or module, and a significant amount of time has elapsed since this change.

You can perform the following actions from this section:

- Remove an application or module by clicking delete.
- Deploy an application or module by clicking **deploy now**. This display the Deploy to Production tab.
- Click an application or module to open the Callflow Editor.
- View the current status of each application or module by clicking System Pulse in the toolbar.
- View the Site ID for each application.

Show Graphical View

Click **Show Graphical View** to see a high-level graphical illustration of the application callflow. In particular, this view shows the modules used in the application callflow and the paths that lead to these modules. Applications are represented by green boxes and modules by gray boxes.

To view a different application, select it in the **Choose Application** drop-down menu.

You can perform the following actions from this section:

- Click System Pulse to view the current health of each module within the application callflow. This displays a heart icon in the top-left corner of each module.
- Click Advanced Details to view the deployment status of each module. This displays a flag in the bottom-left corner of each module.
- Click an application or module to view or edit it. This displays either the **Settings** tab (if you clicked a prebuilt module) or Callflow Editor (if you clicked an application).
- Depending on your user privileges, a **Create New Module** button might also be displayed in the **Shortcuts** toolbar.

Callflow Editor

Tip

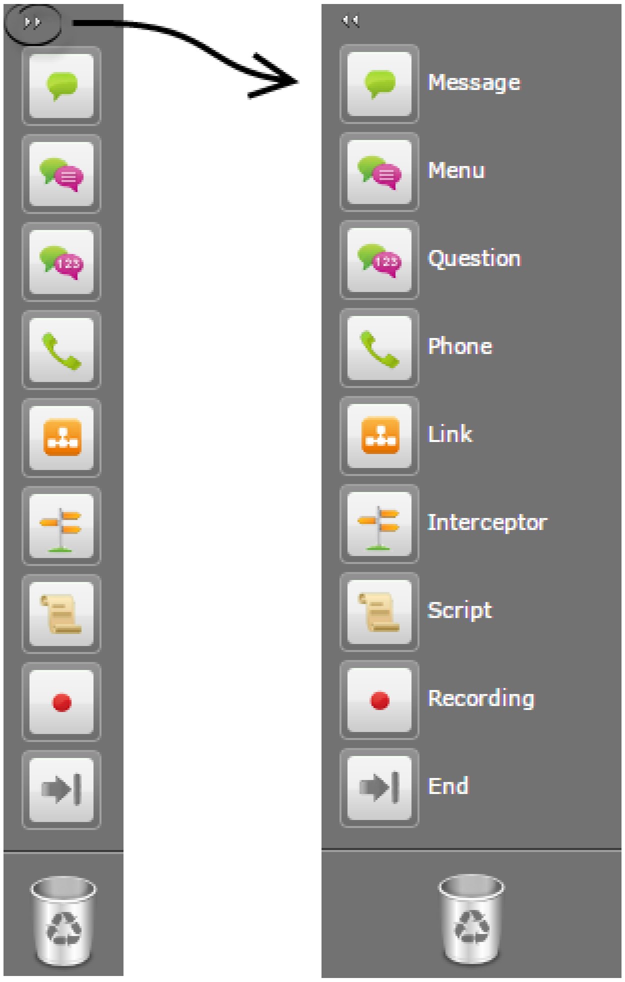
See the Using the Callflow Editor page for more information.

You can access the Callflow Editor by clicking an application or module in either the Applications view or the Show Graphical View.

The Callflow Editor is a visual drag-and-drop designer that allows you to design and configure new applications/menus, and maintain existing ones.

Each application is made up of several blocks, each linked by a specific path. Each block represents a step that the caller goes through during a call, such as a Menu block, a Question block, or a Link block. This enables you to understand the steps that a caller has gone through, as well as easily interpret reporting.

The different blocks you can add to a callflow are listed on the left side of the Callflow Editor in the toolbox. The names of the blocks in the toolbox can be displayed or hidden, as shown below:



The Callflow Editor has four other tabs

- Prompt List
- Application/Module Details
- Deploy to Production
- Opening Hours

You might also see product-specific tabs, depending on which module you are currently editing.

The callflow of a product-specific module cannot be changed (for example, you cannot add or remove blocks), but specific options can be changed, such as the wording of prompts.

Integration

The Integration view is also the Integration Hub (commonly referred to as *iHub*) interface, in which you create the necessary processes and configuration to integrate Genesys Intelligent Automation with third-party resources, connecting the Intelligent Automation callflow engine to a customer's back-end web services and databases.

Before iHub, custom-defined web service *wrappers* were used to bridge the gap between Genesys Intelligent Automation and the user's system. These wrappers took considerable time and knowledge, and created separate code artifacts that needed to be supported and maintained throughout the lifetime of the project. iHub provides the interface for a user to easily create these wrappers, called Processes, while looking after internal details itself. The Process is then incorporated into callflows through the Interceptor block.

This section provides an overall description of iHub and its integration capabilities, and the terminology used with it. To use iHub to integrate Intelligent Automation and its customers, refer to Integrating Intelligent Automation and Customer Environments.

Benefits of iHub

iHub offers the following benefits when integrating systems:

- Provides an easy-to-use interface for an Author to create web service wrappers for Intelligent Automation with minimal effort.
- Frees the Author from operational concerns with the deployment and maintenance of integration components.
- Frees the Author from essential cross-cutting concerns, such as logging, performance, security, and so on.
- Enables integration with the rest of the Intelligent Automation product family.

Limitations

iHub does have the following limitations:

- HTTP(S) connection pools apply to the whole environment, and are not configurable on a per-Process basis. You can designate a pool to be used only by one Process, but that pool is not officially linked to the Process. That is, there is nothing other than your good intentions preventing the pool from being used by another Process.
- SQL connection pool settings apply to the whole environment, and are not configurable on a per-Process basis.
- There is no access to integration server log files from the iHub interface.
- There are no built-in editors for REST, SOAP, or SQL queries. SQL queries are supported, but in the standard Process editor and using API methods.
- There are no prewritten (public) processes or templates.

- No specific reports are created for Integration Processes.
- There is no debugging available for Integration Processes.
- Users cannot import JAR files.
- Integration Scripts cannot use SNMP for alerts.

Terminology

In addition to the terminology used throughout Intelligent Automation, the following terms are used when discussing integration and iHub:

- Author
- Process
- Library
- Environment

Author

An Author is someone who uses iHub to integrate Intelligent Automation with the customer's back-end. It could be a member of the Genesys Professional Services team, or it could be a software developer in the customer’s enterprise. The Author must know how to write software code. The Author must also understand data contracts for Intelligent Automation products, and for the customer's own back-end and downstream databases, web services, and so on.

Process

An Integration Process is essentially a script that is executed upon receiving an HTTP(s) request from the Intelligent Automation VUI and eventually returns an XML response.

Integration Script

What an Integration Script does is entirely up to the Author. It might send a request to a downstream web service to get a list of accounts; perform some business logic; find a customer record in a database; send an email; or all of the above. Authors can write the Scripts in Groovy or JavaScript. As with callflow Script Blocks, this piece of code will run within the constraints of a Java Security Manager to limit its influence on the rest of the process/server.

iHub provides an API of helper methods to expose useful functionality such as:

- HTTP requests (including raw SOAP XML)
- Logging
- Formatting for dates and currency amounts
- Escaping functions (XML/JS/URL)

- Parsing JSON and XML
- Getting/setting variables
- Accessing request body
- Selecting a response XML template
- Caching arbitrary data

For a list of the helper methods, see Scripting Commands.

Response Template

When the Script has finished executing, the resulting variables are placed into an XML (or JSON) Response Template, and the resulting text is sent back to the VUI that originally called the Process. A Process can have only one Script, but multiple Response Templates. The Response Template looks after correctly escaping any values (a common error when coding manually), and can include logic (such as looping and conditional items), if required.

Process Endpoint

A Process can be triggered by an external force (for example, a VUI callflow Script block) by making a request to a Process Endpoint. If the iHub server is already known, the Process Endpoint will take the form of an HTTP or HTTPS URL, with server hostnames and port number specified in the URL. For example:

```
https://myserver:8080/fish-integration/go/company/1/process/{36310e3b-5ae9-43bd-b95e-f9fef327aa33}
```

If the iHub server is not known (for example, when storing details of a Process Endpoint within a callflow block), the Process Endpoint could instead take the form of:

```
ihub://{36310e3b-5ae9-43bd-b95e-f9fef327aa33}/My%20Process%20Name
```

In this case, the `ihub://` protocol notation is replaced at runtime by `https:// protocol`, as above, complete with server hostnames and port numbers.

Library

The Library is a set of reusable components and scripts that are made available to all Processes in a Company. The main component of a Library is a shared Script for each language, created using the Groovy editor and/or JavaScript.

Environment

In iHub, the term *Environment* refers to the environment that you are integrating, that is, a single Intelligent Automation Company and its own iHub setup with its own pool of HTTP(S) connections, JDBC data sources, and Environment settings. It represents Company-specific configuration information that is expected to be used with a given Intelligent Automation callflow engine installation. This would typically include HTTP connection pool settings (including security settings such as client-side certificates), and any other configuration settings generic to the Company.

You cannot export environment settings, including:

- JDBC data sources
- HTTP connection pool settings (including security settings such as client-side certificates)
- Any other generic configuration settings

Reports

The **Reports** view displays various data about your contact center, including:

- Online Reports - Statistics about how your contact center is operating, including the number of inbound calls, business task success rate, recognition accuracy, and much more.
- Customer Journeys - Detailed maps of how customers proceed through your callflow(s).
- Raw Data - Detailed information about each call recorded in the database.

Online Reports

You can use the **View Online Reports** tab to generate various reports on the operation of your contact center. The table below describes details about the standard reports included with Intelligent Automation. For more detailed information, see the Report types section below.

Type	Description	Example Use Cases
Summary	Shows high-level statistics on call volumes, business tasks, and input recognition.	• How many calls went through an application in October? • What percentage of calls originated from mobile devices? • What is the average call duration compared to last month? • Following a prompt-wording change, what percentage of people successfully went through the Identification and Verification module last month compared to the month before? • Which menus are giving the highest breakouts?
Calls per Day	Shows the total calls per day for a module or application.	How many calls did we receive on the 27th of December following the Christmas period?
Calls by Time of Day	Shows total calls per hour for a module or application.	• What is our busiest hour? • How many staff do we need?

Type	Description	Example Use Cases
Block Results	Shows the volume of callers and results for each callflow step in the IVR application.	• How many callers hung up at the welcome prompt? • How many times were callers not found after entering their account number? • How many callers had trouble answering a particular question? • Of all callers that reached a particular call step, what percentage hung up?
Recognition Summary	Shows detailed recognition statistics for each question in the callflow including, percent recognized, percent unsuccessful, average duration and average retries.	• How many callers hit the Get Identifier question? • Which question has the lowest recognition rate? • Where do most people hang up? • Which question is taking the longest to answer? • Which question requires the most retries?
Business Task Summary	Shows the number and percentage of business tasks completed throughout the IVR application.	• How many callers successfully passed through Identification and Verification? • What percentage of people did not make it through Identification and Verification? • What is the average call duration for someone to make a payment?

Generating a report

To generate a report, configure the following:

- **Report Type** - Select one of the report types (described in the table above).
- **Start Date** - Select the start date for reporting data.
- **End Date** - Select the end date for reporting data.
- **Call type** - Select whether to use test-version data or production-version data.

- **Application** - Select whether to use all applications, or a specific application, for reporting data.
- **Report Options** - This section only appears if you choose the **Block Results** or **Recognition Summary** report type.
  - **Include All Modules** - If checked, the report includes data from all modules. If not checked, you can choose one or more modules to include in the report.
  - **Show Tagged Blocks Only** - For the **Blocks Results** report only. If you have tagged (filtered) certain blocks previously, you can select this box to use this tagging schema again. Otherwise, do not select this box. See the Block Results section for more information.
- **Report Format** - Select whether to view the report on-screen or to download a CSV file.

Click **Generate Report**.

Report types

This following table provides detailed information about each report.

Summary

Section	Statistic	Description
Calls	Number of calls	Total number of inbound calls.
	Average calls per day	Average number of inbound calls for the date range.
	% were mobile calls	Percentage of inbound calls from mobile devices.
	Average call duration	Average call duration for all inbound calls in the IVR application.
Business Tasks	% Task success rate	Percentage of callers who completed a business task (for example, making a payment). This statistic includes outcomes such as card declined, as the underlying product is still behaving according to specification.
	% Caller perceived success rate	Percentage of callers who completed a business task (for example, making a payment). This statistic only includes outcomes in which the caller was successful (for example, the card was not declined).
	Most frequent tasks	Shows the three most-frequent tasks.
Recognition	Average timeouts/retries	Average number of attempts to get a successful recognition. For example, if the statistic is **0.069**, then on average it takes callers

Section	Statistic	Description
		1.069 attempts to complete a question.
	Average recovery attempts	Average number of attempts to use the recovery default path.
	Last menu	Shows the three most-frequent menus where someone has exited the application.

Calls per Day

Section	Statistic	Description
Calls by Date	Date	Date the calls occurred.
	Number of Calls	Number of inbound calls received.

Calls by Time of Day

Section	Statistic	Description
Calls by Hour	Time	The hour in which the calls occurred.
	Number of Calls	Number of inbound calls received during the hour.

Block Results

The report shows one or more sections, depending on the number of modules and business processes available in your company, and whether you selected to view specific applications when you generated the report.

You can also create tags to filter which blocks appear in the report.

- Create a tag: In the **Report Options** section, click the link to create a tag. A pop-up window appears in which you can select which blocks to display in the report. After you select one or more blocks, enter a name for this tag in the **Tag Name** field and click **Save** at the bottom of the block list. To create another tag, you must ensure a tag is currently applied to the report before the link appears to create a tag.
  Link to video
- Delete a tag: In the **Report Options** section, apply the tag you want to delete. Next, click the **tag** link to open the tag's properties. In the pop-up window, click the link to delete the tag.
  Link to video

Statistic	Description
Block	The name of a block within this module or product.
Result	The result of the block. Each block can have different results.
Count	Number of calls that encountered this result for this block.

Statistic	Description
% of All Visits to This Block	Percentage of visits to this block that encountered this result.

Recognition Summary

The report shows one or more sections, depending on the number of modules and business processes available in your company, and whether you selected to view specific applications when you generated the report.

Statistic	Description
Block	The name of a block within this module or product.
Number of Visits	Number of times this input block has been entered.
% Recognised	Percentage of callers who successfully passed this input block by using either DTMF or voice.
% Hangups	Percentage of callers who hung up during this block.
% Max Retries	Percentage of callers who had trouble providing a response at this input block and received the maximum number of retries.
% Max Timeouts	Percentage of callers who did not respond to the question for this block and received the maximum number of attempts.
% Other	Percentage of callers who encountered another event at this block, such as a platform error due to licensing.
Ave. Duration (secs)	Average duration that a caller spends on this input block.
Ave. Retries	Average number of retry attempts that it takes for a caller to respond successfully to this input block.
Ave. Timeouts	Average number of timeout attempts that it takes for a caller to respond to this input block.
Ave. Helps	Average number of help requests for this input block.
Ave. Recovery Attempts	Average number of recovery attempts for this input block.

Business Task Summary

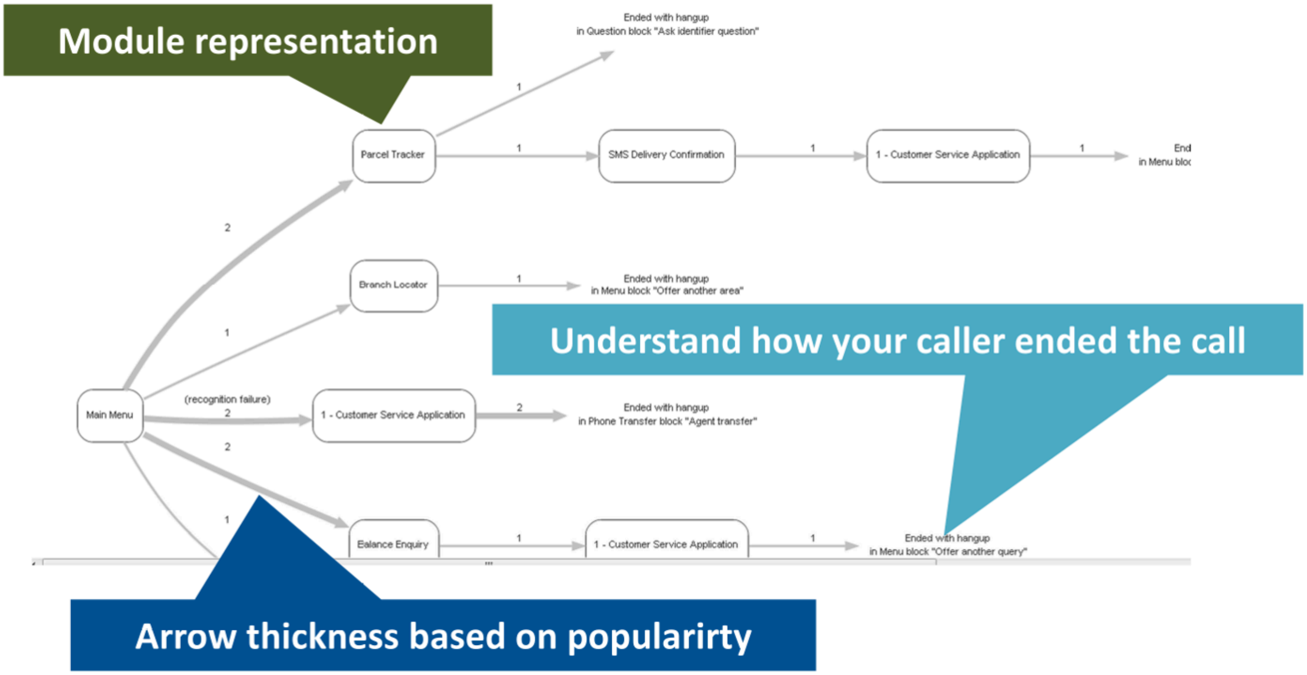
The report shows one or more sections, depending on the number of modules and business processes available in your company, and whether you selected to view specific applications when you generated the report.

Statistic	Description
Outcome Category	Type of outcome for this business task.
Outcome Description	The event generated by this outcome.
Count	The number of times this event has been

Statistic	Description
	generated by this outcome.
% of All Attempts	The percentage of attempts that resulted in this event for this outcome.
Ave. Duration (secs)	The average duration spent on achieving this event for this outcome.

Customer Journeys

You can use the **View Customer Journeys** tab to view easy-to-understand graphical representations of how callers progress through your applications.


Module representation
Ended with hangup
in Question block "Ask identifier question"
1
Parcel Tracker
1
SMS Delivery Confirmation
1
1 - Customer Service Application
1
End
in Menu bloc
2
Branch Locator
1
Ended with hangup
in Menu block "Offer another area"
Understand how your caller ended the call
1
Main Menu
(recognition failure)
2
1 - Customer Service Application
2
Ended with hangup
in Phone Transfer block "Agent transfer"
2
1
Balance Enquiry
1
1 - Customer Service Application
1
Ended with hangup
in Menu block "Offer another query"
Arrow thickness based on popularirty

Viewing customer journeys offers various benefits to your business, such as:

- Tracking the most-popular paths through an application at particular points in time to better understand the impact of business changes.
- Rearranging call-steering or self-service options based on usage, to ensure your applications present the most-popular options first to reduce the cost to serve.
- Determine up-sell and cross-sell opportunities based on customer behavior.
- Easily compare the customer experience between two versions of the same application to see which is more efficient.

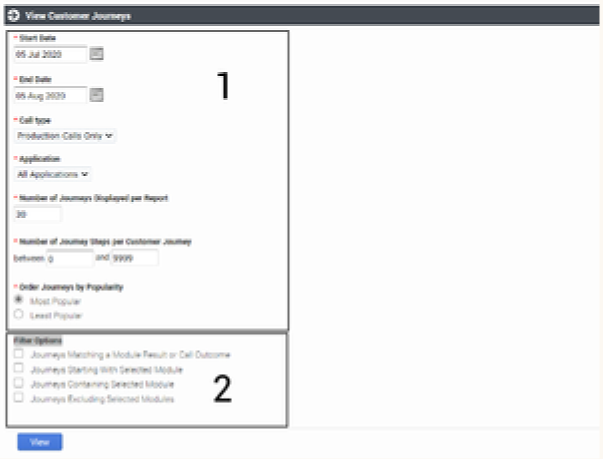
- Monitor the impact of self-service or call-steering changes.
- Make improvements to other parts of the business based on consumer behavior.

Viewing customer journeys

Important

The journey report is a two-step process.

- Intelligent Automation generates a list of journeys based on the **Number of Journeys Displayed per Report** option and the rules except the **Filter options**.
- The **Filter options** are applied to that list to generate the final report.



View Customer Journeys

* Start Date: 05 Jul 2020

* End Date: 05 Aug 2020

* Call type: Production Calls Only

* Application: All Applications

* Number of Journeys Displayed per Report: 20

* Number of Journey Steps per Customer Journey: between 0 and 9999

* Order Journeys by Popularity
● Most Popular
○ Least Popular

Filter Options
□ Journeys Matching a Module Result or Call Outcome
□ Journeys Starting With Selected Module
□ Journeys Containing Selected Module
□ Journeys Excluding Selected Modules

To view customer journeys, configure the following:

- **Start Date** - Select the start date for the journey report.
- **End Date** - Select the end date for the journey report.
- **Call type** - Select whether to to view journeys from test-version applications or production-version applications.
- **Application** - Select whether to use all applications, or a specific application, for the journey report.
- **Number of Journeys Displayed per Report** - Specify how many journeys to view per journey report.
- **Number of journey steps per customer journey** - Specify the minimum and maximum number of steps a journey must use to be included in this journey report.
- **Order journeys by popularity** - Select whether to view the most-popular journeys first, or the least-popular journeys first.
- **Filter Options** - Select a filter.
  - **Journeys Matching a Module Result or Call Outcome** - Journeys must match the specified

result or outcome to be included in this journeys report.

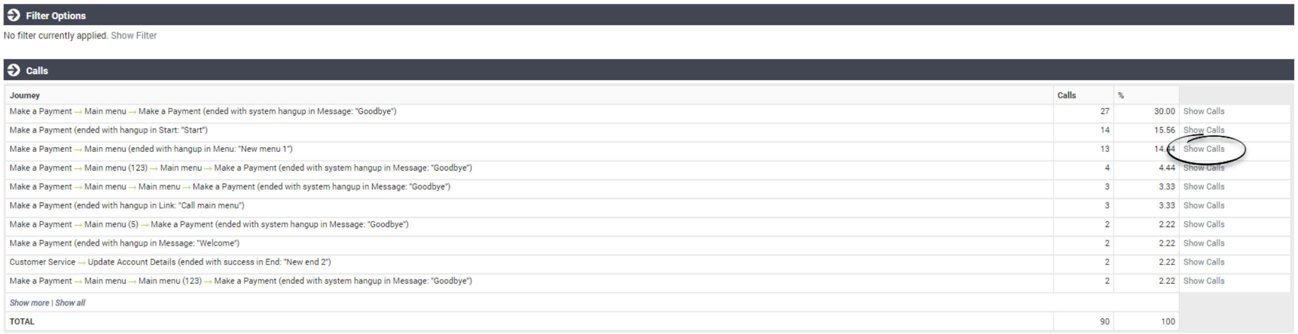
- **Journeys Starting With Selected Module** - Journeys must start with the specified module to be included in this journeys report.
- **Journeys Containing Selected Module** - Journeys must contain the specified module to be included in this journeys report.
- **Journeys Excluding Selected Modules** - Journey must not contain the specified module to be included in this journeys report.

Click **View** to view the journeys report.

Viewing calls related to a specific journey

While viewing the journeys report, you can go to the **Calls** table to view a list of journeys associated with this report, along with how many calls generated each specific journey.

To see information about calls related to a specific journey, click **Show Calls**.



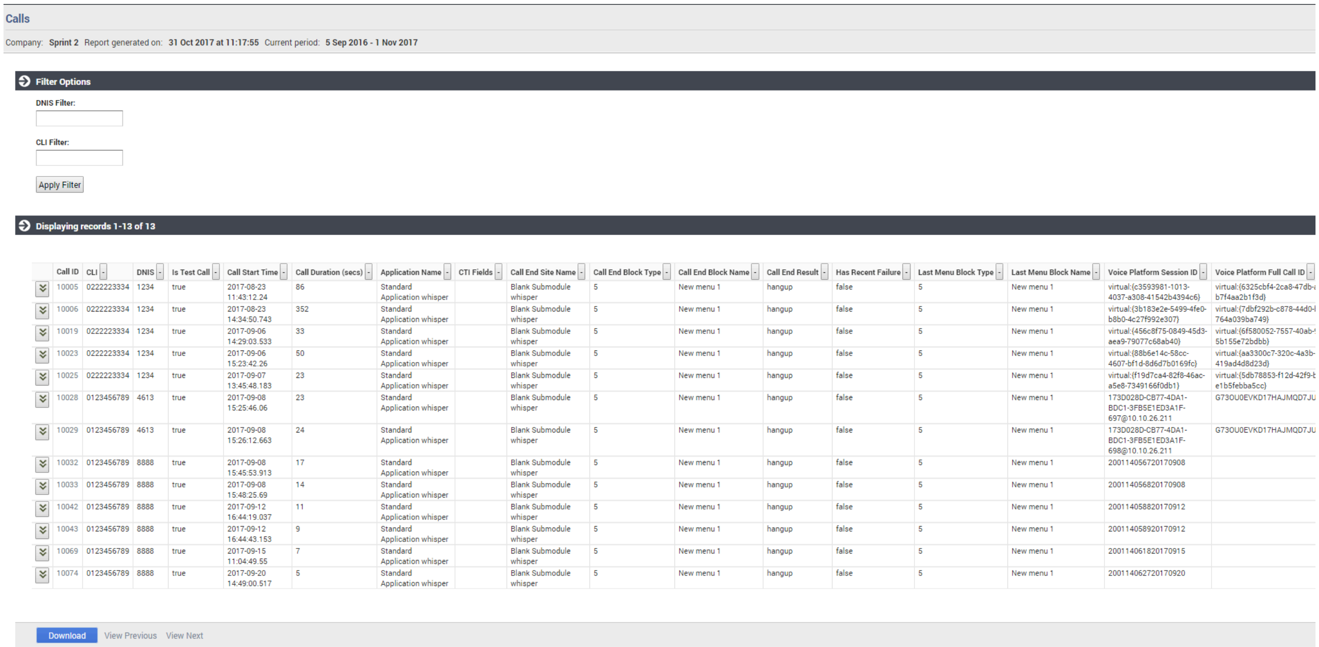
No filter currently applied. Show Filter

Journey	Calls	%	
Make a Payment → Main menu → Make a Payment (ended with system hangup in Message: "Goodbye")	27	30.00	Show Calls
Make a Payment (ended with hangup in Start: "Start")	14	15.56	Show Calls
Make a Payment → Main menu (ended with hangup in Menu: "New menu 1")	13	14.44	Show Calls
Make a Payment → Main menu (123) → Main menu → Make a Payment (ended with system hangup in Message: "Goodbye")	4	4.44	Show Calls
Make a Payment → Main menu → Main menu → Make a Payment (ended with system hangup in Message: "Goodbye")	3	3.33	Show Calls
Make a Payment (ended with hangup in Link: "Call main menu")	3	3.33	Show Calls
Make a Payment → Main menu (5) → Make a Payment (ended with system hangup in Message: "Goodbye")	2	2.22	Show Calls
Make a Payment (ended with hangup in Message: "Welcome")	2	2.22	Show Calls
Customer Service → Update Account Details (ended with success in End: "New end 2")	2	2.22	Show Calls
Make a Payment → Main menu → Main menu (123) → Make a Payment (ended with system hangup in Message: "Goodbye")	2	2.22	Show Calls
Show more | *Show all*			
TOTAL	90	100	

Genesys Intelligent Automation displays a detailed breakdown of each call related to the journey, similar to the information provided in a Raw Data report. You can click the arrows beside each call record to view more detailed information about that particular call.

Raw Data

You can use the **View Raw Data** tab to view detailed data about various aspects of your contact center, including data that are not present in the standard Online Reports.



Calls

Company: **Sprint 2** Report generated on: **31 Oct 2017 at 11:17:55** Current period: **5 Sep 2016 - 1 Nov 2017**

Filter Options

DNIS Filter:

CLI Filter:

Displaying records 1-13 of 13

Call ID	CLI	DNIS	Is Test Call	Call Start Time	Call Duration (secs)	Application Name	CTI Fields	Call End Site Name	Call End Block Type	Call End Block Name	Call End Result	Has Recent Failure	Last Menu Block Type	Last Menu Block Name	Voice Platform Session ID	Voice Platform Full Call ID
10005	0222223334	1234	true	2017-08-23 11:43:12.24	86	Standard Application whisper		Blank Submodule whisper	5	New menu 1	hangup	false	5	New menu 1	virtual:{c3593981-1013-4037-a308-41542b4394c6}	virtual:{6325cbf4-2ca8-47db- b7f4aa2b1f3d}
10006	0222223334	1234	true	2017-08-23 14:34:50.743	352	Standard Application whisper		Blank Submodule whisper	5	New menu 1	hangup	false	5	New menu 1	virtual:{3b183e2e-5499-4fe0-b8b0-4c27f992e307}	virtual:{7dbf292b-c878-44d0- 764a039ba749}
10019	0222223334	1234	true	2017-09-06 14:29:03.533	33	Standard Application whisper		Blank Submodule whisper	5	New menu 1	hangup	false	5	New menu 1	virtual:{456c8f75-0849-45d3-aea9-79077c68ab40}	virtual:{6f580052-7557-40ab- 5b155e72bdbb}
10023	0222223334	1234	true	2017-09-06 15:23:42.26	50	Standard Application whisper		Blank Submodule whisper	5	New menu 1	hangup	false	5	New menu 1	virtual:{88b6e14c-58cc-4607-bf1d-8d6d7b0169fc}	virtual:{aa3300c7-320c-4a3b- 419ad4d8d23d}
10025	0222223334	1234	true	2017-09-07 13:45:48.183	23	Standard Application whisper		Blank Submodule whisper	5	New menu 1	hangup	false	5	New menu 1	virtual:{f19d7ca4-82f8-46ac-a5e8-7349166f0db1}	virtual:{5db78853-f12d-42f9- e1b5febba5cc}
10028	0123456789	4613	true	2017-09-08 15:25:46.06	23	Standard Application whisper		Blank Submodule whisper	5	New menu 1	hangup	false	5	New menu 1	173D028D-CB77-4DA1-BDC1-3FB5E1ED3A1F-697@10.10.26.211	G73OU0EVKD17HAJMQD7JU
10029	0123456789	4613	true	2017-09-08 15:26:12.663	24	Standard Application whisper		Blank Submodule whisper	5	New menu 1	hangup	false	5	New menu 1	173D028D-CB77-4DA1-BDC1-3FB5E1ED3A1F-698@10.10.26.211	G73OU0EVKD17HAJMQD7JU
10032	0123456789	8888	true	2017-09-08 15:45:53.913	17	Standard Application whisper		Blank Submodule whisper	5	New menu 1	hangup	false	5	New menu 1	200114056720170908	
10033	0123456789	8888	true	2017-09-08 15:48:25.69	14	Standard Application whisper		Blank Submodule whisper	5	New menu 1	hangup	false	5	New menu 1	200114056820170908	
10042	0123456789	8888	true	2017-09-12 16:44:19.037	11	Standard Application whisper		Blank Submodule whisper	5	New menu 1	hangup	false	5	New menu 1	200114058820170912	
10043	0123456789	8888	true	2017-09-12 16:44:43.153	9	Standard Application whisper		Blank Submodule whisper	5	New menu 1	hangup	false	5	New menu 1	200114058920170912	
10069	0123456789	8888	true	2017-09-15 11:04:49.55	7	Standard Application whisper		Blank Submodule whisper	5	New menu 1	hangup	false	5	New menu 1	200114061820170915	
10074	0123456789	8888	true	2017-09-20 14:49:00.517	5	Standard Application whisper		Blank Submodule whisper	5	New menu 1	hangup	false	5	New menu 1	200114062720170920	

The table below describes the report types.

Type	Description
Call Details	Low-level data based on calls during a particular time period. This report give details on a per-call basis.
Call Steps	Low-level data based on calls steps during a particular time period. This report gives details on a per-block basis.
Business Tasks	Low-level data based on the outcome of business tasks within the IVR. This report gives details on each business task recorded.
GUI Actions	Detailed data on changes made to applications within the Graphical User Interface (GUI). This report provides data such as the person who made the change, what module(s) were changed, and when.

Generating a report

To generate a report, configure the following:

- **Data Set** - Select one of the report types (described in the table above).
- **Start Date and Time** - Select the start date and time to include in reporting data.
- **End Date and Time** - Select the end date and time to include in reporting data.
- **Number of Records Displayed per Page** - Specify the number of records to display on each page when viewing online. Each page provides controls to navigate to another page. This option does not

apply to downloaded reports.

- **Include Header Row** - If checked, the header row is included in the report.
- **Search Calls by CLI** - If checked, only data pertaining to the specified CLI is included in the report.
- **Search Calls by DNIS** - If checked, only data pertaining to the specified DNIS is included in the report.

Click **View Report** to view the report in your browser, or click **Download Report** to download the report as a CSV file.

> ## Important
> Contact your Genesys representative if you have questions about specific fields in a **Raw Data** report.

Personas

The **Personas** view lists information about the personas you have created in your environment.

A persona is a distinct personality you use for your Genesys Intelligent Automation applications. For example, you might create a distinct persona for each language your company serves. as such, you can use one persona for English-speaking customers and another persona for French-speaking customers.

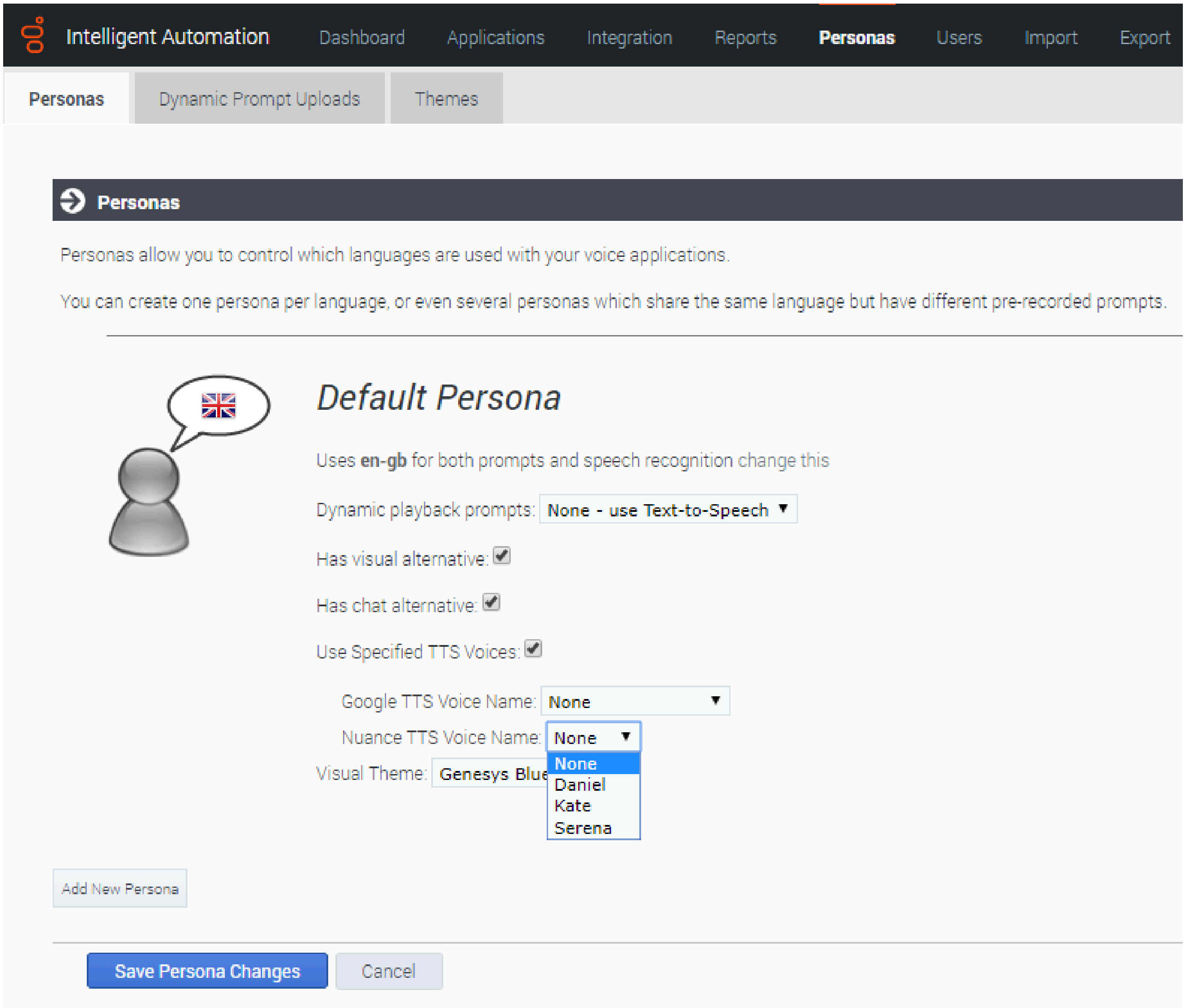
In addition, each persona can use distinct pre-recorded prompts. This is helpful if you want to add distinct personas within a language group to appeal to various subsets of customers. You can have one persona that deals with general English-speaking calls, and another persona that caters to known callers from a particular age group, region, segment level, and more.

You can upload your own dynamic prompts to use with personas as a superior alternative to TTS (Text-to-Speech) prompts.

You can also create visual themes for each persona to use with WebIVR applications.

Personas tab

The **Personas** tab lists the personas you have created. Intelligent Automation comes pre-installed with a default persona, shown below:



Personas allow you to control which languages are used with your voice applications.

You can create one persona per language, or even several personas which share the same language but have different pre-recorded prompts.

Default Persona

Uses **en-gb** for both prompts and speech recognition change this

Dynamic playback prompts: None - use Text-to-Speech

Has visual alternative: ☑

Has chat alternative: ☑

Use Specified TTS Voices: ☑

Google TTS Voice Name: None

Nuance TTS Voice Name: None

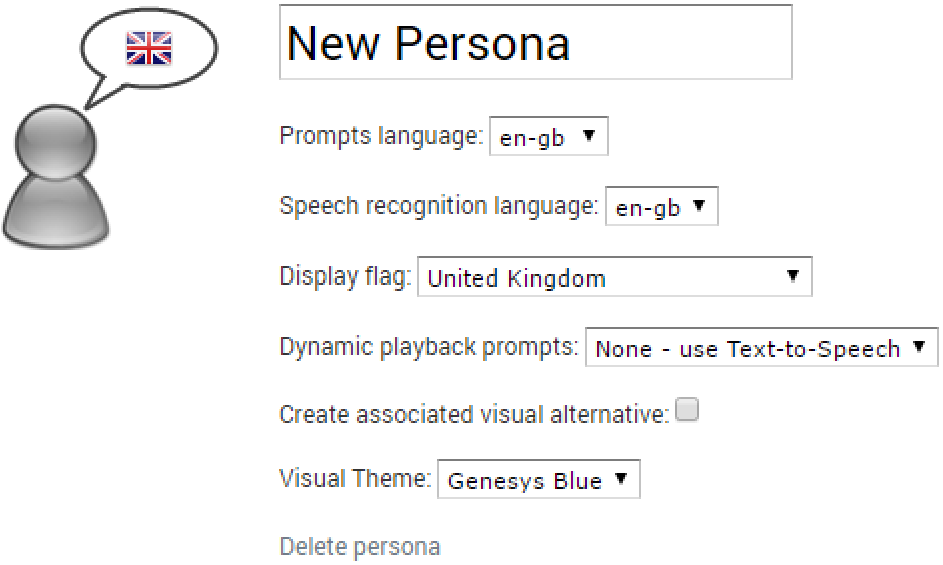
Visual Theme: Genesys Blue

This displays the following information about the persona:

- The name of the persona.
- The language used by the persona (in this case, British English). It also states this language is used for both prompts and speech recognition.
- This persona uses TTS (text-to-speech) for verbalizing information to the caller. However, if you have uploaded dynamic prompts, you can choose the prompt package here.
- Associated Chat and Visual alternatives
- Visual Theme

Creating a persona

Click **Add New Persona** to create a persona for your company. The new persona appears in the list:



New Persona

Prompts language: en-gb ▼

Speech recognition language: en-gb ▼

Display flag: United Kingdom ▼

Dynamic playback prompts: None - use Text-to-Speech ▼

Create associated visual alternative: □

Visual Theme: Genesys Blue ▼

Delete persona

Configure the following:

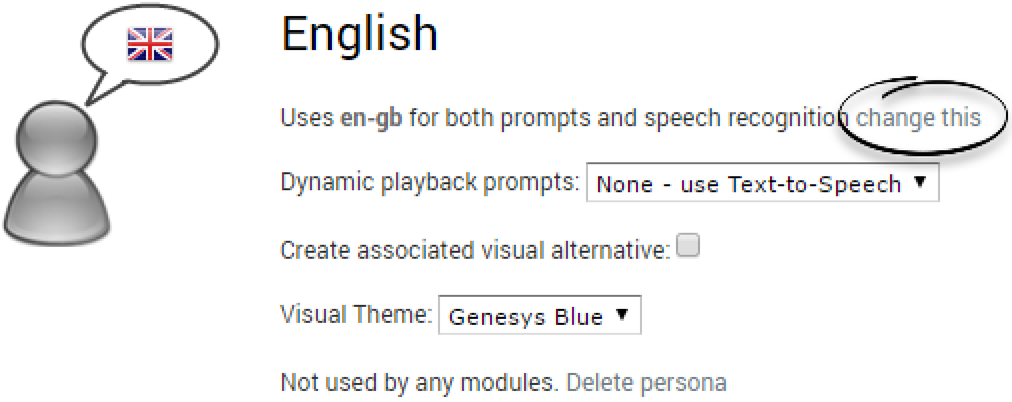
- **New Persona** - The name of the persona. Choose a name that concisely describes the persona's function. In some cases, this might be as simple as a language (**English** or **French**). However, if you want to use more than one persona per language, use a name that describes its purpose (for example, **English - Gold Segment**).
- **Prompts language** - Select the language that this persona uses for TTS prompts.
- **Speech recognition language** - Select the language that this persona uses for speech recognition. This is often the same language you selected for **Prompts language**, but you can choose another language for speech recognition if needed (for example, if non-native speakers are frequently misunderstood by a particular language's speech-recognition engine and you want to use an alternative).
- **Display flag** - Select a flag to identify your persona. This icon is seen in the Callflow Editor for specific blocks, such as Message blocks, that allow you to select a persona.
- **Dynamic playback prompts** - Select whether to use Text-to-Speech or a dynamic prompt package that you previously uploaded.
- **Create associated visual alternative** - If enabled, Intelligent Automation creates a visual persona for use in WebIVR applications.
- **Create associated chat alternative** - If enabled, Intelligent Automation creates a chat persona for use in chat-based applications.
- **Use Specified TTS Voices** - If enabled, Intelligent Automation provides the option to select which TTS voice to use for a supported TTS engine (Nuance or Google). Allowed voices can be configured by an administrator. See Configuring TTS Voices.
- **Visual Theme** - If this persona is used in a WebIVR application, this drop-down menu allows you to

select which theme to use.

Click **Save Persona Changes**.

Editing a persona

Click the **change this** link within a persona to change its details. You can configure any of the fields described in the Creating a new persona section.



English

Uses en-gb for both prompts and speech recognition change this

Dynamic playback prompts: None - use Text-to-Speech ▼

Create associated visual alternative: □

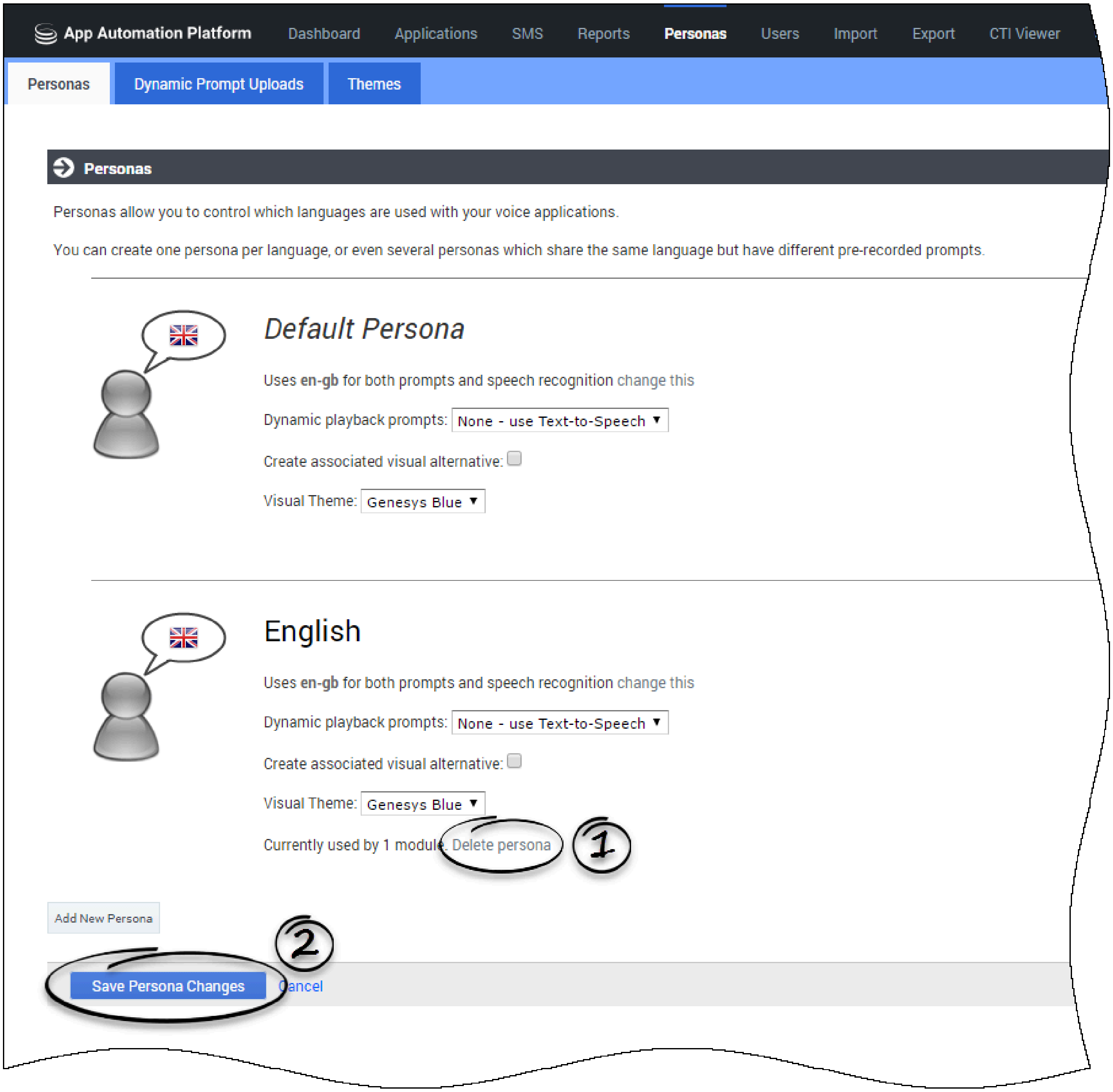
Visual Theme: Genesys Blue ▼

Not used by any modules. Delete persona

Click **Save Persona Changes**.

Deleting a persona

Click the **Delete persona** link within a persona to delete it, then click **Save Persona Changes**.



Personas allow you to control which languages are used with your voice applications.

You can create one persona per language, or even several personas which share the same language but have different pre-recorded prompts.

Default Persona

Uses **en-gb** for both prompts and speech recognition change this

Dynamic playback prompts: None - use Text-to-Speech

Create associated visual alternative: □

Visual Theme: Genesys Blue

English

Uses **en-gb** for both prompts and speech recognition change this

Dynamic playback prompts: None - use Text-to-Speech

Create associated visual alternative: □

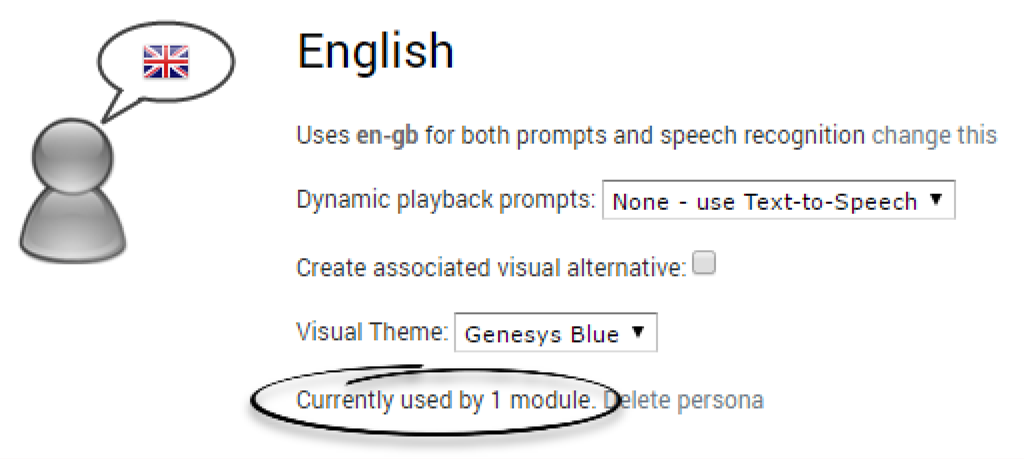
Visual Theme: Genesys Blue

Currently used by 1 module. Delete persona

To the left of the **Delete persona** link, Intelligent Automation states how many applications or modules are currently using this persona. Ensure you understand the risk of deleting a persona that is being used in an application or module.

Warning

If you delete a persona already in use by an application, both the test and production versions of that application will revert back to the default persona. Genesys recommends you switch to another persona in the affected applications before proceeding with deletion. The **Personas** list tells you if the persona is in use by an application.



English

Uses en-gb for both prompts and speech recognition change this

Dynamic playback prompts: None - use Text-to-Speech ▼

Create associated visual alternative: □

Visual Theme: Genesys Blue ▼

Currently used by 1 module. Delete persona

Configure Allowed TTS Voices

You can configure the TTS voices for both Nuance and Google on a per-language basis. In **Administration** > **Default Server Settings**, add a setting for each language and TTS engine you wish to support: Personas.<*TTS Engine*>.<*Language*>.TTSVoiceNames For example:

- Personas.Google.en-gb.TTSVoiceNames = en-GB-Standard-A,en-GB-Standard-B
- Personas.Nuance.en-gb.TTSVoiceNames = Daniel,Serena

Dynamic Prompt Uploads tab

The **Dynamic Prompt Uploads** tab lists the dynamic prompts you have uploaded to your environment.

Intelligent Automation uses dynamic prompts to give applications more natural-sounding language when speaking dynamic information back to customers. For example, when giving a calendar date to a customer, TTS (Text-to-Speech) might sound more uneven ("January One Two Zero One Seven" for January 1, 2017). The TTS voice might also not be in the tone or dialect that your callers expect. However, with dynamic prompts, you can use a native speaker to provide snippets of sounds that Intelligent Automation uses to concatenate more natural-sounding language for callers ("January First Twenty Seventeen" for January 1, 2017).

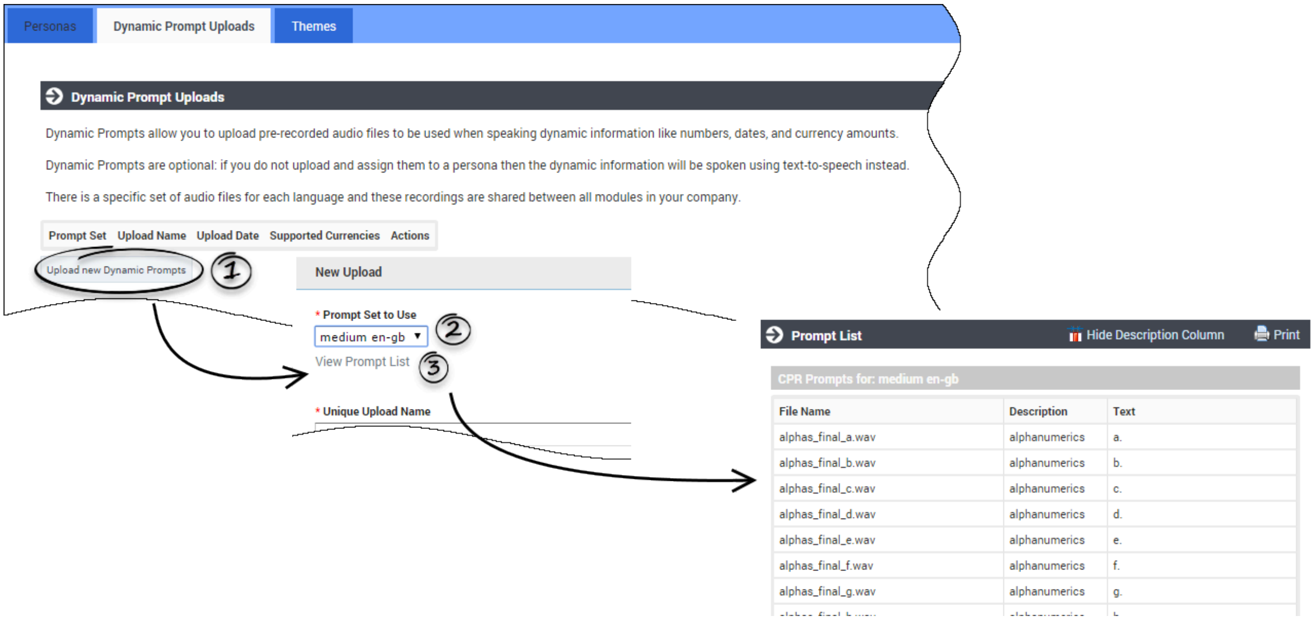
Before you can upload a new dynamic prompts package, you must prepare a ZIP file that contains recordings of the various sounds needed to produce a dynamic prompt. For example, you must have a speaker record sounds of the alphabet, numbers, times, dates, and more. Ensure that:

- The recording package includes all the listed files.
- The files are **.wav** file type.
- The file names must be an **exact** match to the filename provided by Intelligent Automation.

Important

If you are unable to upload the package, set the `Resources.AssumeRawAudioUploads` option (under **Administration** > **Default Server Setting**) to *True*.

To view a list of the required sounds and filenames, click **Upload new Dynamic Prompts**. In the **Prompt Set to Use** menu, select a language (for example, **medium en-gb** for British English with a medium-sized subset of sounds), then click **View Prompt List**.



Dynamic Prompt Uploads

Dynamic Prompts allow you to upload pre-recorded audio files to be used when speaking dynamic information like numbers, dates, and currency amounts.

Dynamic Prompts are optional: if you do not upload and assign them to a persona then the dynamic information will be spoken using text-to-speech instead.

There is a specific set of audio files for each language and these recordings are shared between all modules in your company.

Prompt Set	Upload Name	Upload Date	Supported Currencies	Actions

* Prompt Set to Use: medium en-gb

* Unique Upload Name

Prompt List

CPR Prompts for: medium en-gb

File Name	Description	Text
alphas_final_a.wav	alphanumerics	a.
alphas_final_b.wav	alphanumerics	b.
alphas_final_c.wav	alphanumerics	c.
alphas_final_d.wav	alphanumerics	d.
alphas_final_e.wav	alphanumerics	e.
alphas_final_f.wav	alphanumerics	f.
alphas_final_g.wav	alphanumerics	g.
alphas_final_h.wav	alphanumerics	h.

Warning

The rest of this section assumes you have already configured your environment for supported languages and currencies to be used by your dynamic prompts. If this has

> not been done, go to Default Server Settings to configure your environment before proceeding.

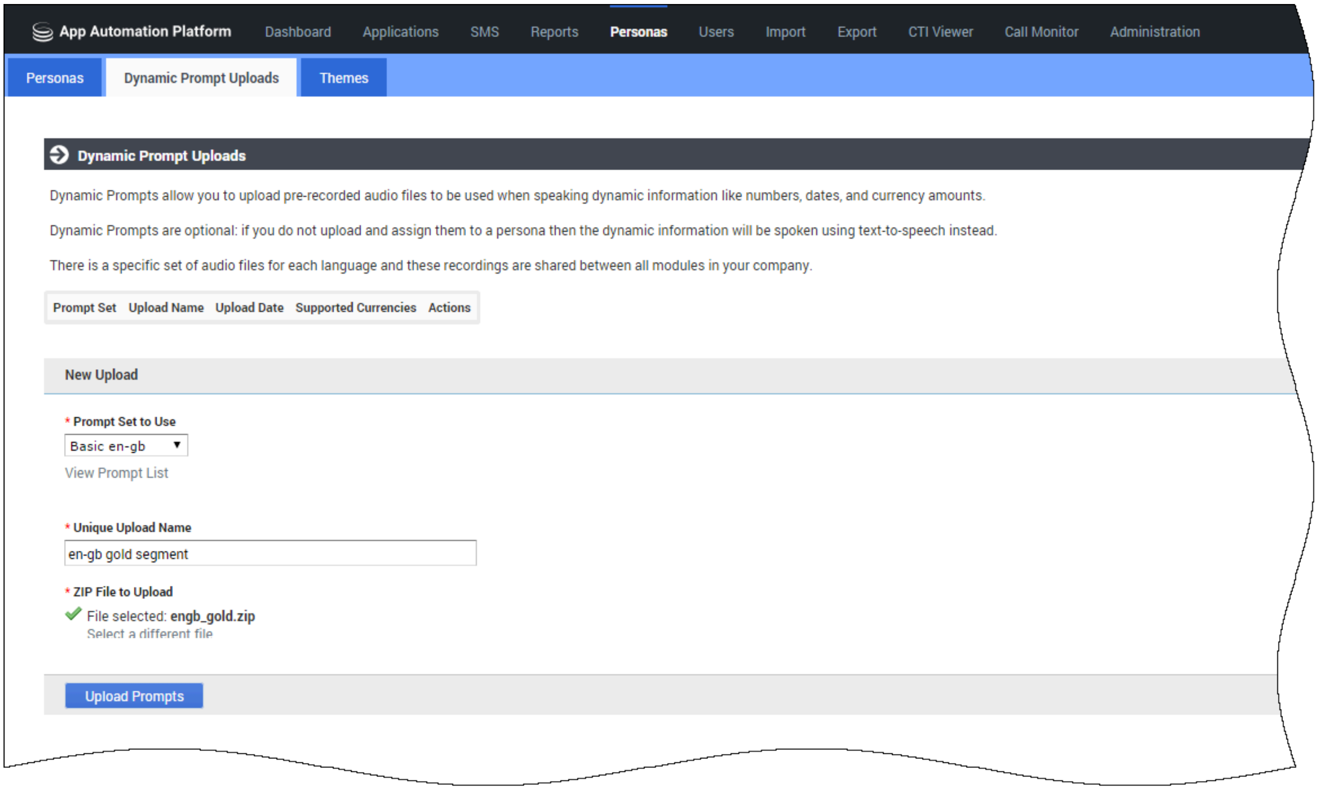
Uploading a new dynamic prompts package

Once the package is ready, click **Upload new Dynamic Prompts**.

In the **Prompt Set to Use** menu, select a language.

In the **Unique Upload Name** field, provide a name for this dynamic prompts package. Choose a descriptive name that describes the purpose of the dynamic prompts.

Click **Choose File** to select the ZIP file on your computer, and then click **Upload Prompts** to upload the file.



Dynamic Prompt Uploads

Dynamic Prompts allow you to upload pre-recorded audio files to be used when speaking dynamic information like numbers, dates, and currency amounts.

Dynamic Prompts are optional: if you do not upload and assign them to a persona then the dynamic information will be spoken using text-to-speech instead.

There is a specific set of audio files for each language and these recordings are shared between all modules in your company.

Prompt Set	Upload Name	Upload Date	Supported Currencies	Actions

New Upload

* Prompt Set to Use: Basic en-gb

View Prompt List

* Unique Upload Name: en-gb gold segment

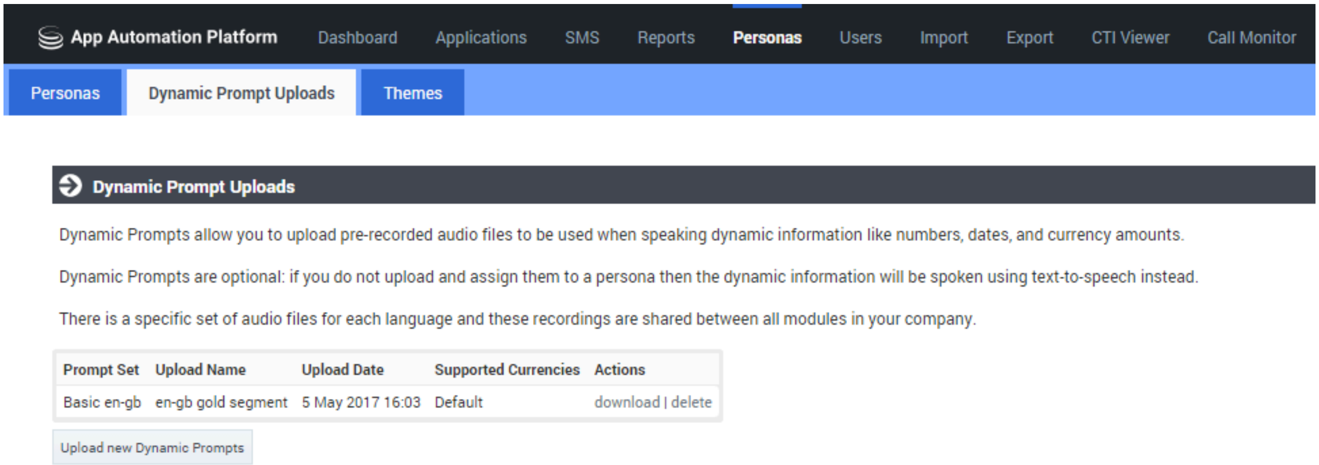
* ZIP File to Upload

✓ File selected: **engb_gold.zip**

Select a different file

Viewing information about your dynamic prompts

Once you have uploaded a package of dynamic prompts, the list updates to show information about the package. For example:



Dynamic Prompts allow you to upload pre-recorded audio files to be used when speaking dynamic information like numbers, dates, and currency amounts.

Dynamic Prompts are optional: if you do not upload and assign them to a persona then the dynamic information will be spoken using text-to-speech instead.

There is a specific set of audio files for each language and these recordings are shared between all modules in your company.

Prompt Set	Upload Name	Upload Date	Supported Currencies	Actions
Basic en-gb	en-gb gold segment	5 May 2017 16:03	Default	download | delete

In the example above, you can view the following information:

- **Prompt Set** - The language set for these prompts.
- **Upload Name** - The name given in the **Unique Upload Name** field when the package was uploaded.
- **Upload Date** - The date the package was uploaded.
- **Supported Currencies** - The currencies this package supports. In this example, it supports this language's default currency. However, you might have a package that supports prompts for euros, pounds, dollars, and more. If so, these currencies are listed in this field.

Downloading a dynamic prompts package

In the **Actions** column, click **download** to download a ZIP file of the dynamic prompts package.

Deleting a dynamic prompts package

In the **Actions** column, click **delete** to delete the dynamic prompts package. Intelligent Automation displays a warning message that states any prompts using this package will revert to TTS (Text-to-Speech). If you understand the warning and agree to the deletion, click **OK**.

Themes tab

The **Themes** tab lists the themes available in your environment for WebIVR applications.

By default, your Intelligent Automation installation comes with the **Genesys Blue** theme. However, you can create your own theme to suit your business needs.

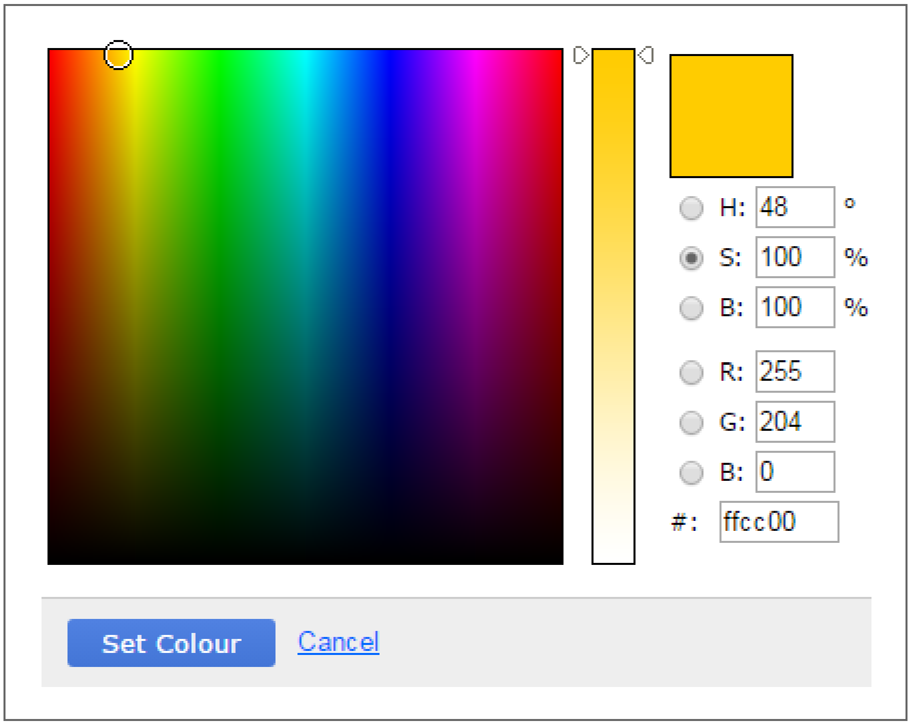
Creating a new theme

> **Important**
>
> Genesys recommends that users have some CSS development experience before creating a theme.

Click **Create new Theme**. The **Edit Theme** screen appears.

In the **Name** field, enter a unique name for your theme that describes its style and purpose. For example, you might call a theme **Sales - Red**, to indicate the theme is used by your company's sales department and the theme is based on the color red.

In the **Colour Palettes** section, specify which colors are available for use in this theme. You can click the **X** beside a color to remove it from the palette, making it unavailable for selection when configuring this theme. Or, you can add a color by clicking the **+** button. When you add a new color, Intelligent Automation displays a color-picker screen to allow you to customize the color. Click **Set Colour** when done to save the color to the theme's palette.



○ H: 48 °
● S: 100 %
○ B: 100 %
○ R: 255
○ G: 204
○ B: 0
#: ffcc00

Below the color palette, you can define CSS-based settings for everything from the theme's header to the appearance of validation messages.

At the bottom of the settings list is a section called **CSS Override**. You can provide custom CSS in this field to further customize your theme. Any CSS specified in the **CSS Override** section supersedes CSS settings in the other sections. For example, if you set a particular border style in the

Header section but then specified a different setting in the **CSS Override** section, the **CSS Override** setting is used.

> ### Important
> Refer to the Genesys Intelligent Automation Reference Guide for more information on using the **CSS Override** section.

Users

This page describes how to create, edit, and delete users that are stored in the Intelligent Automation database.

If you want to work with users defined in the Genesys Configuration Layer and stored in the Genesys Configuration Database, you must use Genesys Administrator Extension (GAX) to manage those users. Refer to User Authentication and Authorization for information about how to work with these *external* users.

Creating users

> **Important**
>
> You can only create user accounts with a role equal to or lesser than your own. For example, an **Application Designer** cannot create a **Full Administrator**.

To create a user:

1. Click **Create a new user**. The **Edit User** page opens.
2. Enter the following information:
   - **Email Address** - The user's email address. This address is the user's login name, as well as the email account Genesys Intelligent Automation uses for password-reset requests.
   - **Name** - The user's display name (for example, **John Smith**).
   - **Password** - The password must be between eight and 15 characters in length. See Password Rules for setting up password rules for users.
   - **Confirm Password** - Re-renter the password.
   - **Role** - The role for this user, which defines the permissions and features available to this user in Intelligent Automation.
   - **Force User to Change Password on Next Login** - Select this option to force the user to change their password the next time that he or she logs in to Intelligent Automation. Depending on the policies of the organization, this might be required for first-time users, users who have forgotten their password and need it reset, or for some other reason. This setting is cleared when the password is changed, so must be reactivated each time the password is to be changed.
3. Click **Save**.

Editing users

In the **Users** view, click **Edit** beside the user account you want to edit. The **Edit User** page opens.

The fields in the **Edit User** page work as described in the Creating users section, but you do not have to specify values in the **Password** and **Confirm Password** fields if you do not want to change the user's password.

Deleting users

In the **Users** view, click **Delete** beside the user account you want to edit.

> **Important**
>
> You cannot delete the user account you are currently using in Intelligent Automation.

Password Rules

To enable rules for passwords, set the **Login.Security.Strict** option in the **Default Server Settings** to *true*.

The following settings control how the password rules are enforced:

- Login.Failure.Threshold: The number of incorrect passwords that is allowed. When this number is exceeded, the account is locked.
- Login.Failure.LockoutDelayInSecs: The number of seconds for which an account is locked after the specified threshold limit.
- Password.PreviousPasswordsCount: The number of previous passwords that cannot be reused.
- Login.Password.ExpiryDays : The number of days after which the current password will expire and a new password has to be created for the account.
- Password.Rule.AlphanumericPatterns: The rule pattern that a password should match. For example, to allow alphanumneric paswords, set this to *[a-z]|[A-Z]|[0-9]*. The separator pipe symbol (|) indicates that the password must match all rules. In the example, this indicates that the password should contain a lowercase character, an uppercase character, and a number.

Import

The **Import** view allows you to import a ZIP file containing various application or module details.

To import files:

1. In the **What to Import** section, select one of the following options:
   - **Import Everything** – Includes all callflows, uploaded grammars, product-specific settings, and uploaded audio files.
   - **Import Prompts Only** – Includes uploaded audio files, such as those that are part of the callflow and those that are product-specific.
   - **Import Product-Specific Data and Prompts Only** – Unlike **Import Everything**, this option excludes callflow information, static prompts, and grammars.
2. In the **ZIP File to Import** section, select the zip file you want to import, and then click **Choose Modules to Import**.
3. The **Choose Modules to Import** window displays the modules that GAAP found in the ZIP file. You can:
   - Click **Ignore** beside a module that you do not want to import.
   - Enable the **Deploy these templates to production after import** check box to deploy these modules directly to your production environment.
4. Click **Import**.

Important

If a module is modified after import and reimported to a different environment, the **Link** blocks to other modules become empty. To avoid this, export all apps and modules even if only one module was modified.

Important

If you import an application or module that uses multiple retry or timeout prompts, but your current environment does not support this feature, the additional prompts are hidden from view but not deleted. These additional prompts appear only if it is supported by your environment.

Export

You can use the **Export** view to export various module details in XML format. You can also download any previously uploaded audio prompts.

To export files:

1. Click **Export** in the navigation bar to access the **Export** view.
2. In the **What to Export** section, select one of the following options:
   - **Export Everything** – Include all callflows, uploaded grammars, product-specific settings, and uploaded audio files.
   - **Export Prompts Only** – Include uploaded audio files, which includes those that are part of the callflow and those that are product-specific.
   - **Export Product-Specific Data and Prompts Only** – Unlike **Export Everything**, this option excludes callflow information, static prompts, and grammars.
3. In the **Modules to Export** section, select which Modules you want to export. You can select multiple modules by holding the **Ctrl** key on your keyboard as you click modules.
4. In the **Export Options** section, enable the **Use Production Version of Each Module** check box to export the production version of the module, or disable the check box to export the test version of the module.
5. Click **Export**.

> ## Important
>
> If a module is modified after import and exported to a different environment, the **Link** blocks to other modules become empty. To avoid this, export all apps and modules even if only one module was modified.

CTI Viewer

The CTI Viewer uses a combination of Whisper Transfer and special access to the reporting database to allow you to pass attached data between the caller and the agent, without requiring a CTI-enabled telephony environment.

If you have enabled Whisper Transfer in your Phone block, just before the transfer takes place, Genesys Intelligent Automation attempts to log the call history to the database and the database generates a new **Call ID** record. Intelligent Automation then creates a special scripting variable called **WhisperID** that you can include in the Whisper Transfer prompt (for example: **The code is [var:WhisperID]. Press 1 to accept the call**).

The CTI Viewer can be viewed by most user roles. However, there is a special **CTI Agen**t user role that is only allowed access to this page.

To use the CTI Viewer:

1. The agent logs into Intelligent Automation and clicks **CTI Viewer** in the navigation bar.
2. In the **CTI Viewer** view, click **Launch**.
3. In the **CTI Viewer** dialog box, enter the code provided by Whisper Transfer and click **Lookup**.
4. The call details display, including the CLI, any CTI attached-data fields that were attached by Script blocks, and a table showing all of the business tasks that were attempted during the call.

> ### Important
> CTI Viewer might show agents an extensive list of information. If needed, agents can click ★ to specify which details they want to view at the top of the list.

Call Monitor

Use the **Call Monitor** view to monitor and test your applications. There are two methods you can use to test an application:

- Call Monitoring - Allows you to monitor calls from a specific CLI. As the call progresses, you can visually follow the call as it flows through blocks and monitor call data and variables. This option is useful for situations in which you need to quickly test an application and do not need the robust analysis provided by a virtual call.
- Virtual Call - Allows you to virtually call an application and view a detailed analysis of variables, call settings, and call preferences in use. You can quickly start and restart calls, and select different callflow scenarios to test various aspects of your application. This option is useful in situations in which you need to drill down into more complex scenarios that are difficult to simulate in Call Monitoring, such as callflows that rely on web-service calls.

Call Monitoring

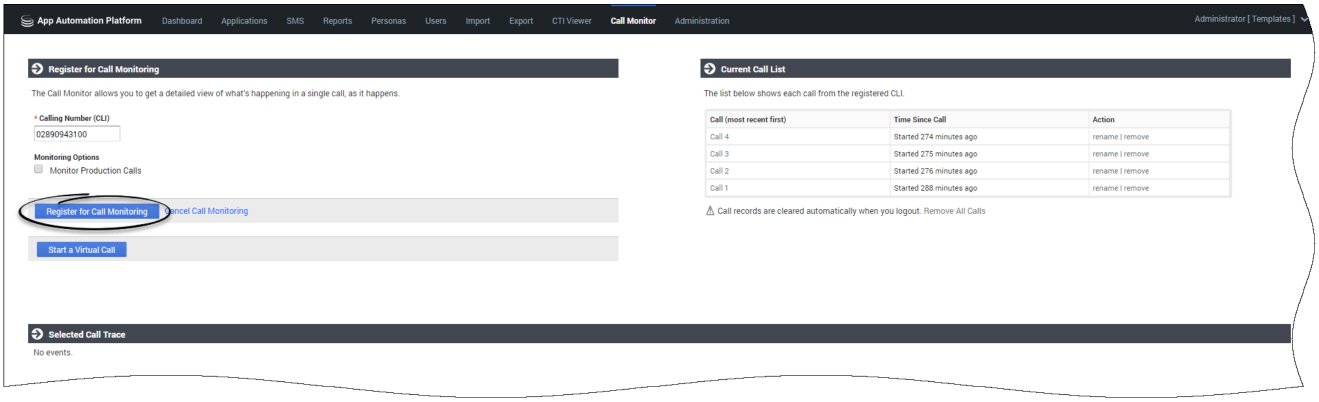
You can use call monitoring to view how an application responds to a caller in real time. Prompts and menu choices are displayed as the caller progresses through the call, allowing you to easily monitor the caller's path through the application, along with information about variables and other call data that are set by the application in response to the caller.

Starting call monitoring

In the **Call Monitor** view, go to the **Register for Call Monitoring** section and configure the following:

- **Calling Number (CLI)** - Enter the CLI to monitor. If you do not know the proper CLI format, you can check the call log in the Reports view and copy the CLI value into this field.
- **Monitor Production Calls** - By default, call monitoring only monitors calls to your test application. Select **Monitor Production Calls** if you also want to monitor calls to your production application.

Click **Register for Call Monitoring** to begin monitoring this CLI.



Register for Call Monitoring

The Call Monitor allows you to get a detailed view of what's happening in a single call, as it happens.

* Calling Number (CLI): 02890943100

Monitoring Options

□ Monitor Production Calls

Current Call List

The list below shows each call from the registered CLI.

Call (most recent first)	Time Since Call	Action
Call 4	Started 274 minutes ago	rename | remove
Call 3	Started 275 minutes ago	rename | remove
Call 2	Started 276 minutes ago	rename | remove
Call 1	Started 288 minutes ago	rename | remove

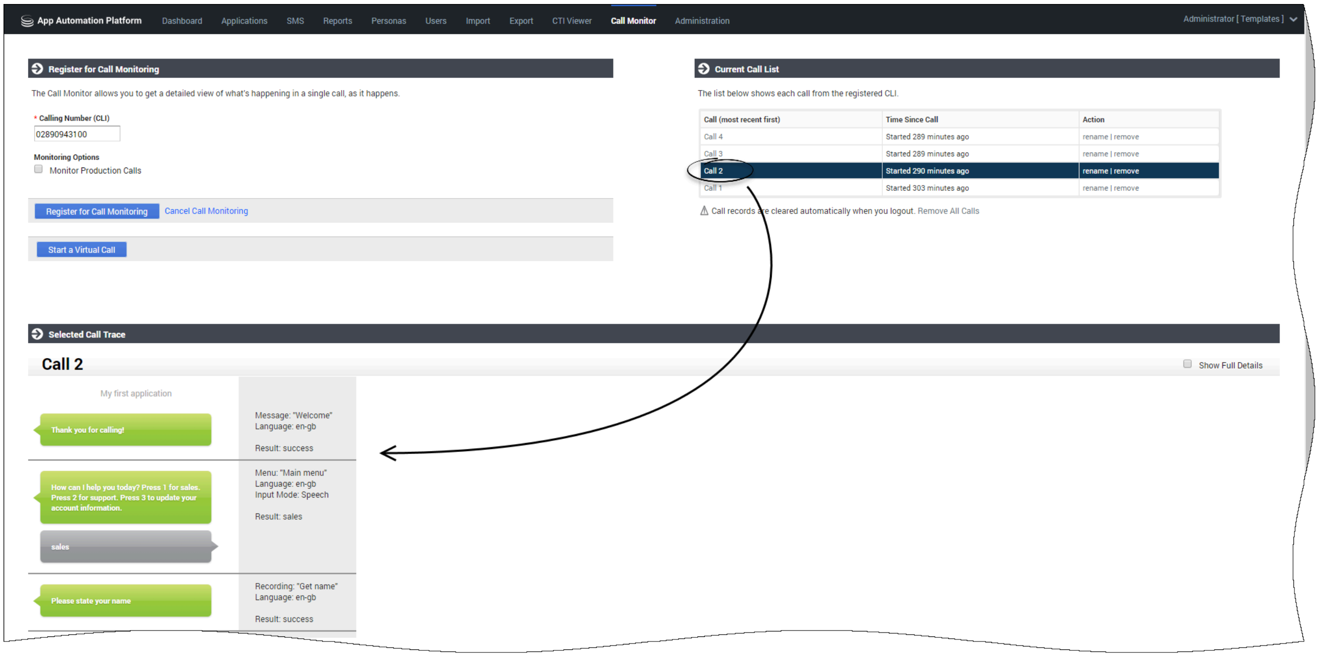
⚠ Call records are cleared automatically when you logout. Remove All Calls

Selected Call Trace

No events.

Viewing a monitored call

Once a monitored CLI calls, an entry for that call displays in the **Current Call List** section. Click the name of the call record (for example, **Call 2**) to view the call in the **Selected Call Trace** section.



Register for Call Monitoring

The Call Monitor allows you to get a detailed view of what's happening in a single call, as it happens.

* Calling Number (CLI): 02890943100

Monitoring Options

□ Monitor Production Calls

Current Call List

The list below shows each call from the registered CLI.

Call (most recent first)	Time Since Call	Action
Call 4	Started 289 minutes ago	rename | remove
Call 3	Started 289 minutes ago	rename | remove
Call 2	Started 290 minutes ago	rename | remove
Call 1	Started 303 minutes ago	rename | remove

⚠ Call records are cleared automatically when you logout. Remove All Calls

Selected Call Trace

Call 2

□ Show Full Details

My first application

Message: "Welcome"
Language: en-gb

Result: success

Menu: "Main menu"
Language: en-gb
Input Mode: Speech

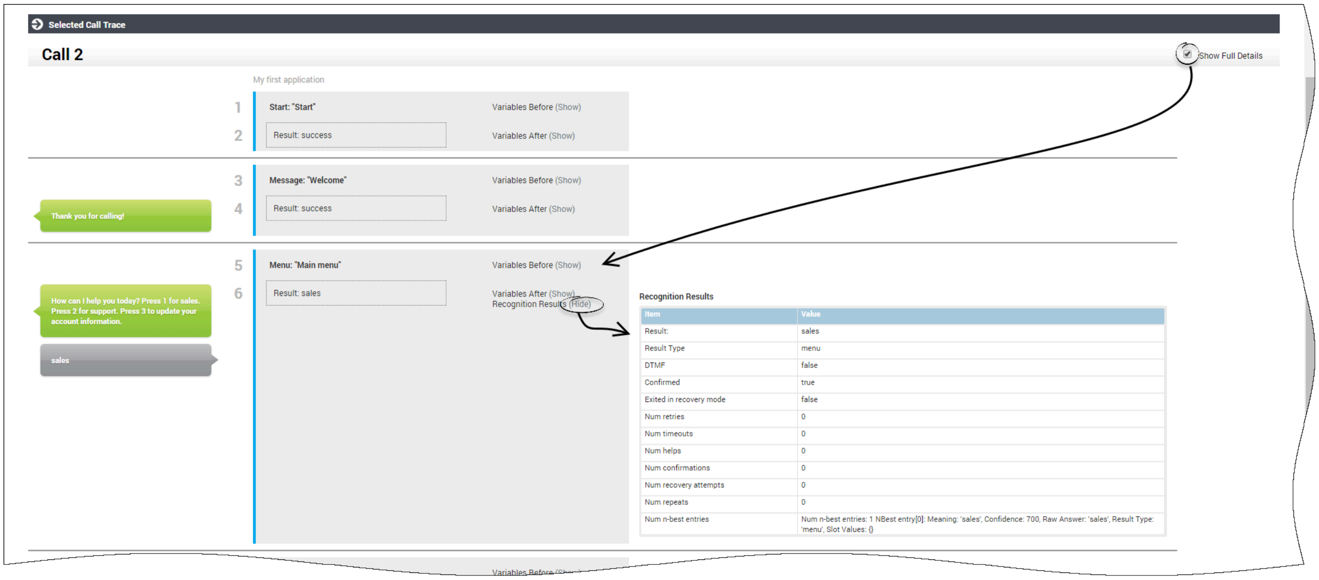
Result: sales

Recording: "Get name"
Language: en-gb

Result: success

By default, the **Selected Call Trace** section shows a high-level overview of the call's progression. You can see prompts played to the caller, menu options the caller has selected, responses to questions, and activity within linked modules. You do not see progression through blocks that do not require user input, such as Script blocks.

Click **Show Full Details** to view more information about the call, such as the value of variables at each step, detailed breakdowns of responses to Menu and Question blocks, and much more.



Selected Call Trace

Call 2

☑ Show Full Details

My first application

1 Start: "Start" Variables Before (Show)
2 Result: success Variables After (Show)

Thank you for calling!

3 Message: "Welcome" Variables Before (Show)
4 Result: success Variables After (Show)

How can I help you today? Press 1 for sales. Press 2 for support. Press 3 to update your account information.

sales

5 Menu: "Main menu" Variables Before (Show)
6 Result: sales Variables After (Show)
Recognition Results (Hide)

Recognition Results

Item	Value
Result:	sales
Result Type	menu
DTMF	false
Confirmed	true
Exited in recovery mode	false
Num retries	0
Num timeouts	0
Num helps	0
Num confirmations	0
Num recovery attempts	0
Num repeats	0
Num n-best entries	Num n-best entries: 1 NBest entry[0]: Meaning: 'sales', Confidence: 700, Raw Answer: 'sales', Result Type: 'menu', Slot Values: {}

Variables Before (Show)

Ending call monitoring

Genesys Intelligent Automationcontinues to monitor a CLI until instructed to stop. Therefore, you might want to cancel call monitoring after you are done testing your application. To do so, go to the **Register for Call Monitoring** section and enter the CLI to stop monitoring in the **Calling Number (CLI)** field. Click **Cancel Call Monitoring**.

Virtual Call

You can use virtual calls to test various aspects of your applications. A virtual call takes you block by block through the application, letting you select various inputs or events to determine how the application responds to a wide range of scenarios. As the call progresses, you can monitor back-end information such as variables, preferences, web-service calls, and even the VXML being generated for voice calls.

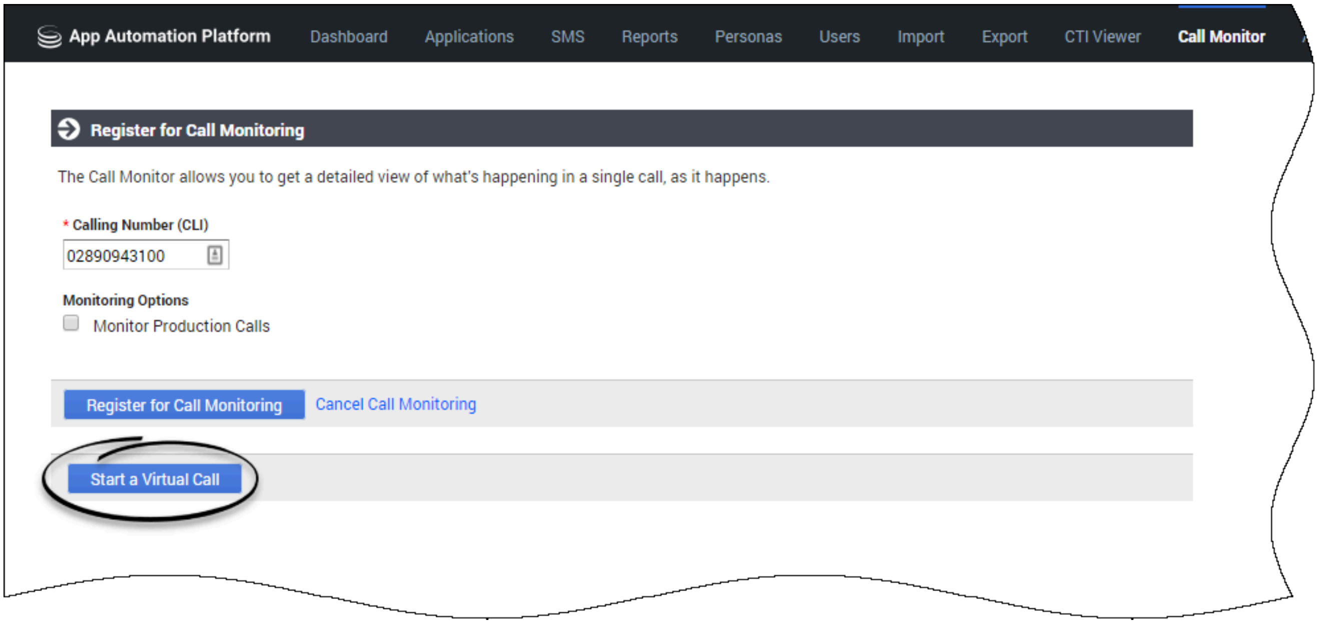
Important

- Virtual calls simulate modules only if they are called from the application (for example, by a Link block). Otherwise, you cannot simulate modules as an isolated entity.
- Virtual calls provide information about elements such as call history, variables, attached data, business tasks, VUI preferences, and more. However, you might not see certain elements in a virtual call until data is available. For example, you cannot see information about a variable or business task until it is invoked during the virtual call.
- If your environment supports multiple retry or timeout prompts, you must click **Toggle**

> **extra prompts** in certain blocks to see these additional prompts.

Starting a virtual call

In the **Call Monitor** view, click **Start a Virtual Call**.



The Call Monitor allows you to get a detailed view of what's happening in a single call, as it happens.

* Calling Number (CLI): 02890943100

Monitoring Options

□ Monitor Production Calls

The **Virtual Call** opens in a new window.

At the top of the **Virtual Call** window, you can select the following settings:

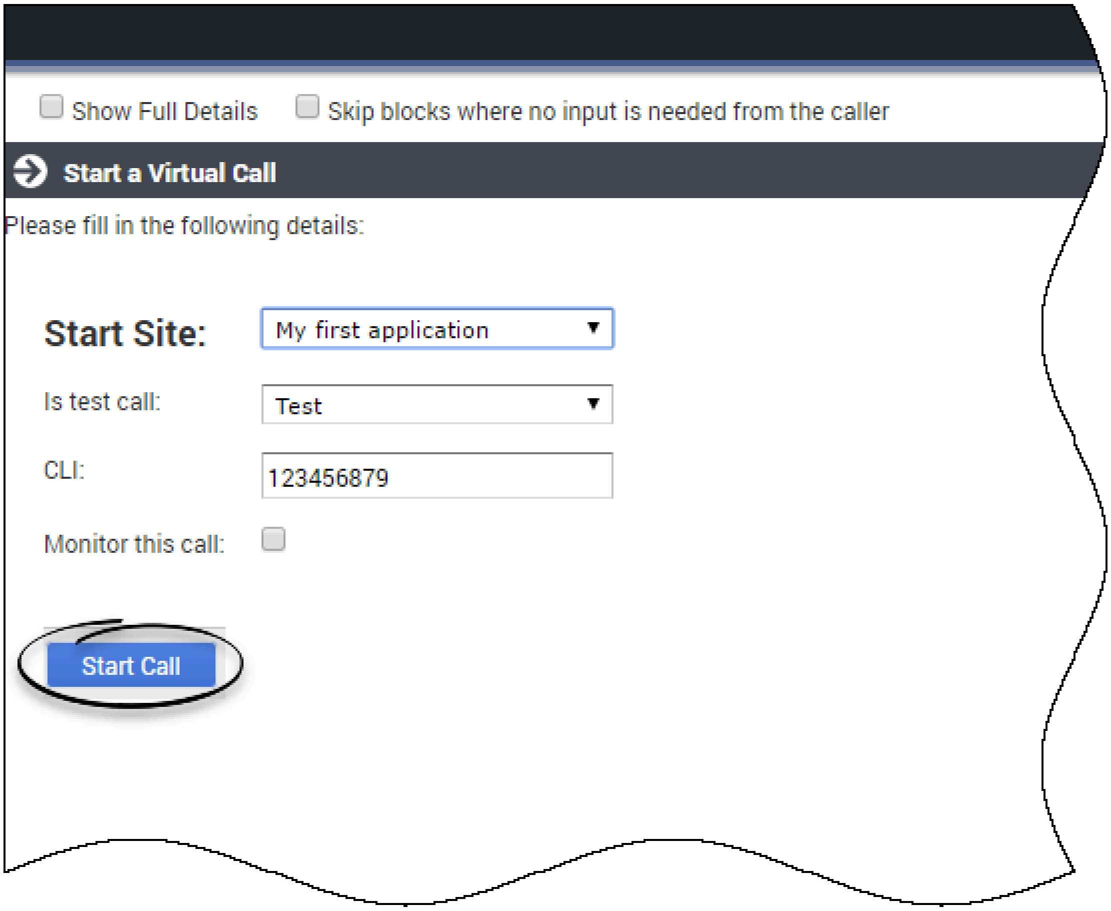
- **Show Full Details** - Select to see the options to be applied to the virtual call. Most options can be left at their default settings. However, you might want to change the following:
  - **Channel** - Select which channel to use for the call. For example, you can select **Voice** for phone calls or **Web** for a Web IVR call. This is useful if you want to simulate customer experiences across various channels.
  - **DNIS** - Select the DNIS that the call used to access the application. This setting is useful for testing how your application responds when accessed by one DNIS as compared to another, such as when one customer segment uses one DNIS and another customer segment uses another DNIS.
- **Skip blocks where no input is needed from the caller** - If you do not select this option, the virtual call proceeds through every block, even if user input is not needed. For example, at the Start block, you must click the **success** event to proceed to the next block. However, if you do select this option, the virtual call stops only when user input is needed, such as at a Menu block.

In the **Start a Virtual Call** section, specify the following:

- **Start Site** - Select the application to test.

- **Is test call** - Specify whether the virtual call uses the test version of the application. If this option is not selected, the production version of the application is used.
- **CLI** - Specify the CLI to use for the virtual call.
- **Monitor this call** - Select this option to also use Call Monitoring with this virtual call.

Click **Start Call** to begin the virtual call.



□ Show Full Details □ Skip blocks where no input is needed from the caller

Start a Virtual Call

Please fill in the following details:

Start Site: My first application

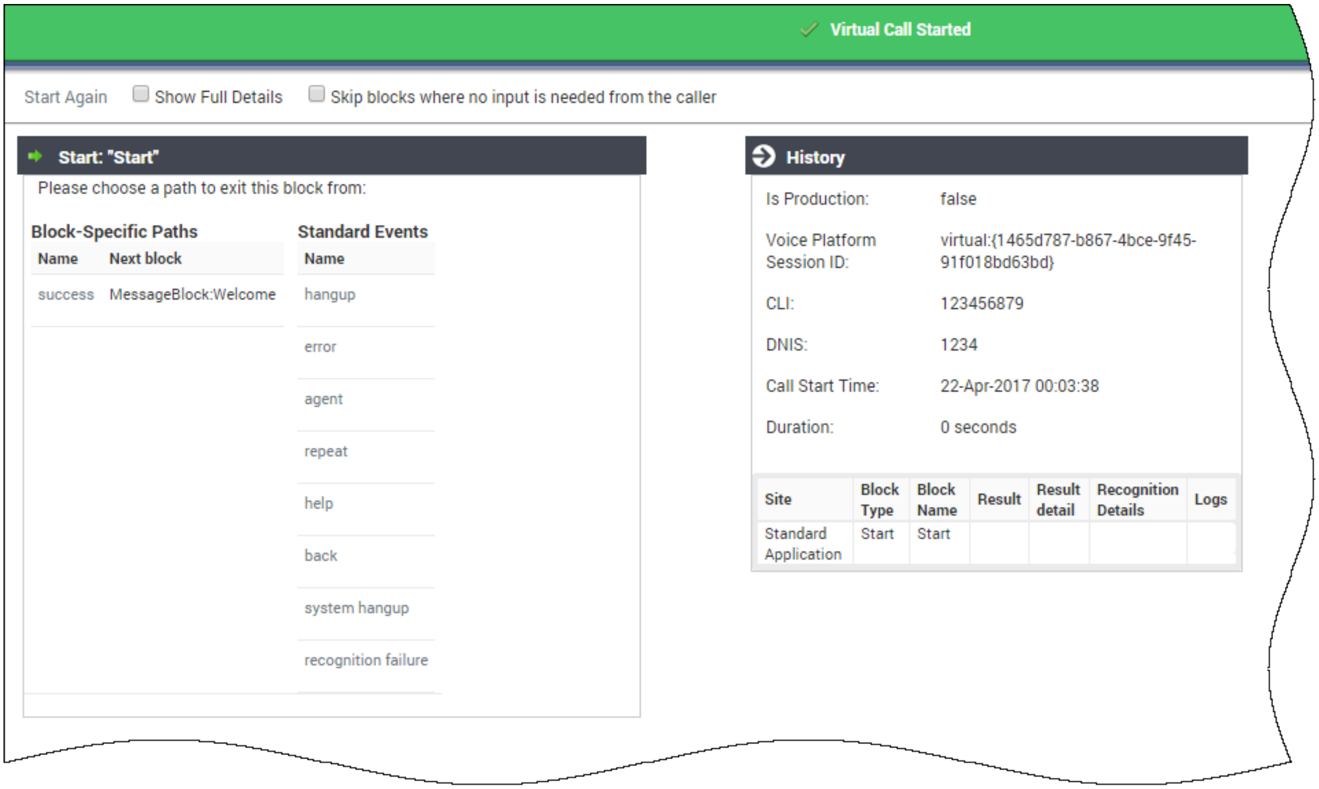
Is test call: Test

CLI: 123456879

Monitor this call: □

The virtual call begins and simulates how the application responds to a caller. You can monitor the value of variables and the status of callflow preferences to determine how they influence the call experience. If you select the **Show Full Details** option, you can also view the VXML code as it is generated.

The power of virtual calls is being able to quickly and easily simulate scenarios that might be time-consuming or difficult to simulate with actual calls. For example, at any point in your application, you can trigger standard events such as **error** to see how the application responds when an error occurs. Or, in a Menu block, you might try specifying various inputs to see how the block processes the input. You can click **Start Again** in the top-left corner of the screen to restart the call and simulate different events and outcomes.



✓ Virtual Call Started

Start Again □ Show Full Details □ Skip blocks where no input is needed from the caller

Start: "Start"

Please choose a path to exit this block from:

Block-Specific Paths

Name	Next block
success	MessageBlock:Welcome

Standard Events

Name
hangup
error
agent
repeat
help
back
system hangup
recognition failure

History

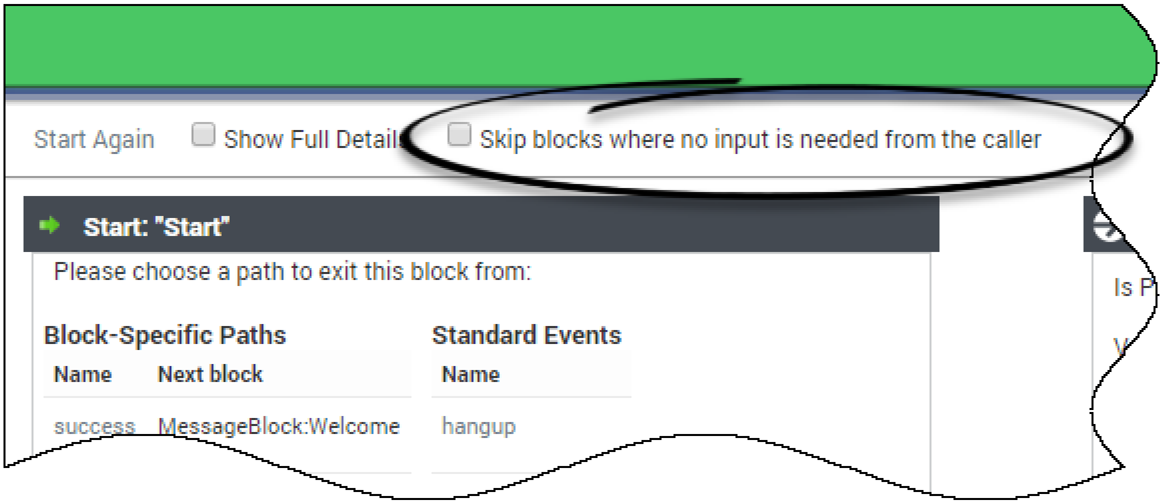
Is Production:	false
Voice Platform Session ID:	virtual:{1465d787-b867-4bce-9f45-91f018bd63bd}
CLI:	123456879
DNIS:	1234
Call Start Time:	22-Apr-2017 00:03:38
Duration:	0 seconds

Site	Block Type	Block Name	Result	Result detail	Recognition Details	Logs
Standard Application	Start	Start				

You can trigger the following standard events at any time during a virtual call:

- **hangup** - Simulates a caller hangup.
- **error** - Simulates an unspecified error.
- **agent** - Simulates the **agent** default path.
- **repeat** - Simulates a global default to repeat the last prompt.
- **help** - Simulates a global default to provide help.
- **back** - Simulates a global default to go back to the previous block.
- **system hangup** - Simulates a system-side hangup.
- **recognition failure** - Simulates an input-recognition failure.

If you run multiple virtual calls to test a certain section of the application, you can skip certain blocks that do not require user input, such as Message blocks. To do this, select the option **Skip blocks where no input is needed from the caller** to display only blocks that require input from the caller, such as a Menu block or Recording block. You can turn this option on and off throughout the virtual call to skip sections of the application that you do not want to include in your test.



Start: "Start"

Please choose a path to exit this block from:

Block-Specific Paths

Name	Next block
success	MessageBlock:Welcome

Standard Events

Name
hangup

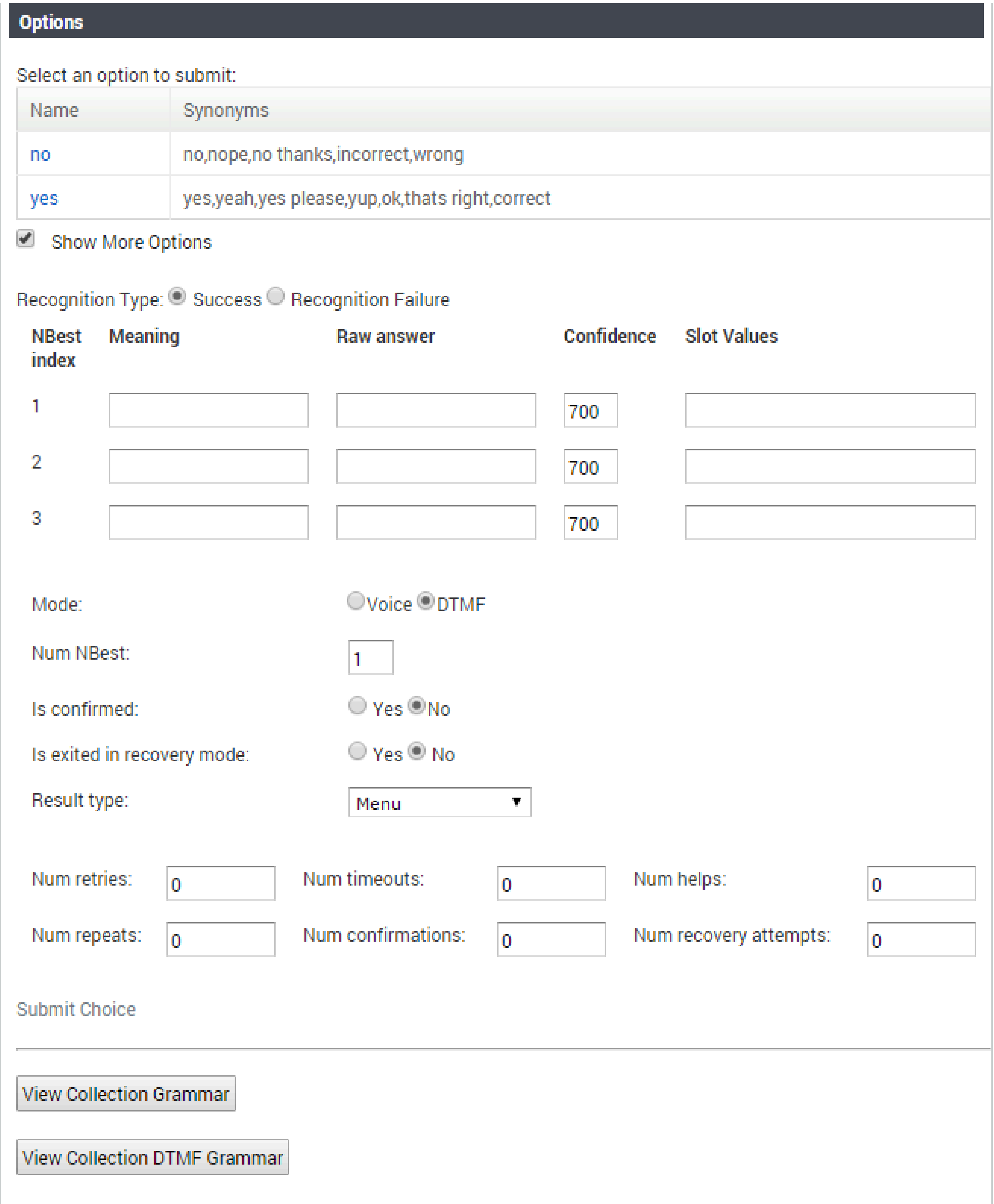
The virtual call ends either when you progress past the last block in the application, or if you simulate an event such as **hangup**.

Some blocks allow you to set specific options. For example, in a Menu block, you can try various responses to test how the application processes the call. The following sections explain specific options that pertain to each block during a virtual call.

Menu Block

The **Options** section allows you to specify the caller's input during the virtual call. For example, if the **Menu** block options are **yes** and **no**, you can select either option and proceed with the virtual call.

Click **Show More Options** to view other input options. For example, you can simulate an input-recognition failure, or you can input one or more words that sound similar to determine if the call proceeds as expected.



Options

Select an option to submit:

Name	Synonyms
no	no,nope,no thanks,incorrect,wrong
yes	yes,yeah,yes please,yup,ok,thats right,correct

☑ Show More Options

Recognition Type: ● Success ○ Recognition Failure

NBest index	Meaning	Raw answer	Confidence	Slot Values
1			700	
2			700	
3			700	

Mode: ○Voice ●DTMF

Num NBest: 1

Is confirmed: ○ Yes ●No

Is exited in recovery mode: ○ Yes ● No

Result type: Menu ▼

Num retries:	0	Num timeouts:	0	Num helps:	0
Num repeats:	0	Num confirmations:	0	Num recovery attempts:	0

Submit Choice

The following options are available if you select **Show More Options**:

- **Recognition Type** - Select one of the following options:
  - **Success** - Simulates a successful input recognition. In the fields below, you can enter and rank various criteria for successful input, as described below:
    - **Meaning** - The interpreted result from input recognition. For example, **1001**.
    - **Raw answer** - The actual input. For example, the caller might say "one double-o one" as a raw answer. In the **Meaning** field, this is recognized as **1001**.
    - **Confidence** - Assign a confidence ranking to this answer, on a scale from **0** (low) to **1000** (high).
    - **Slot Values** - This field is not typically used and can be left blank.
    - **Mode** - Select the input mode to use for input recognition. Select **Voice** for speech recognition and **DTMF** for DTMF keypad input.
    - **Num NBest** - Specify the number of NBest options to consider when processing input.
    - **Is confirmed** - Specify whether the input must be confirmed with the caller before proceeding.
    - **Is exited in recovery mode** - Specify whether to exit this block in recovery mode (for example, if the caller provided too many unsuccessful inputs).
    - **Result type** - Specify the result type to use for input. For example, if the caller says "agent" and you have a menu option called **agent** and a default path called **agent**, this value specifies which type to use in this virtual call.
  - **Recognition Failure** - Select this option to simulate an unsuccessful input recognition.

In the **Num ...** fields, you can specify how many times the caller used each option. For example, enter 2 in the **Num retries** field to specify the caller had two retries when providing input.

Optionally, select one of the following to download and view the grammar file used for each input type:

- **View Collection Grammar**.
- **View Collection DTMF Grammar**.

Click **Submit Choice** to process the input.

Question Block

The input options for Question blocks are similar to Menu blocks. See the **Menu Block** section above for more information.

Phone Block

Specify the outcome of the Phone block. The following standard outcomes are available:

- **Success** - The virtual call simulates a successful transfer and the call ends.
- **Busy** - The virtual call simulates a busy signal during transfer and plays a standard prompt.
- **No Answer** - The virtual call simulates an unanswered transfer and plays a standard prompt.

- **Transfer Failure** - The virtual call simulates a failure during transfer and plays a standard prompt.

Recording Block

You can upload an audio file to simulate the caller providing a voice recording.

Script and Interceptor blocks

These blocks provide the output path returned by the script or defined in the interceptor logic.

> **Important**
>
> Choose the output path returned by the Script or Interceptor block. Choosing another path might lead to errors elsewhere in the virtual call.

Phone Block

You can select the outcome of the transfer.

If an Opening Hours rule applies, the outcome of that rule is shown.

Using the Current Call List

The **Current Call List** shows a record of your monitored calls and virtual calls. This list clears after you log out and does not persist between user sessions.

Besides viewing information about these calls, you can:

- Rename a call record - Click **rename** to rename the call record. By default, calls are named **Call 1**, **Call 2** and so on, but you can use more descriptive names, particularly if you choose to generate test cases based on these calls using Cyara.
- Remove a call record - Click **remove** to remove the call record from the list. Once removed, you cannot retrieve the record again.

Using Cyara to generate test calls

If you have purchased Cyara, you can select items in the Current Call List to generate XML for importing into Cyara. To do so, click the check box in the **Generate?** column for each call you want to select for test cases.

In the **Generate Cyara Test Case** window, enter a filename and click **Save** to save the XML to your computer.

Tip

You can click **Show Cyara Settings** to view and configure information related to the generated XML file. Refer to the Cyara documentation for more information on configuring these settings and using Cyara.

Administration

The **Administration** view allows you to create, modify, and delete settings and configurations that govern your Genesys Intelligent Automation environment.

The following tabs are available:

- Companies
- Roles
- Products
- Maintenance
- Phone Numbers
- CLI Data Settings
- Certificates
- Clusters
- Servers
- Default Server Settings

Companies

Companies are separate entities within your Intelligent Automation environment that have their own callflows, users, and reporting data. Companies are similar to subfolders within an operating system's root folder. All folders share a common set of configurations, but each folder can have its own files, permissions, and rules. Likewise, you can create separate companies if your business requires separate entities with unique user lists, reporting data, and callflows.

> **Important**
>
> The **Templates** company is created after you install Intelligent Automation to host the templates you need to create products. Genesys recommends you do not delete the **Templates** company.

Creating a company

To create a company:

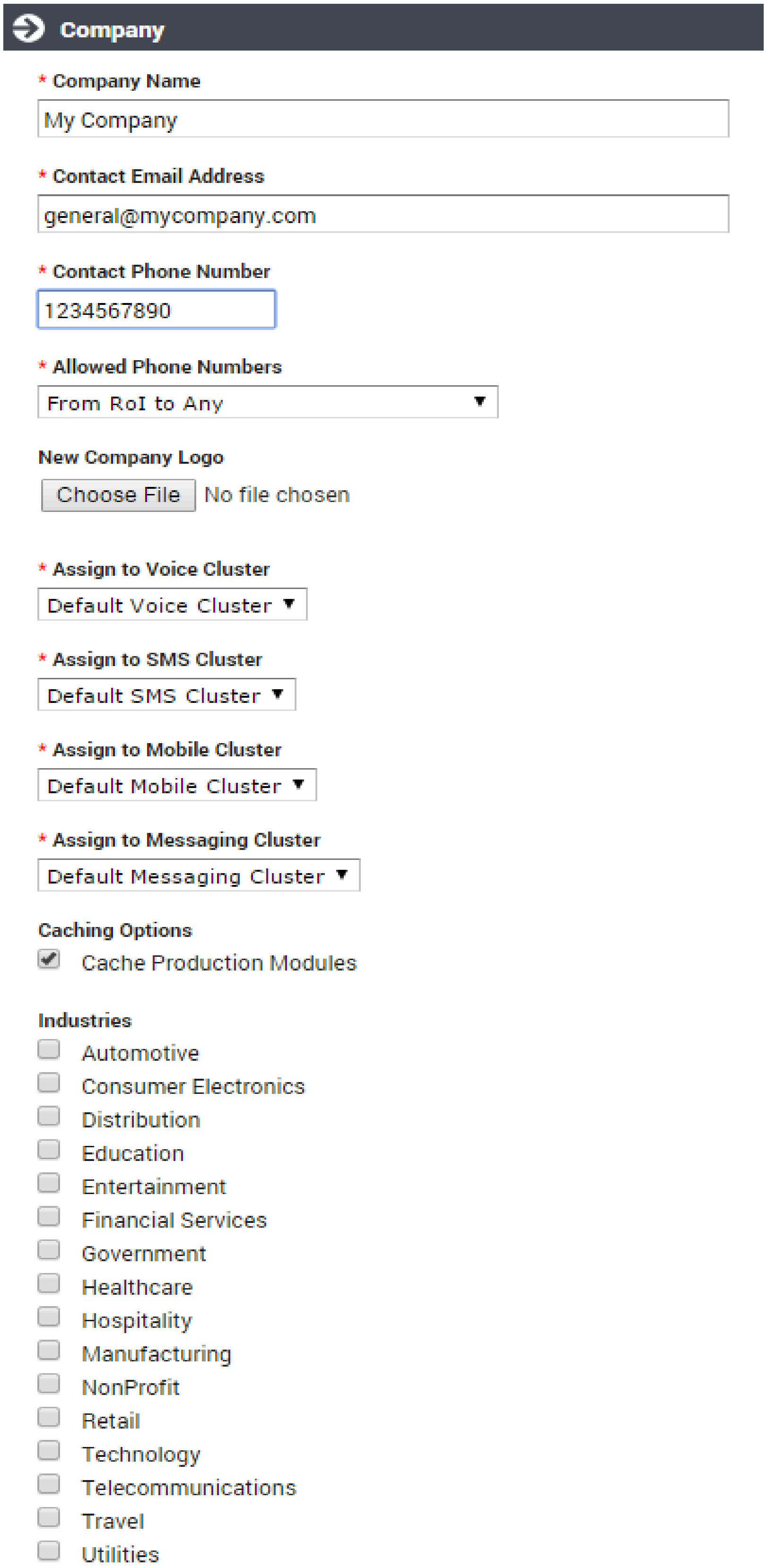
1. Click **Create a New Company**.
2. In the **Company** section, specify the following information:

- **Company Name** - Enter a name for your company.
- **Contact Email Address** - Enter a general email address for your company. This email address is not used by Intelligent Automation, but it might be useful for support personnel.
- **Contact Phone Number** - Enter a general phone number for your company. This phone number is not used by Intelligent Automation, but it might be useful for support personnel.
- **Allowed Phone Numbers** - Specify which phone numbers can be dialled by this company's callflows. In most cases, users select **From RoI to Any** or **From UK to Any**, but you can create your own calling rules in the Phone Numbers tab and select it here.
- **New Company Logo** - Upload a company logo that you can display in the Intelligent Automation user interface.
- **Configuration Server Tenant DBID** - If users stored in the Genesys Configuration Database will be part of this company, enter the ID of the user's tenant that corresponds to this company. This ID will be used to determine the company that is to be loaded when a user who is not an Intelligent Automation user is logging in. See the At login section of the Intelligent Automation Reference Guide.

> Important
>
> This field will not appear unless the **Login.ExternalAuthenticationMode** server setting is set to `ConfigServer`.

- **Assign to Voice Cluster** - Select the voice cluster that this company uses. Most users select **Default Voice Cluster**.
- **Assign to SMS Cluster** - Select the SMS cluster that this company uses. Most users select **Default SMS Cluster**.
- **Assign to Mobile Cluster** - Select the mobile cluster that this company uses. Most users select **Default Mobile Cluster**.
- **Assign to Messaging Cluster** - Select the messaging cluster that this company uses. Most users select **Default Messaging Cluster**.
- **Cache Production Modules** - If enabled, Intelligent Automation uses cached versions of production modules to serve callers. Genesys recommends you enable this setting for consistent performance.
- **Industries** - Optional. Select one of the boxes to identify the industry in which your company operates. This tag is not used elsewhere in Intelligent Automation and is descriptive only.

3. Click **Save**.



Company

* Company Name

My Company

* Contact Email Address

general@mycompany.com

* Contact Phone Number

1234567890

* Allowed Phone Numbers

From RoI to Any ▼

New Company Logo

Choose File No file chosen

* Assign to Voice Cluster

Default Voice Cluster ▼

* Assign to SMS Cluster

Default SMS Cluster ▼

* Assign to Mobile Cluster

Default Mobile Cluster ▼

* Assign to Messaging Cluster

Default Messaging Cluster ▼

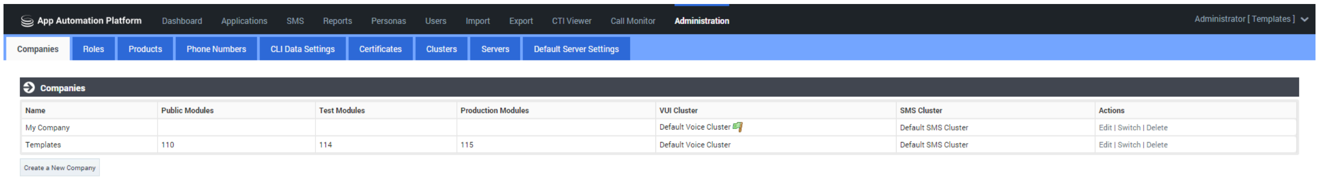
Caching Options

☑ Cache Production Modules

Industries

- □ Automotive
- □ Consumer Electronics
- □ Distribution
- □ Education
- □ Entertainment
- □ Financial Services
- □ Government
- □ Healthcare
- □ Hospitality
- □ Manufacturing
- □ NonProfit
- □ Retail
- □ Technology
- □ Telecommunications
- □ Travel
- □ Utilities

The Companies list now displays the company you created.



Companies

Name	Public Modules	Test Modules	Production Modules	VUI Cluster	SMS Cluster	Actions
My Company				Default Voice Cluster	Default SMS Cluster	Edit | Switch | Delete
Templates	110	114	115	Default Voice Cluster	Default SMS Cluster	Edit | Switch | Delete

Other actions

> ### Important
> Your user account must be assigned to either the **Platform Administrator** or **Full Administrator** role before you can perform the actions below.

In the **Actions** column, you can click:

- **Edit** - Edit company details.
- **Switch** - Switch to this company. When using Intelligent Automation, if your user account has sufficient privileges, you can switch between using one company and another.
- **Delete** - Delete the company and all information related to the company, such as callflows and user accounts.

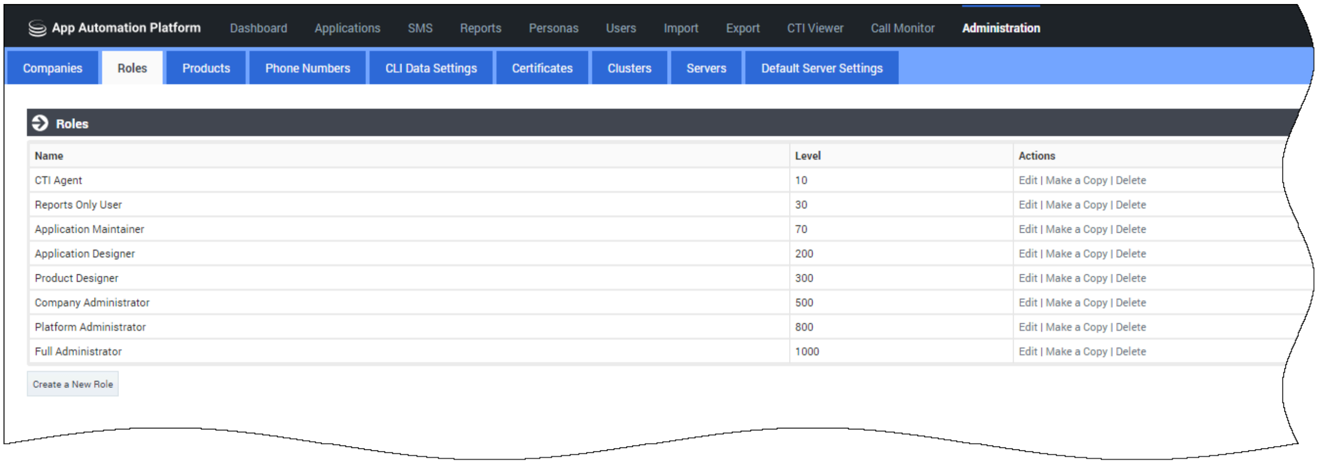
Roles

Roles govern what a user can do in Intelligent Automation. In effect, they determine what a user can do to what he or she can see. For example, a user with the **Full Administrator** role has full permissions to configure Intelligent Automation, from creating companies, users, server settings, and much more. However, a user with the **Reports Only User** role can view information related to reports but cannot create applications.

This section describes how Roles are managed for users stored in the Intelligent Automation database. If you are going to work with users stored in the Genesys Configuration Database, the Roles to which they are subject are defined by the Genesys Configuration Server. See User Authentication and Authorization for more information.

> ### Important
> For ease of use and effective troubleshooting, Genesys recommends that you do not modify the standard user roles in Intelligent Automation. Instead, click **Make a Copy** beside the role you want to modify and create a new role with modified permissions.

The **Roles** list shows the roles available in your company. The **Levels** column is descriptive only and is meant to illustrate the relative permission levels between each role.



Roles

Name	Level	Actions
CTI Agent	10	Edit | Make a Copy | Delete
Reports Only User	30	Edit | Make a Copy | Delete
Application Maintainer	70	Edit | Make a Copy | Delete
Application Designer	200	Edit | Make a Copy | Delete
Product Designer	300	Edit | Make a Copy | Delete
Company Administrator	500	Edit | Make a Copy | Delete
Platform Administrator	800	Edit | Make a Copy | Delete
Full Administrator	1000	Edit | Make a Copy | Delete

The following is a general description of what each role can do in Intelligent Automation. Unless otherwise noted, each successive role in the list builds upon the permissions of the previous role:

- **CTI Agent** - Can access the CTI Viewer view.
- **Reports Only User** - Can also access the Dashboard, Applications, and Reports views.
- **Application Maintainer** - Can also access the Personas view and modify certain aspects of callflows.
- **Application Designer** - Can also access the Import, Export, and Call Monitor views and create callflows.
- **Product Designer** - Can also access the Products tab in the **Administration** view.
- **Company Administrator** - Cannot access the Products tab in the **Administration** view, but can edit company details in the My Account page.
- **Platform Administrator** - Can also access the **Administration** view and switch companies.
- **Full Administrator** - Can access any page or setting in Intelligent Automation.

Creating a role

If needed, you can create roles to fulfill particular business needs. In most cases, Genesys recommends creating a copy of an existing role and modifying its permissions. Click **Make a Copy** beside the role you want to modify. If necessary, you can choose to create a new role by clicking **Create a New Role**. For both methods, do the following:

- **Role Name** - Enter a descriptive name for the role.
- **Level** - Enter a numerical value to help describe the relative permission level of this role. For example, if you are making a copy of the **Application Designer** role but you want to add some additional permissions, you can set a higher level value (for example, **250** instead of the original **200**) to show this role has more permissions.
- **Role Permissions** - Enable or disable various permissions for the role.

Click **Save**.

Other actions

In the **Roles** list, you can select one of the following in the **Actions** column:

- **Edit** - Edit the role.
- **Make a Copy** - Make a copy of the role.
- **Delete** - Delete the role. You cannot delete a role that is assigned to at least one user account.

Products

Products are prebuilt modules that you can insert into your applications. Although you can't edit product modules, you can set parameters to customize them for your business needs.

The **Products** list shows the products available in your environment. The products in this list are governed by your Intelligent Automation license, so the list can differ from one company to another.

Deleting a product

You can delete a product that is no longer in use. Click **Delete** beside the product you want to remove.

Importing a product

You can import a new product supplied by Genesys by going to the bottom of the **Products** list and clicking **Import a Product**.

Maintenance

> **Important**
>
> Ensure that data is backed up before performing any maintenance options.

The **Maintenance** tab allows you to perform maintenance tasks for databases. The following operations are supported:

- Trimming iHub tables: This option allows you to delete iHub processes before the specified date.
- Trimming Aggregate tables: This option allows you to remove Aggregate table entries before the specified date. See Aggregate tables in Oracle and Aggregate tables in MSSQL for list of tables. If you want to modify the block count, set the **Maintenance.AggregateTable.BlockDeleteSize** server

setting to a desired value. Changing the default value to a higher value could impact performance.

- Cleaning up Aggregate tables in Microsoft SQL tables: The **Purge Duplicates** option allows removing duplicate entries in the Aggregate tables.

Phone Numbers

The **Phone Numbers** tab allows you to specify rules for which phone numbers can be called by Intelligent Automation callflows. For example, you might want callflows to use a ruleset that prohibits calls to mobile numbers.

The following rule sets are installed with Intelligent Automation:

- **From RoI to Any** - Allows calls from the Republic of Ireland to anywhere in the world.
- **From RoI to RoI Landlines** - Allow calls only from the Republic of Ireland to landline numbers in the Republic of Ireland.
- **From RoI to RoI Landlines and Mobiles** - Allow calls only from the Republic of Ireland to landline and mobile numbers in the Republic of Ireland.
- **From UK to Any** - Allows calls from the United Kingdom to anywhere in the world.
- **From UK to UK Landlines** - Allow calls only from the United Kingdom to landline numbers in the United Kingdom.
- **From UK to UK Landlines and Mobiles** - Allow calls only from the United Kingdom to landline and mobile numbers in the United Kingdom.

Creating a new rule set

Click **Create a New Ruleset** to define a new set of rules. In the **Phone Number Ruleset** section, configure the following:

- **Ruleset Name** - Enter a descriptive name for the rule set.
- **Rules** - Define the rules that govern this rule set. Click **Add row** to add a new rule.
  - **Is Allowed?** - Select **Yes** to allow calls that match the rule, or **No** to prohibit calls that match the rule.
  - **Prefix** - Specify the prefix of the phone number for the rule.
  - **Min Length** - Specify the minimum number of phone-number digits for the rule.
  - **MaxLength** - Specify the maximum number of phone-number digits for the rule.

Click **Save**.

Other actions

In the **Phone Number Rulesets** list, you can select one of the following in the **Actions** column:

- **Edit** - Edit the rule set.
- **Delete** - Delete the rule set.

CLI Data Settings

> **Important**
> This section describes how to use the **Built-in** CLI Data Mechanism. Contact your Genesys representative if you want information for how to integrate this feature with Genesys Conversation Manager.

The **CLI Data Settings** tab allows you to use regular expressions to define allowed phone numbers against which to store CLI data in the database.

You can configure the following:

- **Positive Pattern for Valid CLIs** - Enter a regular expression to define a range of allowed phone numbers against which CLI data *can* be stored.
- **Negative Pattern for Valid CLIs** - Enter a regular expression to define a range of allowed phone numbers against which CLI data *cannot* be stored.

Certificates

The **Certificates** tab allows you to import and create certificates for use with HTTPS connections in your Intelligent Automation environment.

Importing a certificate

Click **Import a new Certificate** to import a certificate from a machine on your network. You must provide the IP or FQDN of the server that has the certificate, along with the appropriate port number. Click **Get Certificate**.

If the certificate is found, Intelligent Automation displays details about the certificate to be imported. Enter a description to describe the purpose of the certificate, then click **Save**.

Creating a self-signed certificate

Click **Create a Self Signed Certificate** and enter the following information:

- **Server(s)** - Enter the server IP or FQDN that will host the certificate.

- **Pass Phrase** - Enter a pass phrase for the certificate. Note this value for future use.
- **Add to Trusted Certificates List** - If enabled, the self-signed certificate is trusted. This might prevent certain security warnings from appearing in your browser.

As this is a self-signed certificate, you do not need to enter information in the fields **Organisation Unit**, **Organisation**, **City**, **State**, and **Two-Letter Country Code** unless directed to do so by Genesys or your company.

Click **Generate**.

> Important
>
> After you generate the certificate, you must update the password on the host machine.
>
> 1. Open the file **server.xml** in the following location: **C:\SpeechStorm\Platform\TomcatVUI\conf**.
> 2. Update the **SSLPassword** value to the pass phrase you entered when you created the self-signed certificate.
> 3. Restart the server.

Other actions

In the **Certificates** list, you can select one of the following in the **Actions** column:

- **Edit** - Edit the certificate's details.
- **Delete** - Delete the certificate.

Clusters

The **Clusters** tab allows you to group servers to reflect your business needs. For example, you might want to create a cluster of VUI servers assigned to a particular load balancer to better manage your call volume. Or, if you are in a multi-tenant environment, you can use clusters to segregate groups of servers to particular tenants and ensure resources for one company are not shared with another.

The **Clusters** list shows the clusters in your environment, and the load balancer and resources cluster to which each cluster is assigned.

Creating a cluster

Click **Create a New Cluster** and select a cluster type.

Enter the following information:

- **Cluster Name** - Enter a descriptive name for the server.
- **Load Balancer Configuration** - Select one of the following options:
  - **No load balancer** - Do not use a load balancer with this cluster.
  - **Use internal Load Balancer cluster** - Select a load balancer that you previously created in the Servers tab.
  - **Use an external Load Balancer** - Select an external load balancer. You must provide a URL and port number to which this cluster must send traffic.

Click **Save**.

Using load balancers

You can assign clusters to use load balancers to better manage Messaging Server traffic on your network. Intelligent Automation provides an internal load balancer that you can use, or you can provide your own solution.

To create a load balancer, click **Create a New Cluster** and select **New Load Balancer Cluster** in the pop-up menu. You must configure the following:

- **Cluster Name** - Provide a descriptive name.
- **Load Balancer Servers Will Balance Requests Arriving at This Port** - Specify the protocol (HTTP or HTTPS) and server port number for the load balancer.
- **Hostname Used in External Links to this Cluster** - Provide a hostname that programs and services can use to connect to the load balancer. For example, do not provide an internal IP address or hostname that cannot be accessed by customers who click links to WebIVR applications.

Click **Save**.

To assign clusters to use a load balancer, click **Edit** beside a cluster and select the load balancer, and then click **Save**.

Using resource clusters

You can use resource clusters to provide a central storage location for all your customer-specific resource files, such as audio files. Otherwise, Intelligent Automation stores these files in the **resources** folder of the current VUI (for example, C:\SpeechStorm\Platform\TomcatVUI\webapps\fish-vui\resources).

To create a resource cluster, click **Create a New Cluster** and select **New Resources Cluster** in the pop-up menu. Provide a descriptive name in the **Cluster Name** field and click **Save**. Before you can use the cluster, you must create a resources server in the Servers tab and assign this server to the resource cluster.

To assign VUI servers or clusters to use the resource cluster, click **Edit** beside a voice cluster and select the resource cluster in the **Resources Cluster**, and then click **Save**.

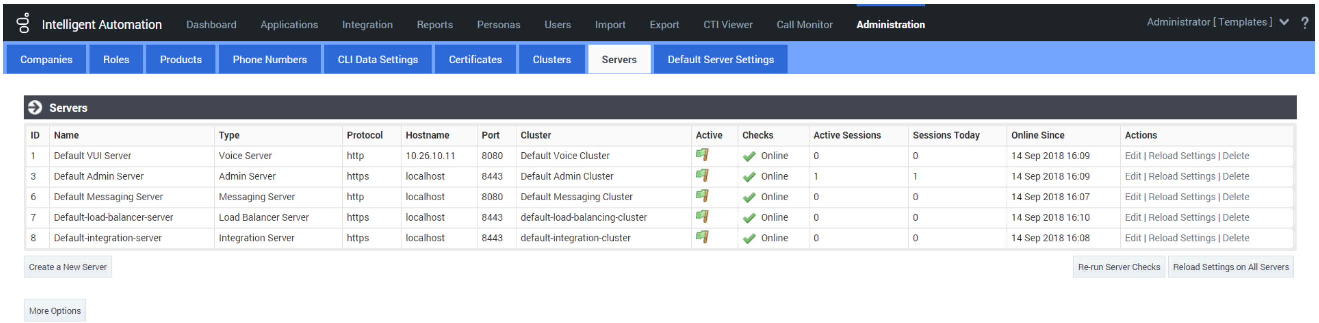
Other actions

In the **Clusters** list, you can select one of the following in the Actions column:

- **Edit** - Edit the cluster settings.
- **View Statistics** - View statistics about how many modules and applications are using the cluster.
- **Delete** - Delete the cluster from the Intelligent Automation environment.

Servers

The **Servers** tab allows you to monitor and manage the servers in your Intelligent Automation environment. You typically use this tab when taking a server off **Active** status so you can upgrade to a newer version of Intelligent Automation.



Servers

ID	Name	Type	Protocol	Hostname	Port	Cluster	Active	Checks	Active Sessions	Sessions Today	Online Since	Actions
1	Default VUI Server	Voice Server	http	10.26.10.11	8080	Default Voice Cluster	✓	✓ Online	0	0	14 Sep 2018 16:09	Edit | Reload Settings | Delete
3	Default Admin Server	Admin Server	https	localhost	8443	Default Admin Cluster	✓	✓ Online	1	1	14 Sep 2018 16:09	Edit | Reload Settings | Delete
6	Default Messaging Server	Messaging Server	http	localhost	8080	Default Messaging Cluster	✓	✓ Online	0	0	14 Sep 2018 16:07	Edit | Reload Settings | Delete
7	Default-load-balancer-server	Load Balancer Server	https	localhost	8443	default-load-balancing-cluster	✓	✓ Online	0	0	14 Sep 2018 16:10	Edit | Reload Settings | Delete
8	Default-integration-server	Integration Server	https	localhost	8443	default-integration-cluster	✓	✓ Online	0	0	14 Sep 2018 16:08	Edit | Reload Settings | Delete

The **Servers** list shows various information about each server, such as:

- The server type
- The protocol used by the server (HTTP or HTTPS)
- The server's hostname
- The port used by the server
- The cluster to which the server belongs
- Whether the server is active and able to process calls and other tasks
- Whether the server is online
- The number of active sessions currently, and the number of total active sessions today
- The date when the server came online.

Taking a server offline

When upgrading Intelligent Automation, you must take a server offline so it is not active and able to take news calls. Once a server is no longer active, it finishes its current calls but is no longer able to accept new calls. Monitor the **Active Sessions** value for the server, which tracks the number of calls currently being processed by the server. Once the **Active Sessions** value is **0**, the server has finished processing calls and you can take it offline for upgrading.

To take an active server offline:

1. Click **Edit** beside the server.
2. Clear the **Active** check box.
3. Click **Save**.

When you are ready to put the server online again, follow the steps above and ensure you select the **Active** check box, and then click **Save**. In the **Servers** list, the server has a green flag in the **Active** column to show the server is active and able to process calls.

Creating a server

You can create a server to add functionality to your Intelligent Automation environment. For example, you might add a Messaging server to support Web IVR, or another VUI server to handle future call loads.

Click **Create a New Server** and select a server type.

Enter the following information:

- **Server Name** - Enter a descriptive name for the server.
- **Server Connection Details** - Select a connection type (HTTP or HTTPS) and enter a hostname and port. If you select HTTPS and the server has a self-signed certificate, you must add that certificate to the certificate store by using the Certificates tab.
- **Cluster** - Specify which cluster to attach this server. Typically, this setting is unchanged from the default cluster.
- **Server Status** - Select the **Active** check box to make this server active and able to process requests.
- **Settings** - Add one or more server settings specific to this server. These settings supersede those set in the Default Server Settings tab.

Click **Save**.

> ### Important
> Remember to update the appropriate **local.properties** file for the server you create to reference the Server ID number in Intelligent Automation. For more information, refer to the Genesys Intelligent Automation Migration Guide.

Other actions

In the **Servers** list, you can select one of the following in the Actions column:

- **Edit** - Edit the server settings.

- **Reload Settings** - If you have made changes to this server, click this option to reload the settings.
- **Delete** - Delete the server from the Intelligent Automation environment.

The following two buttons are also available:

- **Re-run Server Checks** - Polls each server again to check its status.
- **Reload Settings on All Servers** - Reloads settings for all servers.

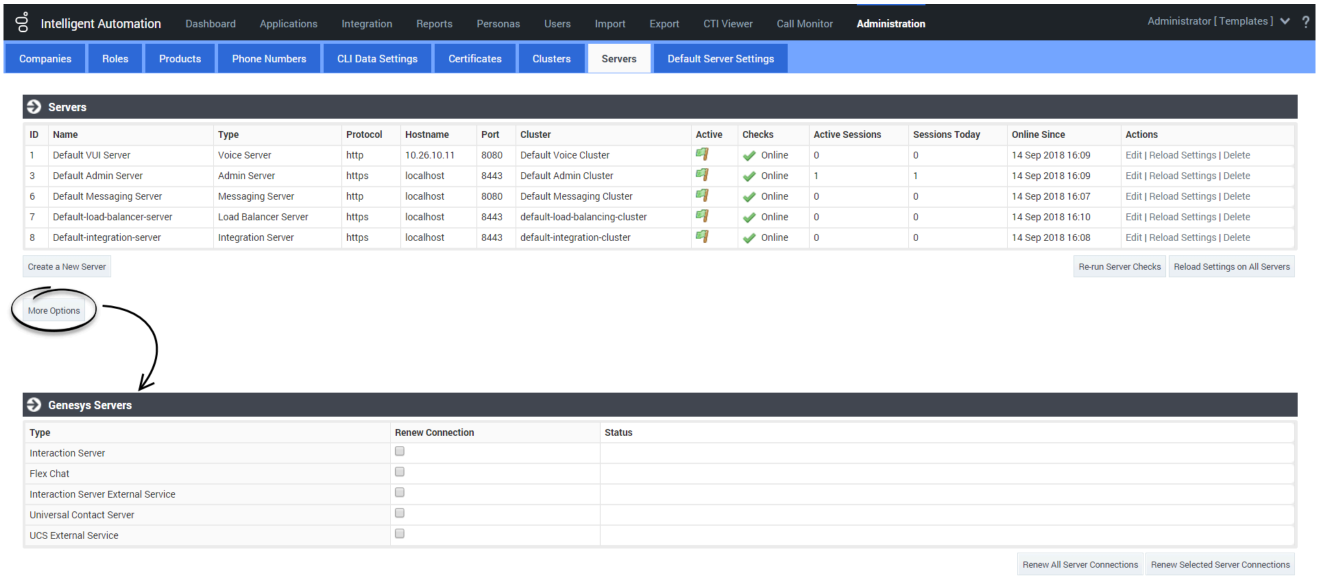
Renewing server connections

> Important
>
> - This feature applies only to the Genesys Engage platform.

You can renew the connections cache that Intelligent Automation uses to interact with Genesys servers. For example, this might be useful when you are testing or troubleshooting your environment, or if a backup server has become a primary server and Intelligent Automation needs to renew the connection.

Click **More Options** to access the **Genesys Servers** list. You can renew the following connections:

- Interaction Server
- Flex Chat (Chat Server)
- Interaction Server External Service
- Universal Contact Server
- UCS External Service



Servers

ID	Name	Type	Protocol	Hostname	Port	Cluster	Active	Checks	Active Sessions	Sessions Today	Online Since	Actions
1	Default VUI Server	Voice Server	http	10.26.10.11	8080	Default Voice Cluster		✓ Online	0	0	14 Sep 2018 16:09	Edit | Reload Settings | Delete
3	Default Admin Server	Admin Server	https	localhost	8443	Default Admin Cluster		✓ Online	1	1	14 Sep 2018 16:09	Edit | Reload Settings | Delete
6	Default Messaging Server	Messaging Server	http	localhost	8080	Default Messaging Cluster		✓ Online	0	0	14 Sep 2018 16:07	Edit | Reload Settings | Delete
7	Default-load-balancer-server	Load Balancer Server	https	localhost	8443	default-load-balancing-cluster		✓ Online	0	0	14 Sep 2018 16:10	Edit | Reload Settings | Delete
8	Default-integration-server	Integration Server	https	localhost	8443	default-integration-cluster		✓ Online	0	0	14 Sep 2018 16:08	Edit | Reload Settings | Delete

Genesys Servers

Type	Renew Connection	Status
Interaction Server	□	
Flex Chat	□	
Interaction Server External Service	□	
Universal Contact Server	□	
UCS External Service	□	

Renew All Server Connections
Renew Selected Server Connections

Renewing a connection does not disrupt an active connection. Rather, the connection cache is renewed so that Intelligent Automation uses the new connection cache data the next time it attempts to connect to the relevant servers.

Important

You can manage the timeout for renewing the Genesys server connections cache by configuring the default server setting **Caching.ReloadGenesysServerCaches.HttpTimeout**. The default value is 5000 milliseconds.

The **Genesys Servers** section shows the connections in use by Intelligent Automation. You can:

- Renew selected connections by selecting one or more checkboxes and clicking **Renew Selected Server Connections**.
- Renew all connections by clicking **Renew All Server Connections**.

Tip

For more information on some of the server connections used by Intelligent Automation, refer to these pages:

- User Authentication and Authorization
- Using WebIVR MicroApps
- Apple Business Chat

Default Server Settings

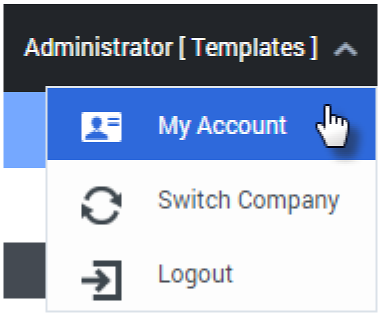
Default Server Settings manage various settings and configurations for how Intelligent Automation operates in your environment.

This section will list settings that you can configure. Updates to this section are made outside of the Intelligent Automation release cycle, so check back often for more information.

Contact your Genesys representative for questions about how to use Default Server Settings.

My Account

You can update your account information by clicking on your login name in the top-right corner of the screen and selecting **My Account** in the pop-up menu.



My Account Details

In the **My Account Details** tab, you can update the following:

- **Email Address** – The email address you use to log in to Genesys Intelligent Automation.
- **Name** – Your display name in Intelligent Automation.
- **Password** and **Confirm Password** – Change your current password. Enter your new password and then re-enter it to confirm that you have typed it correctly. You can leave these fields blank if you do not want to change your password.

Click **Update** to save your changes.

My Company Details

In the **My Company Details** tab, you can update the following:

- **Company Name** – The name of your company, which displays beside your username in the top-right corner of your screen.
- **Contact Email Address** – The email address for the main contact person within your company. Genesys uses this information to contact your company should any issues arise.
- **Contact Phone Number** – The phone number for the main contact person within your company. Genesys uses this information to contact your company should any issues arise.
- **Updated Company Logo** – Upload a new logo image for your company. This logo displays beside your Company Name in the top-right corner of your screen.

Click **Update** to save your changes.

Using iHub

This page describes how to set up Integration Processes to integrate the customer's backend resources, such as web services and databases, with Genesys Intelligent Automation.

Integration Hub is a simple and powerful way to do this, and brings many other benefits such as support for multiple environments' endpoints (dev/test/production, for example) and the ability to create automated test scripts. Using Integration Hub is a good way to keep your Intelligent Automation apps and their assorted presentation logic separate from the details of how to call onto your enterprise's backend systems.

Script blocks are great for adding 'presentation' logic to your app - comparing values and branching out to different blocks, performing date calculations, combining prompts. It's possible to invoke RESTful web services directly from a Script block, and this may be sufficient if the services are simple. It is advised not to the Script block for tasks like handling security certificates, or calling onto databases or SOAP-based web services

> **Tip**
>
> For information on working with Groovy, refer Apache Groovy documentation.

iHub Interface

The iHub interface is where you set up the Processes that integrate Intelligent Automation with the customer's resources, such as databases and web services.

iHub has four tabs, as follows:

- **Processes**—In this tab, you define the Integration Processes that will handle HTTP(S) and JDBC requests that are received from the Intelligent Automation VUI. Use the scripting commands described in Scripting Commands to write any Process and Library scripts.
- **Deploy to Production**—After you have created and tested your Processes, you put them into production using this tab. Deployments in the current calendar year are displayed in tabular and calendar (weekly) format. From the table of deployments, you can also rollback to previous productions if needed.
- **Import**—You can import Process scripts and Shared scripts from other Processes.
- **Export**—You can export Process scripts, including the latest production versions, and library entries into a compressed (**.zip**) file; compatible for use with the Import tab.

Processes

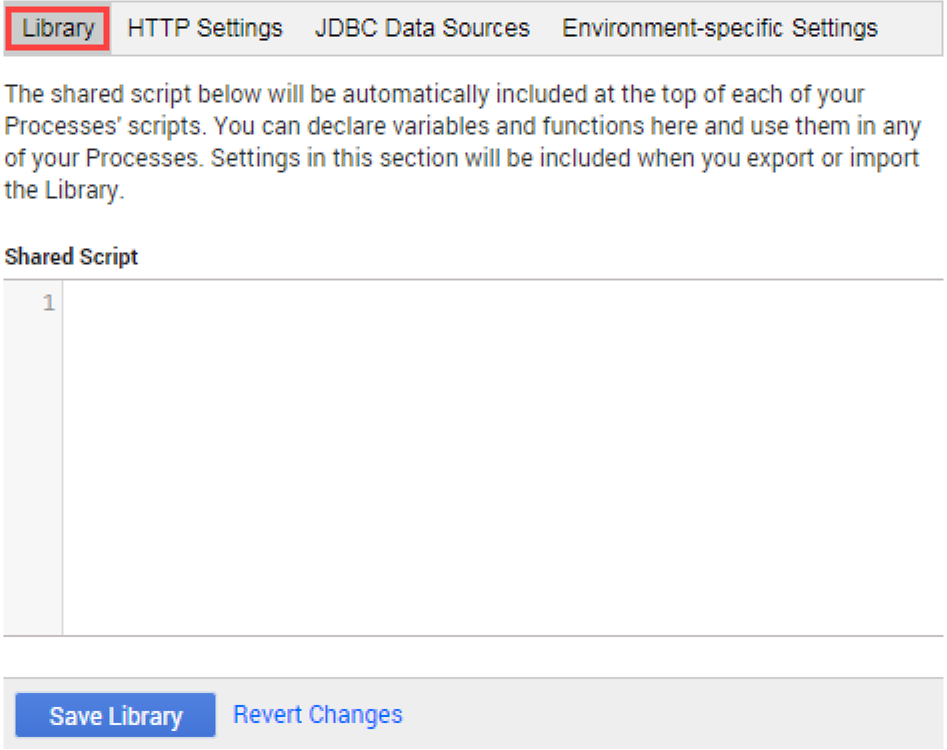
Defining Common Properties

To simplify your work, Intelligent Automation allows you to specify common properties, such as variables, functions, and environment settings, that can be used for all of your processes as required. In addition, if any Company-specific configuration information applies to a specific callflow engine and resides on a web site or in a database, you can specify the location and access credentials of the site or database to which the VUI must connect, even securing the connection if necessary. You can do all this on the right-hand panel of the Processes tab.

> Tip
>
> These settings can be modified or deleted at any time by opening the **Processes** tab and making the necessary changes.

Library tab

On the Library tab, create a list of additional variables and functions that can be shared between Processes. The Library is inserted at the top of a Process script, so you can reference the variables and functions throughout the script, as needed.



The shared script below will be automatically included at the top of each of your Processes' scripts. You can declare variables and functions here and use them in any of your Processes. Settings in this section will be included when you export or import the Library.

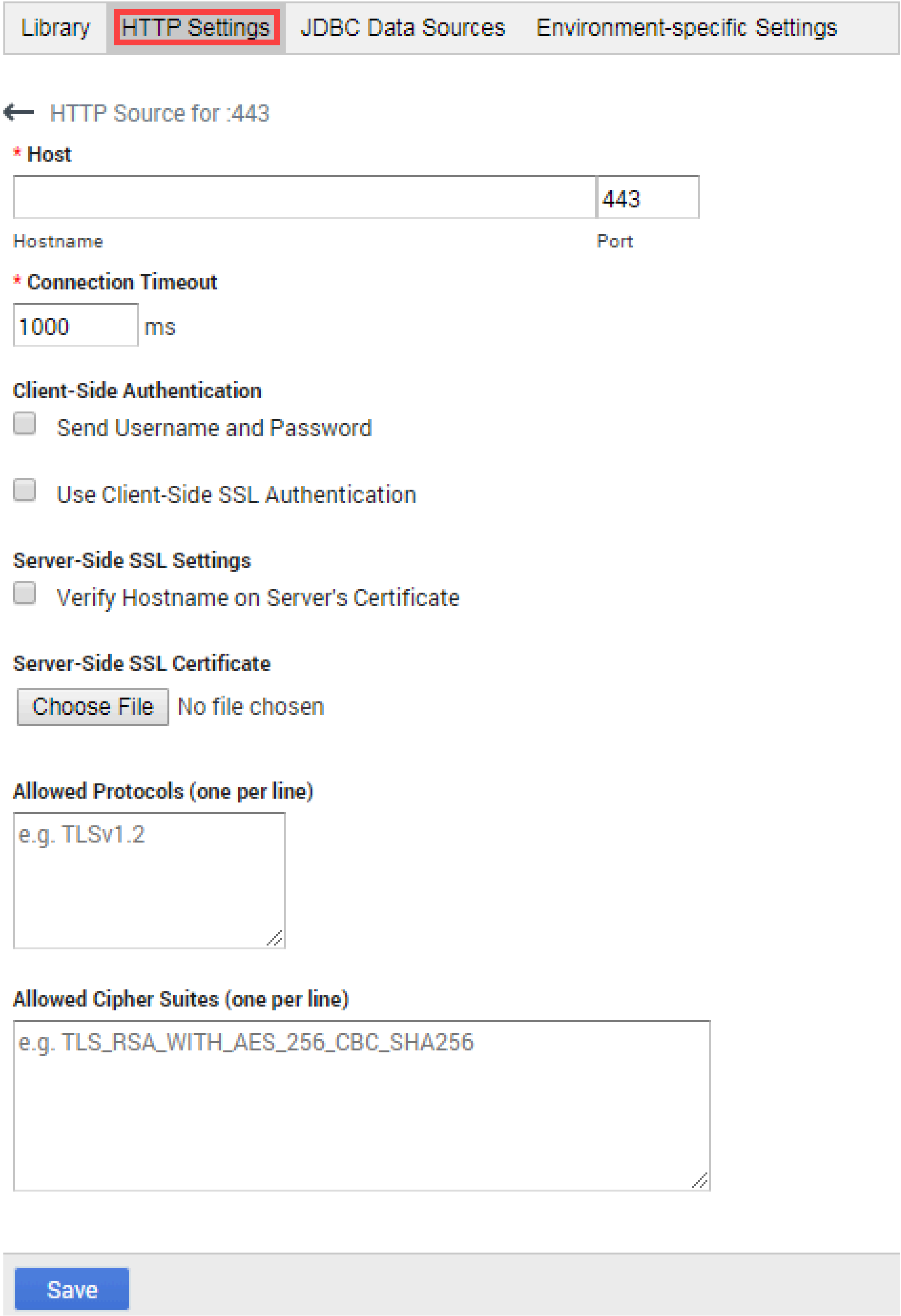
Shared Script

HTTP Settings tab

On the HTTP Settings tab, provide the Company-specific configuration information that applies to a given Intelligent Automation callflow engine installation on a website.

You can set up multiple secure connections between Intelligent Automation and external websites.

Enter only a **Hostname**, **Port** (defaults to the port used by the installation in which you are working), and the **Connection Timeout**. If you want to set up a secure connection, you will have to provide additional information. For more information about the security concepts, refer to the *Genesys Security Deployment Guide*.



HTTP Source for :443

* Host

443

Hostname Port

* Connection Timeout

1000 ms

Client-Side Authentication

□ Send Username and Password

□ Use Client-Side SSL Authentication

Server-Side SSL Settings

□ Verify Hostname on Server's Certificate

Server-Side SSL Certificate

Choose File No file chosen

Allowed Protocols (one per line)

e.g. TLSv1.2

Allowed Cipher Suites (one per line)

e.g. TLS_RSA_WITH_AES_256_CBC_SHA256

When calling an external web site using Groovy, the list of trusted SSL certificates must be provided. Intelligent Automation supports both client-side authentication and server-side authentication.

The **Send Username and Password** option allows access to external servers.

When using **Client-Side Authentication**, Intelligent Automation allows uploading the client certificate including the public key certificate and a private key along with the password. IA stores this certificate information and uses it for verification.

You can upload a server certificate also. The **Verify Hostname on Server's Certificate** option verifies if the host name matches with the certificate.

IA supports both client-side and server-side authentication individually and also together.

JDBC Data Sources tab

If any of this Company-specific information resides in a database or in SQL connection pools, define the JDBC parameters on this tab. The contents of this tab can differ depending on what you specify in the **Database Type** field, and by what you enter in other fields. Enter the required information in this tab, noting the following:

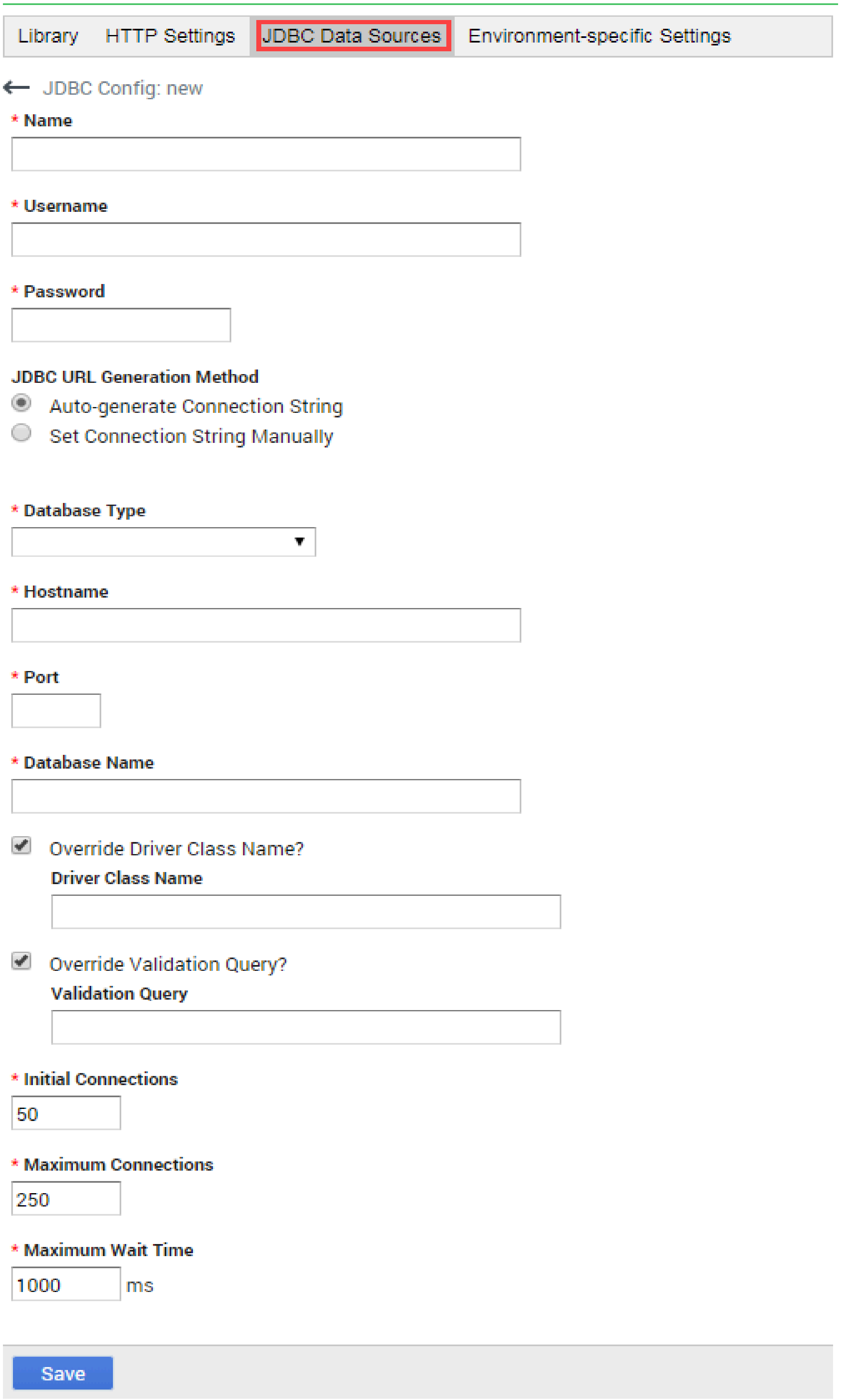
- The entry in the **Name** field is unique, but is for internal use only.
- You can enter the JDBC URL manually by selecting **Set Connection String Manually** or have Intelligent Automation construct it from the information you enter in the fields that open when you click **JDBC URL Generation Method**.
- The are four supported **Database Type**s—SQL Server, PostgreSQL, MySQL, and Oracle (using either a System ID or a Service Name).

> **Warning**
>
> Genesys does not ship with a driver for MySQL and PostgreSQL, so if you are using the MySQL or PostgreSQL DBMS, you will have to obtain that separately.

- Enter a driver name in **Override Driver Class Name** only if you need to override the driver class name, for example if you are using an older or custom version of a driver.
- If you do not want to use the standard validation query specific to the database type (for example, `SELECT 1` for SQL Server) that is used by the connection pool to determine if the connection is working, you can specify a custom one, perhaps based on the Company name, in **Override Validation Query**.
- The **Initial Connections** field indicates the number of connections that are made initially when connecting to the database. The allowed permissible range is `0` to the value configured in **Indy.ConnectionPooling.JDBC.MaxSize**.
- The **Maximum Connections** field indicates the maximum number of connections to the database. The default value is `10`.

> **Important**
>
> To increase the maximum number of connections, modify the **Indy.ConnectionPooling.JDBC.MaxSize** parameter (**Administration** > **Default Server Settings**). The allowed permissible range is `0` to the value configured in **Indy.ConnectionPooling.JDBC.MaxSize**.

- The **Maximum Wait Time** field (or maximum time to wait for a pool connection) is set by default as 1 and maximum of `300000` millisseconds.



← JDBC Config: new

* Name

* Username

* Password

JDBC URL Generation Method

● Auto-generate Connection String

○ Set Connection String Manually

* Database Type

* Hostname

* Port

* Database Name

☑ Override Driver Class Name?

Driver Class Name

☑ Override Validation Query?

Validation Query

* Initial Connections: 50

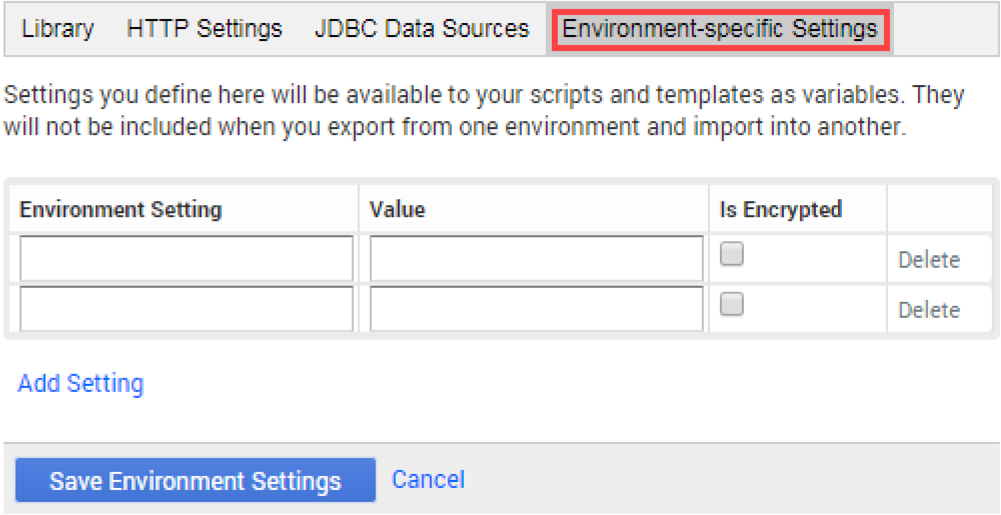
* Maximum Connections: 250

* Maximum Wait Time: 1000 ms

Environment-specific Settings tab

In this tab, define any settings that are specific to the environment in which you are working, such as Lab, Test, or Production.

Variables used during processing (like hostname and port) can be declared here and accessed by Groovy/Java scripts later on.

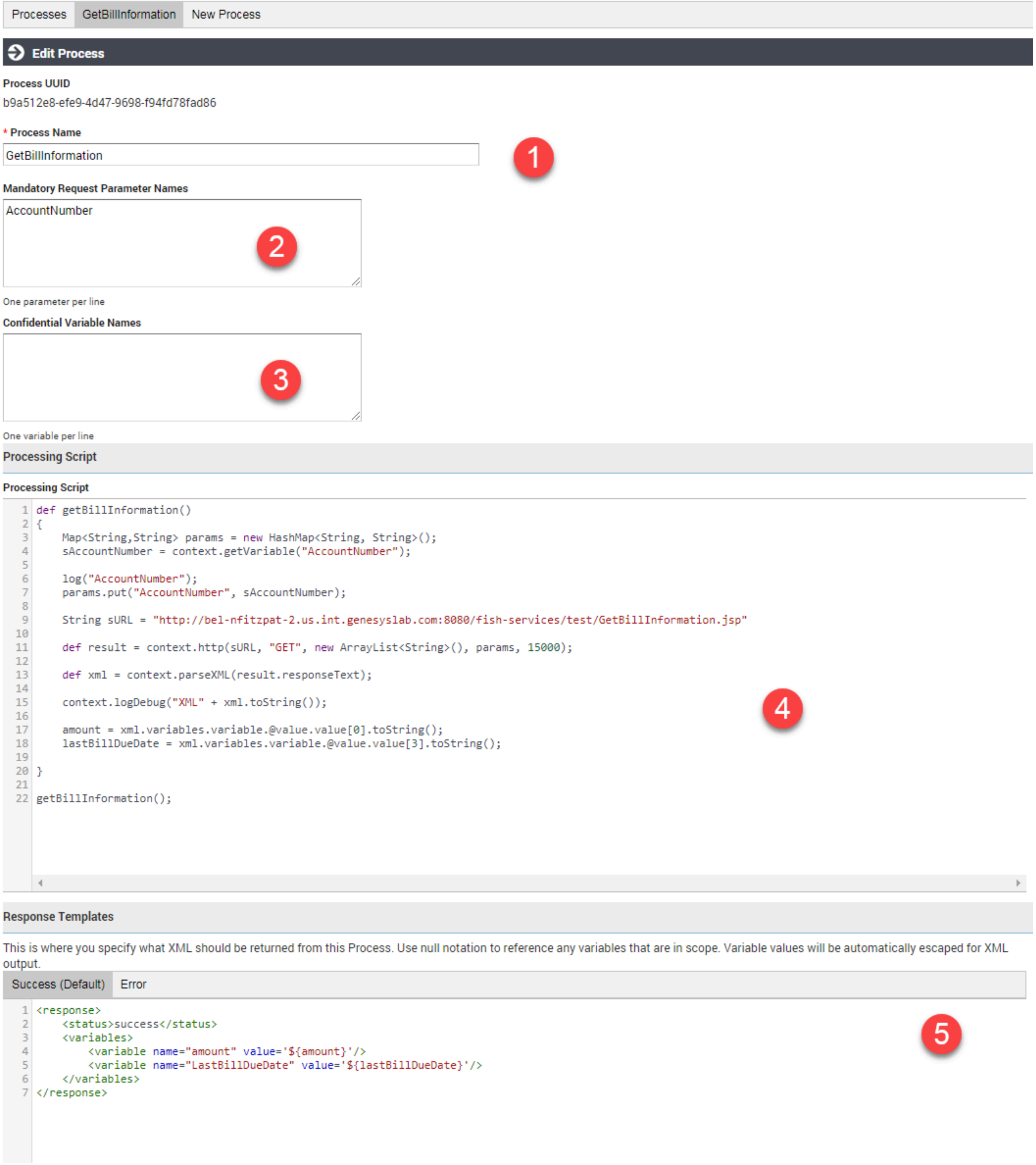


Settings you define here will be available to your scripts and templates as variables. They will not be included when you export from one environment and import into another.

Environment Setting	Value	Is Encrypted	
		□	Delete
		□	Delete

Creating a Process

Create a new Process in the **Processes** tab, as indicated in the following diagram:



Process UUID
b9a512e8-efe9-4d47-9698-f94fd78fad86

* Process Name
GetBillInformation

Mandatory Request Parameter Names
AccountNumber

One parameter per line

Confidential Variable Names

One variable per line

Processing Script

Processing Script

```
def getBillInformation()
{
    Map<String,String> params = new HashMap<String, String>();
    sAccountNumber = context.getVariable("AccountNumber");

    log("AccountNumber");
    params.put("AccountNumber", sAccountNumber);

    String sURL = "http://bel-nfitzpat-2.us.int.genesyslab.com:8080/fish-services/test/GetBillInformation.jsp"

    def result = context.http(sURL, "GET", new ArrayList<String>(), params, 15000);

    def xml = context.parseXML(result.responseText);

    context.logDebug("XML" + xml.toString());

    amount = xml.variables.variable.@value.value[0].toString();
    lastBillDueDate = xml.variables.variable.@value.value[3].toString();

}

getBillInformation();
```

Response Templates

This is where you specify what XML should be returned from this Process. Use null notation to reference any variables that are in scope. Variable values will be automatically escaped for XML output.

Success (Default) | Error

```
<response>
    <status>success</status>
    <variables>
        <variable name="amount" value='${amount}'/>
        <variable name="LastBillDueDate" value='${lastBillDueDate}'/>
    </variables>
</response>
```

After you click **New Process** in the tab's header bar or click the **Create a New Process** button:

1. Enter a unique **Process Name**. Intelligent Automation automatically assigns a **Process UUID** for a new

process. This UUID stays with this Process until the Process is deleted.

2. Enter any parameters that will be required in the request from VUI. These parameters will be used by the process as it is executing. If the request does not contain these parameters, it will be rejected.
3. Enter any confidential parameters. These parameters might contain customer-sensitive data, such as passwords or credit card numbers, and will not be written to logs or reporting databases.
4. Using the scripting commands provided with Intelligent Automation, enter the code for the script that will process the request.
5. Create the **Response Template** that will be returned by this Process.You must create at least one Response Templates - one for successful Process execution (**Success**). A Response Template for unsuccessful execution (**Error**) is optional.

Modifying a Saved Process

To modify a saved (existing) process, even its name, click Processes on the **Processes** tab to view the list of Processes. Select it by name on the Processes tab, and make the appropriate changes.

Deleting an Existing Process

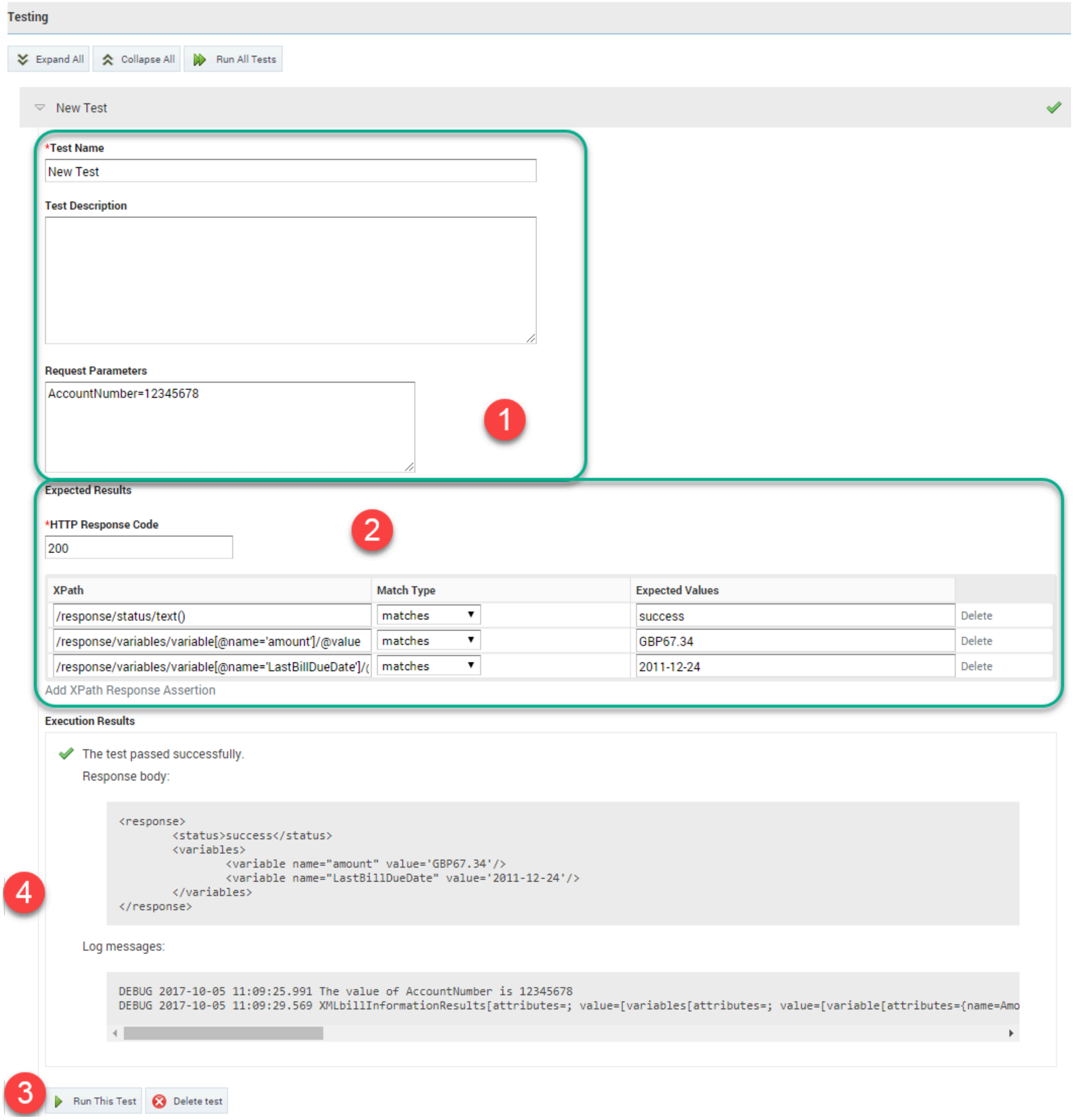
To delete a process, click Processes on the **Processes** tab to view the list of Processes. Select the process you want to delete in the list, and click the garbage can icon to delete it.

Testing a Process

iHub allows you to test your Processes by defining test cases and running them against any Processes you have created, *before* the Processes are put into production. You can also test a production version by copying the currently deployed version of a Process to the test side of the environment.

> ### Warning
>
> If a Process by that name already exists, the test version will be overwritten by the copied production version, and any undeployed changes to the original test version will be lost.

Refer to the following diagram to create, run, and examine the results of a test case. The numbers in the diagram correspond to the tasks that follow.



*Test Name

New Test

Test Description

Request Parameters

AccountNumber=12345678

Expected Results

*HTTP Response Code

200

XPath	Match Type	Expected Values	
/response/status/text()	matches ▼	success	Delete
/response/variables/variable[@name='amount']/@value	matches ▼	GBP67.34	Delete
/response/variables/variable[@name='LastBillDueDate']/(	matches ▼	2011-12-24	Delete

Add XPath Response Assertion

Execution Results

✓ The test passed successfully.

Response body:

```
<response>
        <status>success</status>
        <variables>
                <variable name="amount" value='GBP67.34'/>
                <variable name="LastBillDueDate" value='2011-12-24'/>
        </variables>
</response>
```

Log messages:

```
DEBUG 2017-10-05 11:09:25.991 The value of AccountNumber is 12345678
DEBUG 2017-10-05 11:09:29.569 XMLbillInformationResults[attributes=; value=[variables[attributes=; value=[variable[attributes={name=Amo
```

Creating a Test Case

After clicking **New Test Case**:

1. Define the test case by giving it a specific **Test Name**, an optional **Test Description**, and values of any **Request Parameters** required by the process.

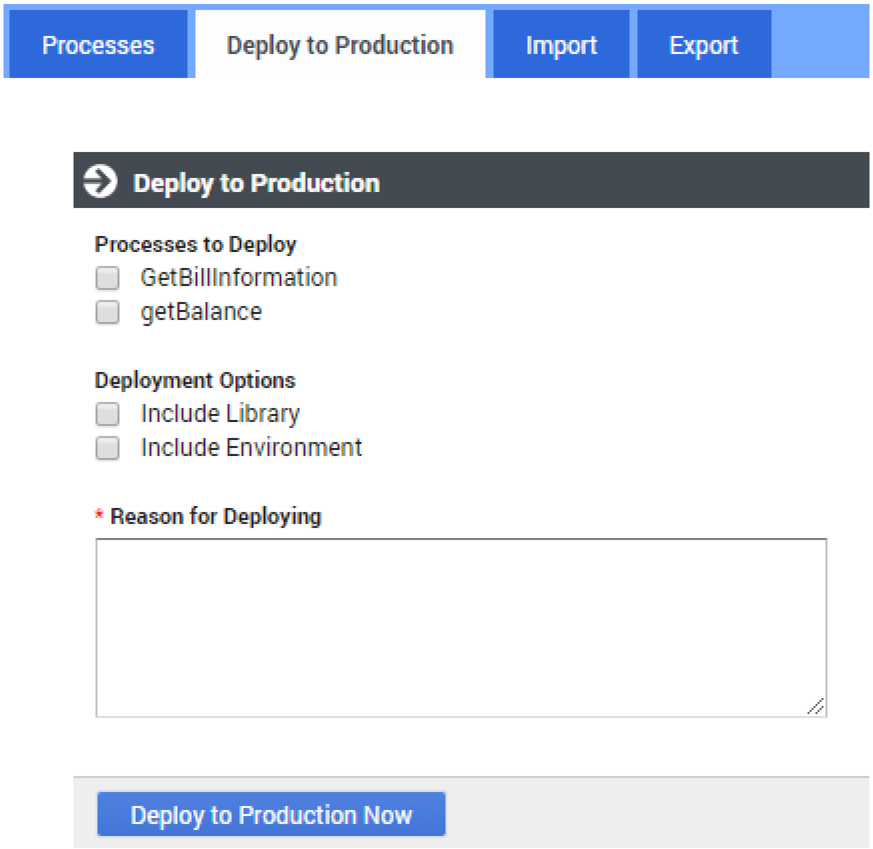
2. Specify the expected **HTTP Response Code** (the default, `200`, indicates success); and optionally, the **XPath Response Assertions**, in which known elements or variables should match or not match specified values. The variety of assertions is based on whatever data is returned by the test stubs. The Author can write as many XPath assertions as they wish to inspect the generated XML, as long as he or she knows what data is returned by a test web service for a particular input. For example:
   - **XPath** is `/response/status/text()`
   - **Match Type** is `matches`
   - **Expected Values** is `success`

Running and Evaluating a Test Case

1. Click **Run this Test**.
2. View the **Execution Results**, including the information from any Response Templates associated with this process.

Deploy to Production

After you have created the Process and tested it, it is ready to be put into the production side of the environment in which you are working. Note that when you deploy a process to production, it is not changed nor removed from the testing environment.



Deploy to Production

Processes to Deploy

□ GetBillInformation

□ getBalance

Deployment Options

□ Include Library

□ Include Environment

* Reason for Deploying

To deploy a process to production:

1. Select the **Processes to Deploy**. If there is more more than one listed, you can deploy one, some, or all of those that are listed.
2. **Deployment Options** give you the option to include the Library and/or Environment information that you entered on the **Processes** tab.

> ## Tip
> If you included any of the library variables or environment information in any of the processes to be deployed, you must include the corresponding deployment option here.

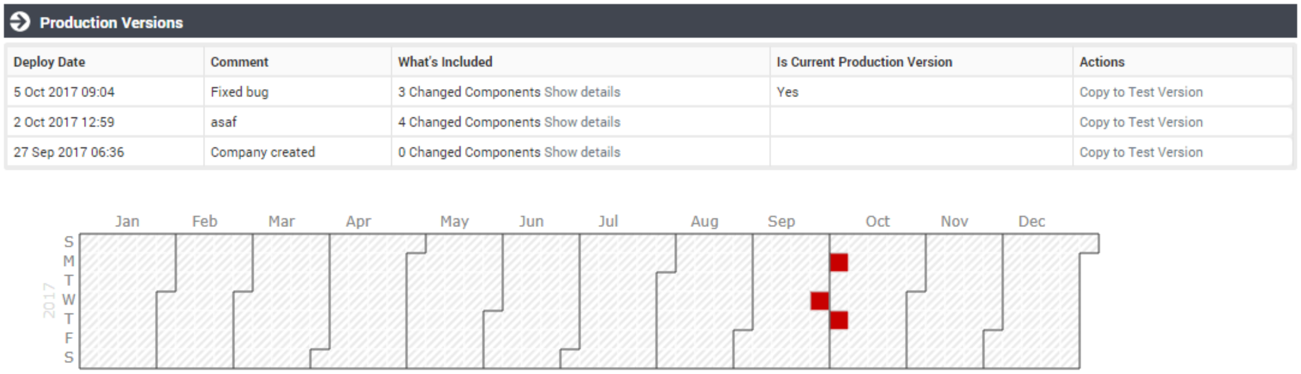
3. You must provide a **Reason for Deployment**, if for no other reason than to identify this particular deployment.

The **Production Version** section (see the diagram below) displays a history of your deployment activities, including which deployment is your **Current Production Version**.

You can also copy a deployment to a test version, effectively overwriting the test version with a copy of the deployed version.

> **Tip**
>
> If your current deployment is not operating as expected, you can copy the previously deployed version to the test side of the environment, test it to make sure that it is running properly, then redeploy this version. This version then becomes the Current Production Version, taking over this position from the more updated, but faulty version. Then, you can work to debug the latest (updated but faulty) version, and deploy it when tested.

The twelve-month calendar for the current year illustrates the distribution of all deployments throughout the current calendar year. Hovering over a red box displays the date and number of deployments on that date; clicking on the red box highlights the deployment in the table above the calendar.



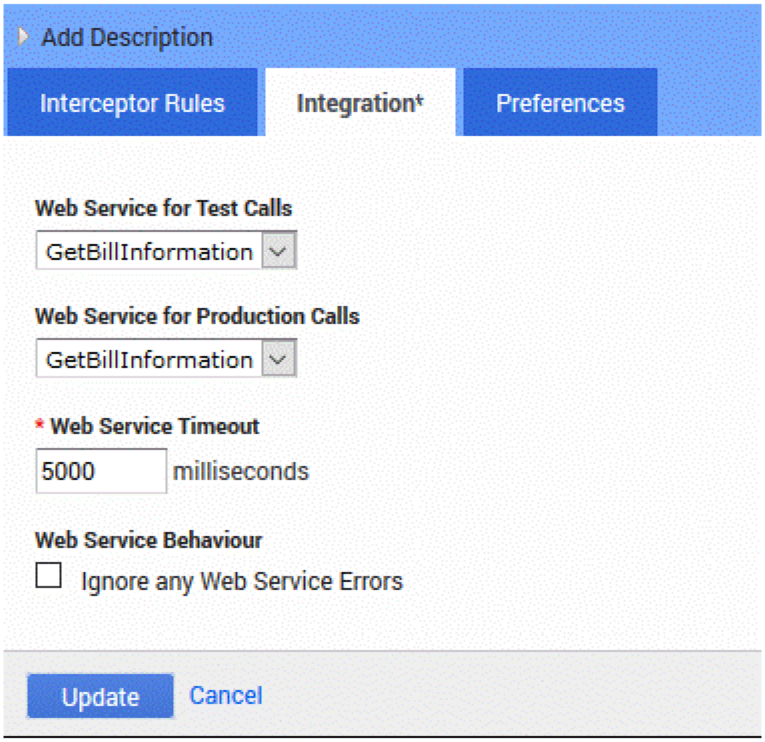
Production Versions

Deploy Date	Comment	What's Included	Is Current Production Version	Actions
5 Oct 2017 09:04	Fixed bug	3 Changed Components Show details	Yes	Copy to Test Version
2 Oct 2017 12:59	asaf	4 Changed Components Show details		Copy to Test Version
27 Sep 2017 06:36	Company created	0 Changed Components Show details		Copy to Test Version

> **Warning**
>
> The Copy-to-test feature in iHub replaces all test processes with the production processes that was in the specific deployment.

Adding Integration Processes to a Callflow

After you have created and tested an Integration Process, you can incorporate it into your callflow by adding it to an Interceptor block, as shown in the following diagram:



Web Service for Test Calls
GetBillInformation

Web Service for Production Calls
GetBillInformation

* Web Service Timeout
5000 milliseconds

Web Service Behaviour
□ Ignore any Web Service Errors

Open the **Integration** tab of the Interceptor block, and do the following:

1. Replace the values of the **Web Service URL for Test Calls** and **Web Service URL for Production Calls** fields with the following:

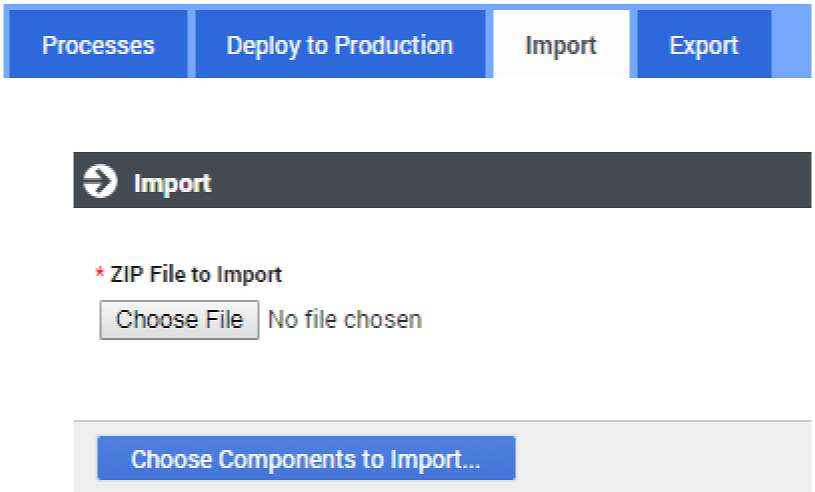
   ```
   ihub://<Process_ID>
   ```

   Where `Process_ID` is the Process UUID shown at the top of the **Processes** tab. You can also add additional parameters, if required.
2. Change the value of the **Web Service Timeout** field, if needed.

Import

On the **Import** tab, you can import iHub Scripts from other processes in the same or a different environment. The input file must be created by using the Export tab.

After you have selected the **.zip** files for importing, click **Choose Components for Import**. You can then choose to use the file contents to create a new process, or to overwrite the corresponding parts of an existing process.

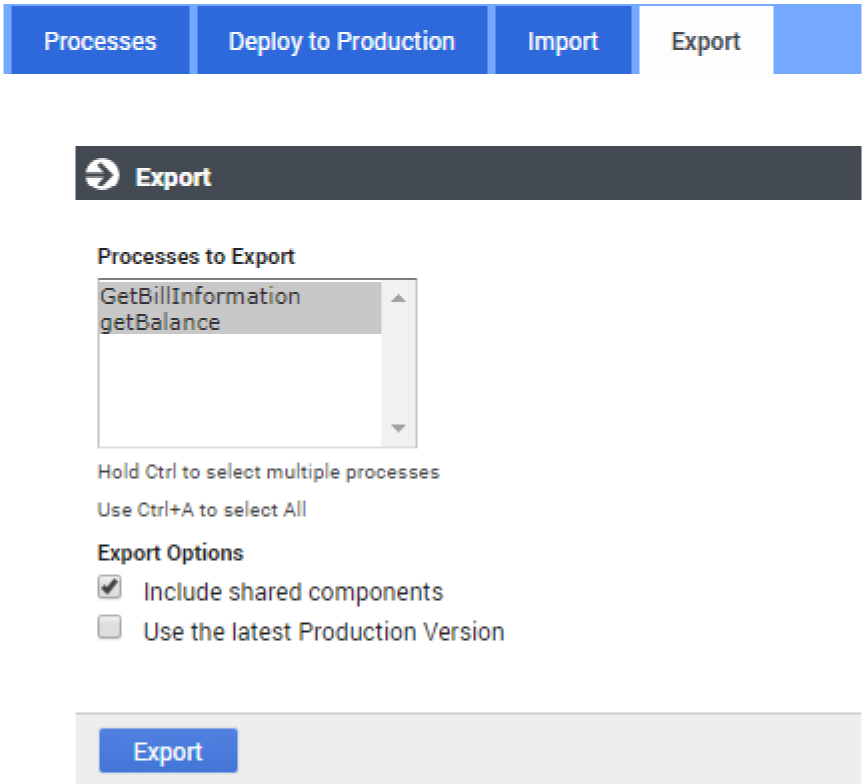


* ZIP File to Import

No file chosen

Export

On the **Export** tab, you can export processes and shared scripts to a **.zip** file. This file can be used for such things as backup or storage, or to import back into iHub, perhaps into another environment.



Processes to Export

GetBillInformation
getBalance

Hold Ctrl to select multiple processes

Use Ctrl+A to select All

Export Options

☑ Include shared components

☐ Use the latest Production Version

You must select one or more processes and/or one or more shared component to export. If any process you selected is also in production, you can also choose to export the latest production version of it instead of the test version.

When you click **Export**, the export operation creates a file called **Integration Export <datestamp> <timestamp>.zip**. This file can then be imported using the Import tab.

Scripting Commands

> **Important**
>
> You can download a JavaDoc version of the scripting commands here.

iHub provides the following commands for use in Process and Library Scripts:

Command	Descriptionhere
context.cacheGet(java.lang.String sKey_p)	Retrieve an item placed in the user cache.
context.cachePut(java.lang.String sKey_p, java.io.Serializable value_p, int ExpiryTimeSecs_p)	Place an item in the user cache.
context.escapeJavaScript(java.lang.String sText_p)	Free up memory used by the given JavaScript string.
context.escapeUrl(java.lang.String sText_p)	Free up memory used by the given URL.
context.escapeXml(java.lang.String sText_p)	Free up memory used by the given XML string.
context.formatCurrency(java.lang.String sCurrencyCode_p, java.math.BigDecimal currencyAmount_p)	Create a correctly formatted currency string by passing in an ISO 4217 currency code and a BigDecimal amount.
context.formatCurrency(java.lang.String sCurrencyCode_p, java.lang.String sCurrencyAmount_p)	Create a correctly formatted currency string by passing in an ISO 4217 currency code and a String amount.
context.formatDate(java.util.Date date_p)	Format a date in the standard Intelligent Automation format (yyyy-mm-dd) using the system timezone.
context.formatDate(java.util.Date date_p, java.lang.String sFormat_p)	Format a date into the standard Intelligent Automationformat (i.e yyyy-MM-dd)
context.formatDate(java.util.Date date_p, java.time.ZoneId timezone_p)	Format a date into a specified format using the system timezone.
context.formatDate(java.util.Date date_p, java.lang.String sFormat_p, java.time.ZoneId timezone_p)	Format a date into a specified format with a specified timezone.
context.getCurrencyAmount(java.lang.String sSpeechStormCurrency_p)	Retrieve the amount from an Intelligent Automation formatted currency string.
context.getCurrencyCode(java.lang.String sSpeechStormCurrency_p)	Retrieve the ISO 4217 from an Intelligent Automation formatted currency string.

Command	Descriptionhere
context.getLastBackendCallResult()	Return the result of the last backend request.
context.getRandomPercentage()	Return a random percentage value that can be used for routing a given number of calls in different directions.
context.getResponseTemplateNames()	Return the set of Response Template names that are configured in the iHub
context.getTimeZone(java.lang.String sTimeZoneName_p)	Return the ZoneId value of a specified timezone name.
context.getVariable(java.lang.String sName_p)	Return the value of a variable held in session by it's name
context.getVariableNames()	Return a Collection of the variable names currently held in session.
context.http(java.lang.String sURL_p, java.lang.String sMethod_p, java.util.List<java.lang.String> headers_p, java.util.Map<java.lang.String,java.lang.String> params_p, int iTimeoutMillis_p)	Send an HTTP request to a specified web service URL containing key/value pair parameters. Important If the specified method requires information to be sent in the body rather than in the query string, iHub automatically adds the parameters to the request body as key/value pairs. The port number listed in the HTTP Settings must be included in the **sURL_p** string, e.g., https://service.domain.com:443/application.
context.http(java.lang.String sURL_p, java.lang.String sMethod_p, java.util.List<java.lang.String> headers_p, java.lang.String sRequestBody_p, java.lang.String sContentType_p, int iTimeoutMillis_p)	Send an HTTP request to a specified web service URL containing a request body and content type. Important When a context.http call is used, an SSL handshake is performed to establish a secure connection (with TLS) with the configured host URL. The port number listed in the HTTP Settings must be included in the **sURL_p** string, e.g., https://service.domain.com:443/application.
context.logDebug(java.lang.String sMessage_p, java.lang.Object... additionalItems_p)	Write a debug statement to the logs.
context.logError(java.lang.String sMessage_p, java.lang.Object... additionalItems_p)	Write an error statement to the logs.
context.logError(java.lang.Throwable error_p, java.lang.String sMessage_p, java.lang.Object... additionalItems_p)	Write an error statement to the logs including a Throwable object.
context.logInfo(java.lang.String sMessage_p, java.lang.Object... additionalItems_p)	Write an info statement to the logs.
context.logWarning(java.lang.String sMessage_p, java.lang.Object... additionalItems_p)	Write a warning statement to the logs.
context.parseDate(java.lang.String sSpeechStormDate_p)	Parse a date in the Intelligent Automation format (yyyy-MM-dd) using the system time zone.
context.parseDate(java.lang.String sDate_p, java.lang.String sFormat_p, java.time.ZoneId timezone_p)	Parse a date in the Intelligent Automation format (yyyy-MM-dd) using the given time zone.
context.parseDate(java.lang.String sDate_p,	Parse a date in a specified format using the system

Command	Descriptionhere
java.lang.String sFormat_p)	time zone.
context.parseDate(java.lang.String sDate_p, java.lang.String sFormat_p, java.time.ZoneId timezone_p)	Parse a date with a specified format and timezone.
context.parseJSON(java.lang.String sJSON_p)	Parse a provided JSON string into a JSON object.
context.parseXML(java.lang.String sXML_p)	Parse a provided XML string into a Node object.
context.selectResponseTemplate(java.lang.String sResponseTemplateName_p)	Specify the response template to use when responding to the VUI.
context.sendEmail(java.util.List<java.lang.String> recipients_p, java.lang.String sFromAddress_p, java.lang.String sSubject_p, java.lang.String sMessage_p)	Send an email.
context.sendSMS(java.lang.String sRecipientNumber_p, java.lang.String sSenderNumber_p, java.lang.String sMessage_p)	Send an SMS.
context.setVariable(java.lang.String sName_p, java.lang.Object value_p)	Add a variable to the session.
context.sqlDelete(java.lang.String sConnectionName_p, int iTimeoutMillis_p, java.lang.String sQuery_p, java.lang.Object... parameters_p)	Perform a DELETE query in the database.
context.sqlInsert(java.lang.String sConnectionName_p, int iTimeoutMillis_p, java.lang.String sQuery_p, java.lang.Object... parameters_p)	Perform an INSERT query in the database.
context.sqlSelect(java.lang.String sConnectionName_p, int iTimeoutMillis_p, java.lang.String sQuery_p, java.lang.Object... parameters_p)	Perform a SELECT query in the database.
context.sqlStoredProcedure(java.lang.String sConnectionName_p, int iTimeoutMillis_p, java.lang.String sStoredProcedureName_p, java.lang.Object... parameters_p)	Run the specified stored procedure.
context.sqlUpdate(java.lang.String sConnectionName_p, int iTimeoutMillis_p, java.lang.String sQuery_p, java.lang.Object... parameters_p)	Perform a UPDATE query in the database.
context.unescapeJavaScript(java.lang.String sText_p)	Allocate memory for the given JavaScript string.
context.sqlDelete(String sConnectionName_p, int iTimeoutMillis_p, String sQuery_p, List<Object> parameters_p)	Perform a DELETE query in the database.
context.sqlInsert(String sConnectionName_p, int iTimeoutMillis_p, String sQuery_p, List<Object> parameters_p)	Perform an INSERT query in the database.
context.sqlSelect(String sConnectionName_p, int iTimeoutMillis_p, String sQuery_p, List<Object> parameters_p)	Perform a SELECT query in the database.

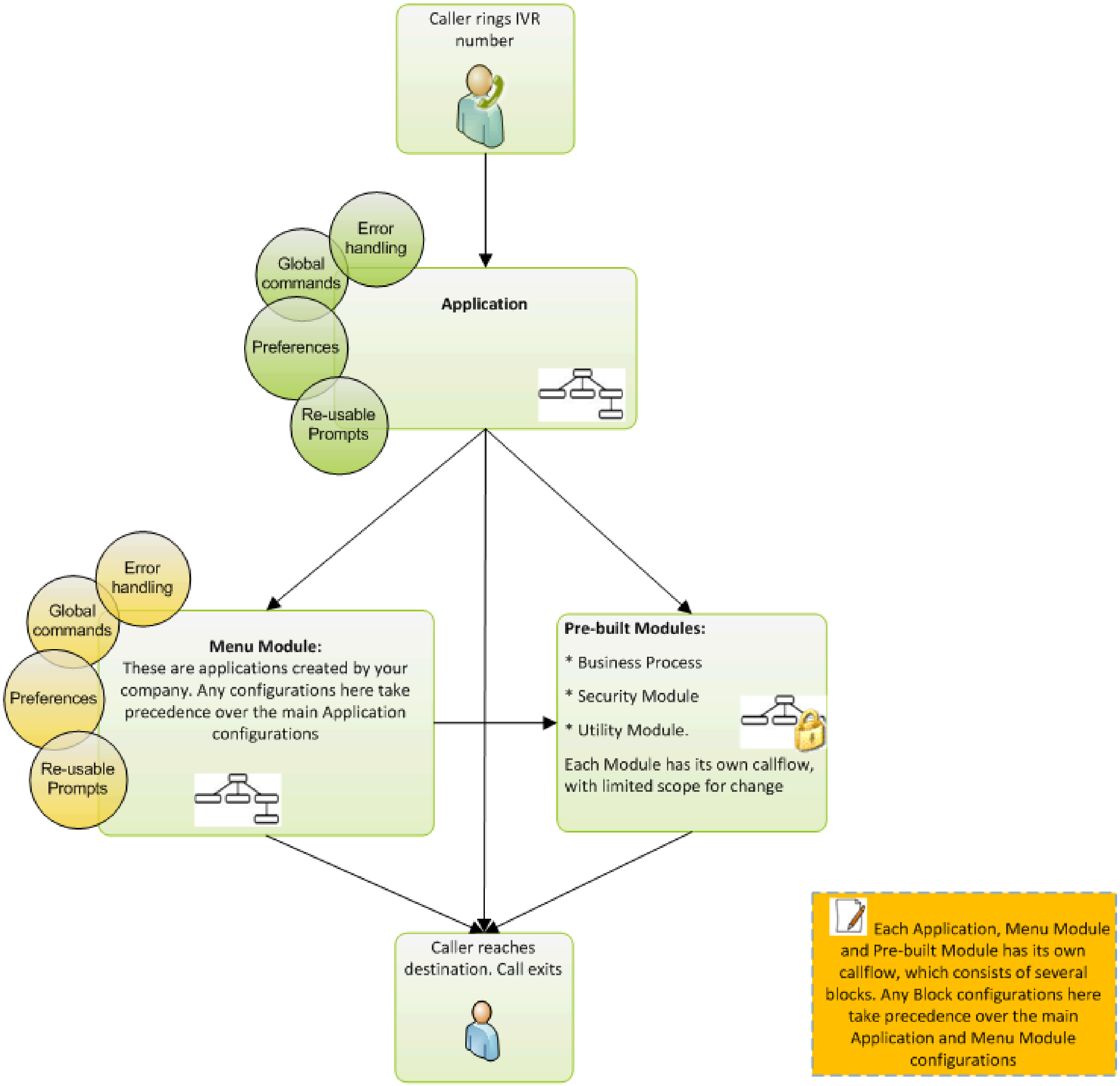
Command	Descriptionhere
context.sqlUpdate(String sConnectionName_p, int iTimeoutMillis_p, String sQuery_p, List<Object> parameters_p)	Perform an UPDATE query in the database
context.sqlStoredProcedure(String sConnectionName_p, int iTimeoutMillis_p, String sStoredProcedureName_p, List<Object> parameters_p)	Run the specified stored procedure.

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:


Caller rings IVR number
Error handling
Global commands
Preferences
Re-usable Prompts
Application
Error handling
Global commands
Preferences
Re-usable Prompts
Menu Module:
These are applications created by your company. Any configurations here take precedence over the main Application configurations
Pre-built Modules:
* Business Process
* Security Module
* Utility Module.
Each Module has its own callflow, with limited scope for change
Caller reaches destination. Call exits
Each Application, Menu Module and Pre-built Module has its own callflow, which consists of several blocks. Any Block configurations here take precedence over the main Application and Menu Module configurations

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.

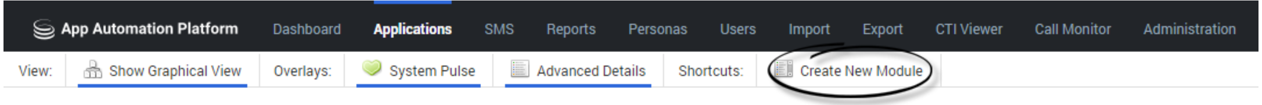
> **Important**
>
> Refer to Understanding settings inheritance for information on the order of

> precedence for each of these aspects.

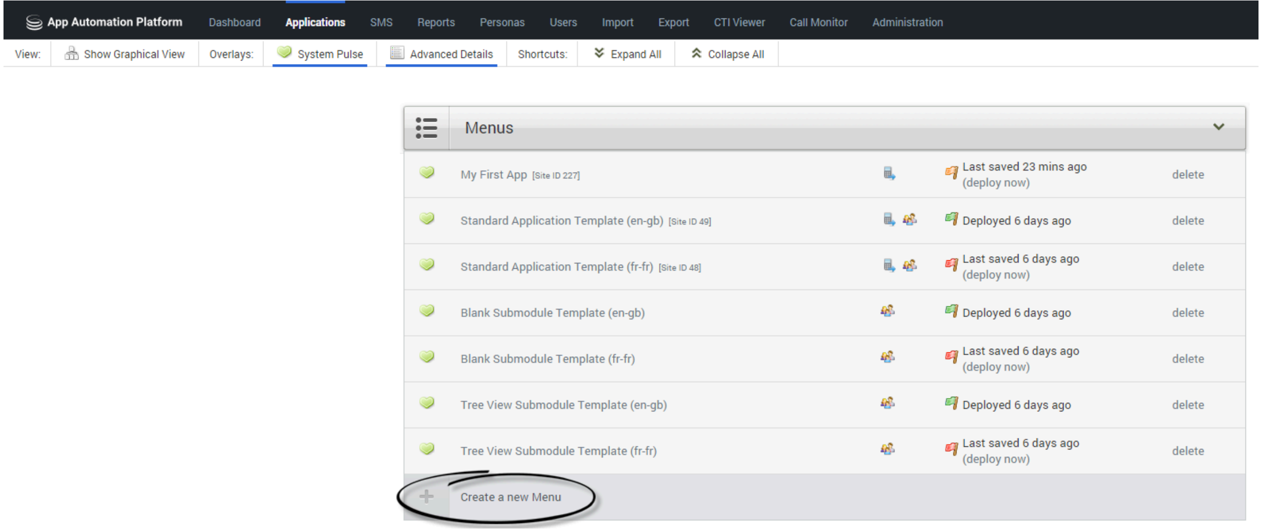
Creating applications and modules

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**.



My First App [Site ID 227]	Last saved 23 mins ago (deploy now)	delete
Standard Application Template (en-gb) [Site ID 49]	Deployed 6 days ago	delete
Standard Application Template (fr-fr) [Site ID 48]	Last saved 6 days ago (deploy now)	delete
Blank Submodule Template (en-gb)	Deployed 6 days ago	delete
Blank Submodule Template (fr-fr)	Last saved 6 days ago (deploy now)	delete
Tree View Submodule Template (en-gb)	Deployed 6 days ago	delete
Tree View Submodule Template (fr-fr)	Last saved 6 days ago (deploy now)	delete
+ 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.

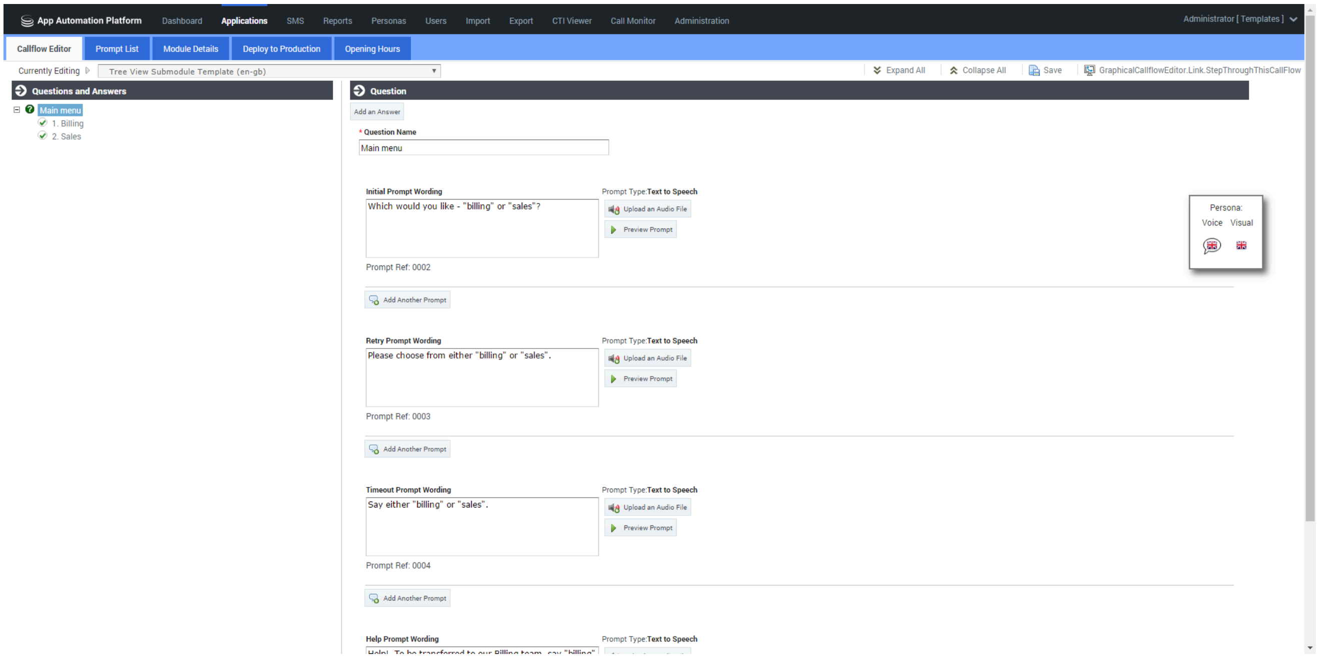
- **Menus** - These allow you to set up your own callflow using the Callflow Editor.

> ## 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.



Currently Editing: Tree View Submodule Template (en-gb)

- Main menu
  - 1. Billing
  - 2. Sales

* Question Name: Main menu

Initial Prompt Wording: Which would you like - "billing" or "sales"?
Prompt Type: **Text to Speech**
Prompt Ref: 0002

Retry Prompt Wording: Please choose from either "billing" or "sales".
Prompt Type: **Text to Speech**
Prompt Ref: 0003

Timeout Prompt Wording: Say either "billing" or "sales".
Prompt Type: **Text to Speech**
Prompt Ref: 0004

Help Prompt Wording: Help! To be transferred to our Billing team, say "billing"
Prompt Type: **Text to Speech**

> ## 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.

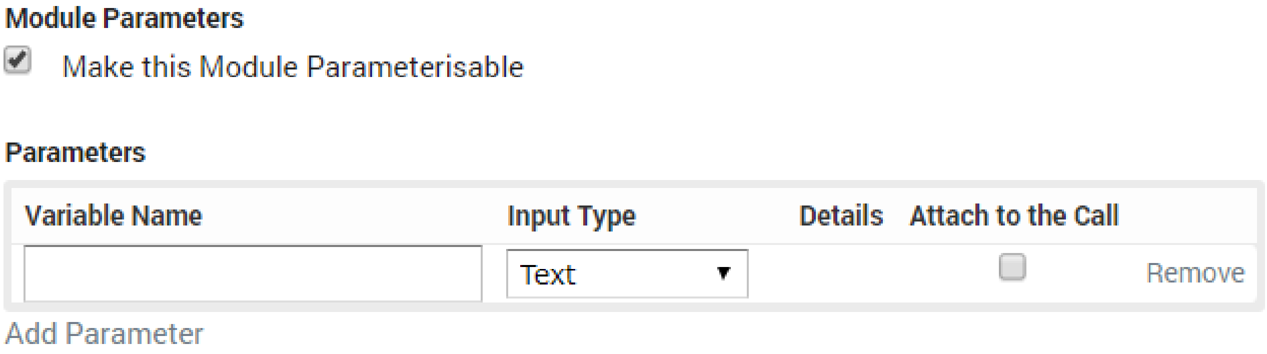
1. Click **Applications** in the navigation bar.
2. Click **Advanced Details**.
3. Click **delete** beside the application or module that you want to delete.

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:



Module Parameters

☑ Make this Module Parameterisable

Parameters

Variable Name	Input Type	Details	Attach to the Call	
	Text ▼		□	Remove

Add Parameter

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

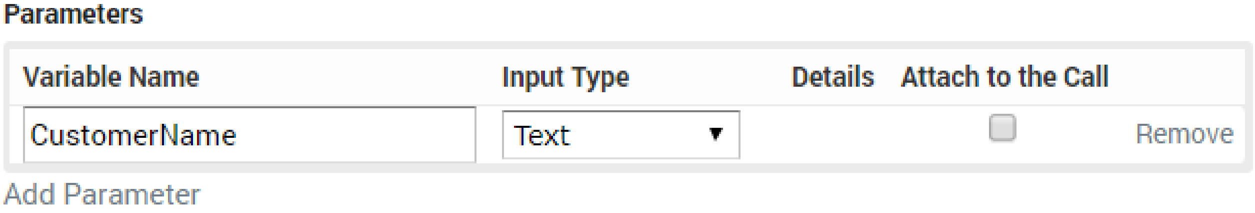
The following sections describe these variables, and how to declare them and specify from where the values are obtained.

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.



Parameters

Variable Name	Input Type	Details	Attach to the Call	
CustomerName	Text ▼		□	Remove

Add Parameter

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

Web Service

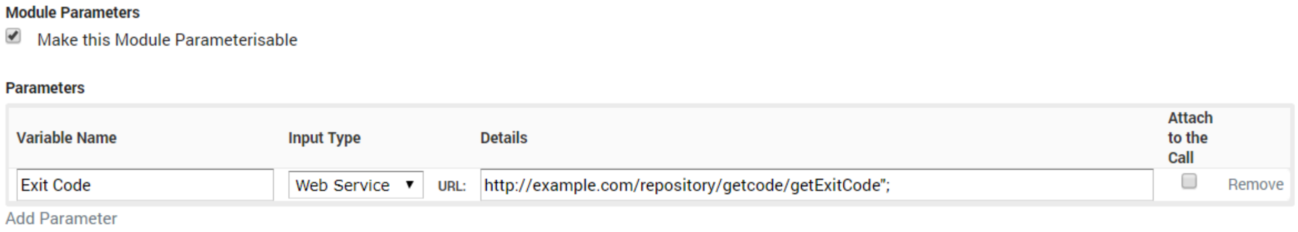
The value of a Web Service variable is derived from a set of values that are returned by a Web service. For example, declare a variable named `exitCode`, and specify the URL of a Web service that returns the following XML:

```
<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.



Module Parameters

☑ Make this Module Parameterisable

Parameters

Variable Name	Input Type	Details	Attach to the Call	
Exit Code	Web Service ▼	URL: http://example.com/repository/getcode/getExitCode";	☐	Remove

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:

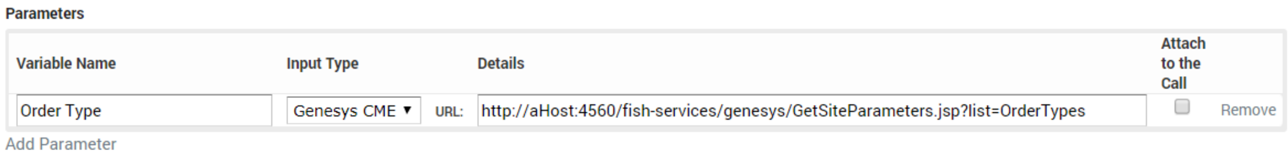
1. In the **Parameters List** section of the **Application Details** or **Module Details** tab, click **Add Parameter**.
2. Enter a **Variable Name**.
3. In the **Input Type** field, select **Genesys CME** from the drop-down menu.
4. In the **Details** field, enter the URL of the Web application from 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.



Parameters

Variable Name	Input Type		Details	Attach to the Call	
Order Type	Genesys CME ▾	URL:	http://aHost:4560/fish-services/genesys/GetSiteParameters.jsp?list=OrderTypes	□	Remove

Deleting Variables

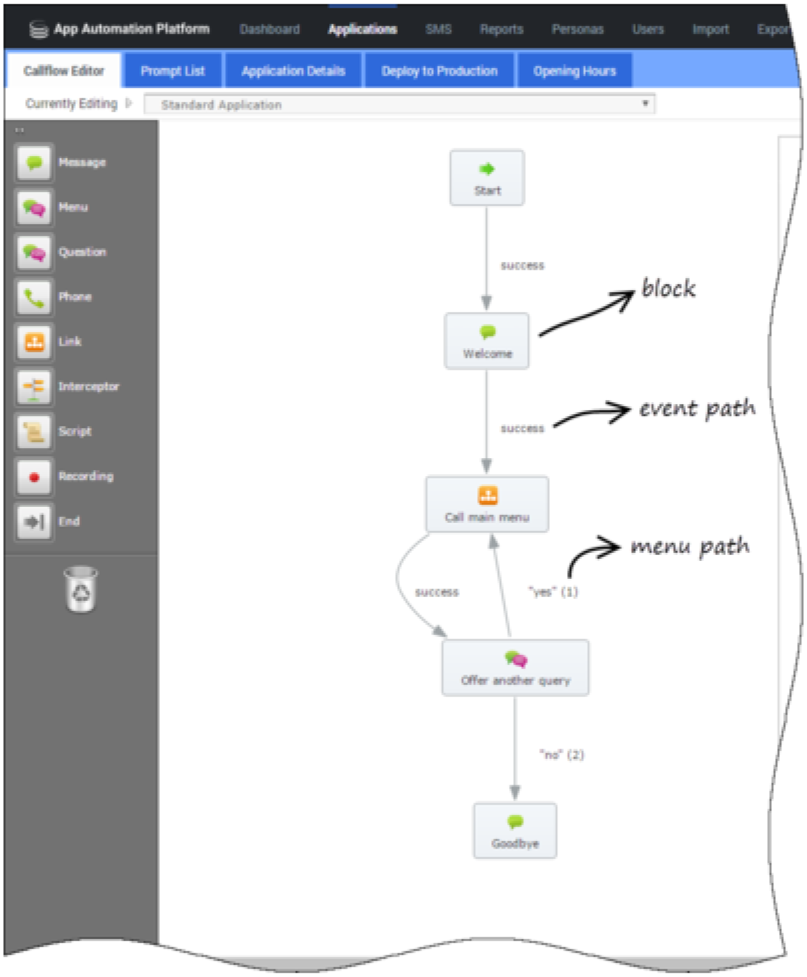
To delete a variable, select the variable in the list of parameters, and click **Remove**, located at the end of the row.

> ### 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.

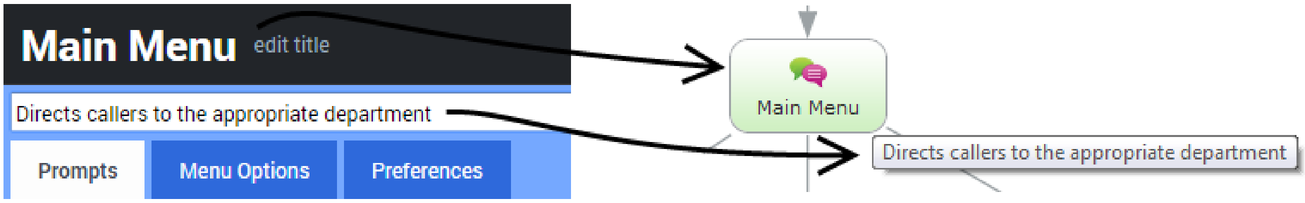
Using the Callflow Editor

Callflows consist of blocks and paths.

- Blocks are the individual steps in a callflow. Blocks perform a single basic function. For example, a Start block signals the start of the application, and an End block signals the end of the application.
- Paths link each block in the callflow. A path can represent:
  - A menu option that a caller must select to proceed to the next block (these paths are surrounded by quotation marks to indicate the option the caller must speak or press in order to progress to the next block. For example, **"yes (1)"** denotes a menu option in which the caller must say "yes" or enter 1 on a keypad.
  - An event that took place in order for the call to be routed in a certain direction (these do not have quotation marks). For example, **success** means this path is used if the previous block ended successfully, and **hangup** means this path is used if the caller hangs up.



You can click the name of a block or path to view its properties. For example, in the **Menu** block shown below, you can change its title or description.


Main Menu
edit title
Directs callers to the appropriate department
Prompts
Menu Options
Preferences
Main Menu
Directs callers to the appropriate department

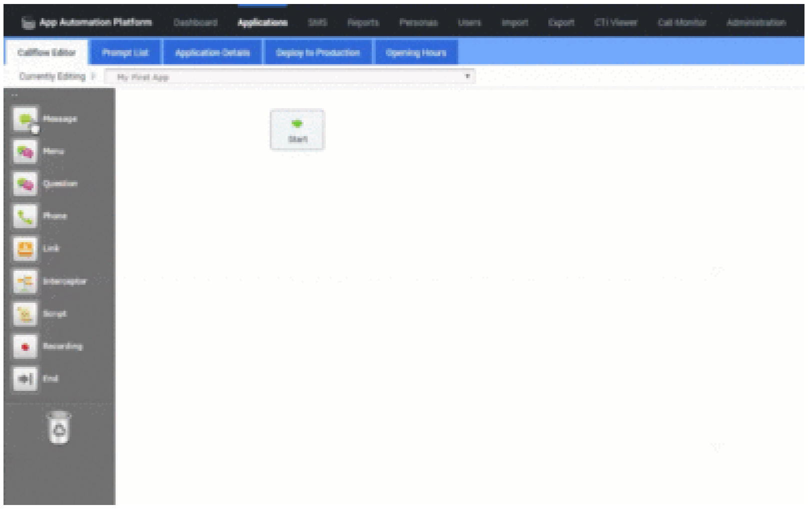
Tips for using the Callflow Editor

> **Important**
>
> This section does not apply to product-specific modules because you cannot alter the callflow of these modules.

Dragging and dropping blocks

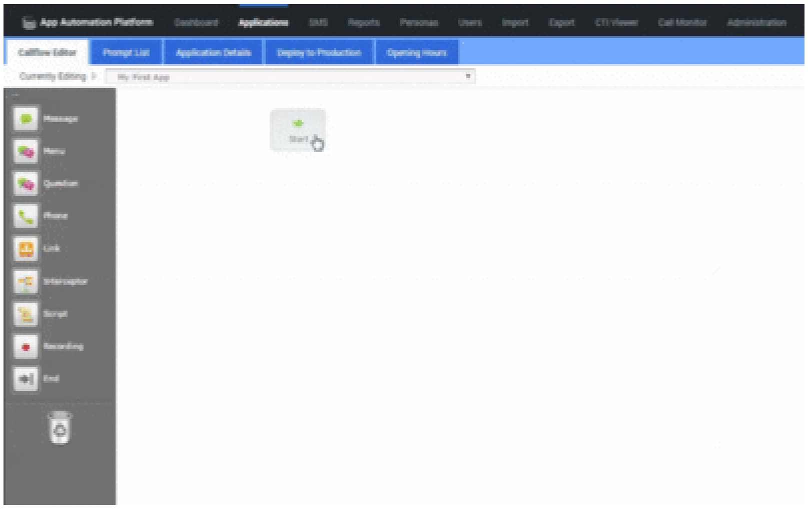
You can use several methods to add blocks to your callflow:

- Drag and drop a block from the toolbox onto a block on the callflow.
  Link to video

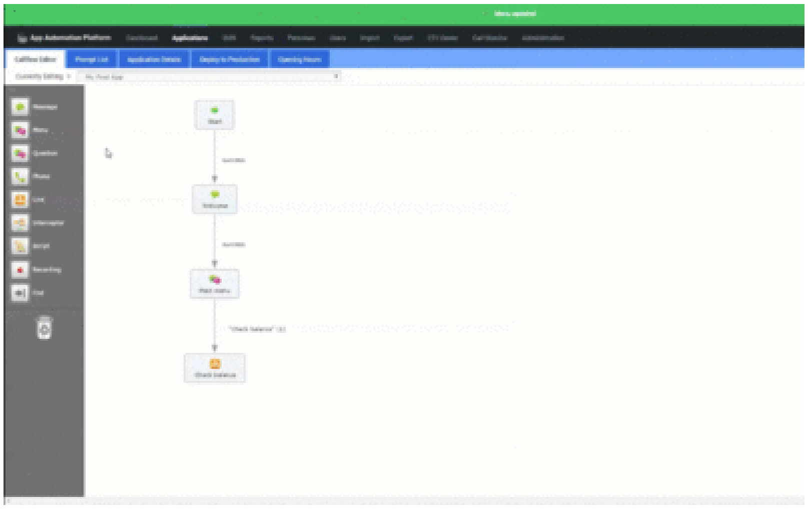


- Drag and drop a block that already exists on the callflow onto a block on the toolbox.

Link to video

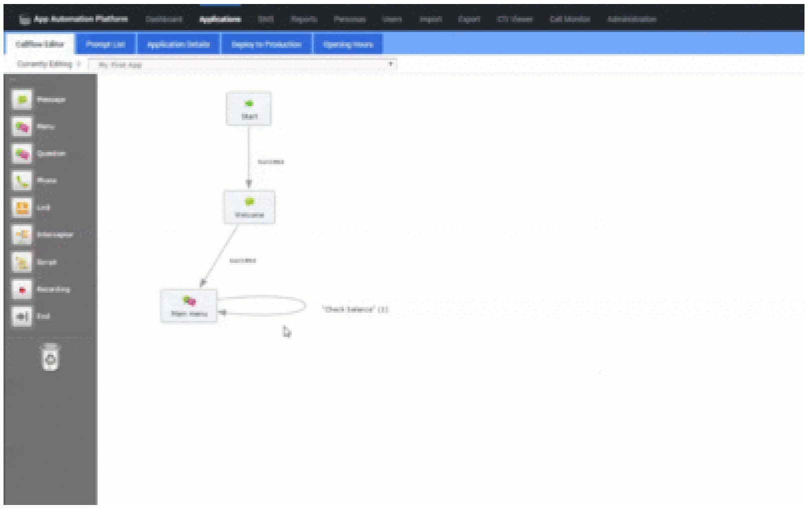


- Drag and drop blocks from the toolbox directly onto path names on the callflow.
  Link to video



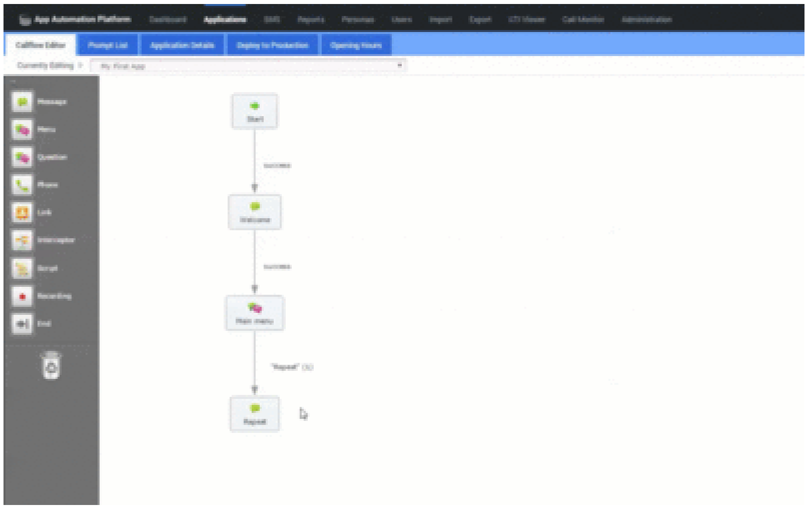
- Drag and drop path names onto blocks in the toolbox.

Link to video



- Drag and drop existing blocks in the callflow onto other blocks in the callflow.

Link to video



Removing blocks and paths

To remove a block from the callflow, drag and drop it onto the recycle bin icon in the toolbox. The following outcomes are possible:

- If the block was the result of a caller input (the path name was surrounded by speech marks, such as **"accounts"**), the path is not deleted with the block. Instead, the path links back to the block from which it originates. You must then decide whether you want to link this path to another block, or remove it by dragging the path name to the recycle bin.
- If the deleted block was the result of an event path, such as **success**, the path is also deleted with the block.

To remove a path from the callflow, drag and drop it onto the recycle bin icon 🗑 in the toolbox. The following outcomes are possible:

- If you delete an event path, the block into which the path leads is also deleted.
- If you delete a caller input path, the block into which the path leads is not deleted. Instead, the block is isolated and moved to the top of the callflow.

Miscellaneous tips

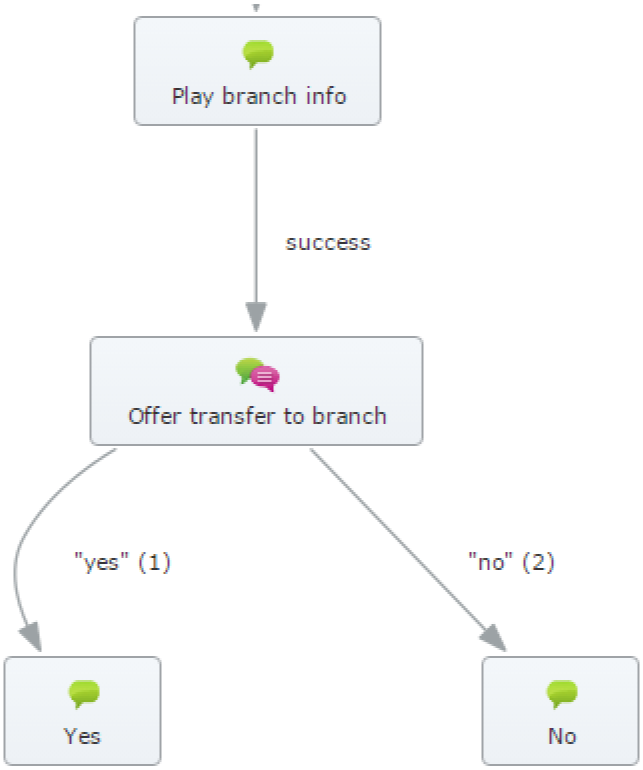
- Clicking outside a dialog box is the same as clicking **Cancel**. For example, if you clicked a Message block in the callflow to edit its prompt, but then decide against the change, you can click outside the dialog box to close it and cancel the changes.
- If required, the Default Behavior callflow can be linked to the application callflow.
  Link to video

Configuring callflow paths

Paths are shown as lines that connect blocks in the **Callflow Editor**.

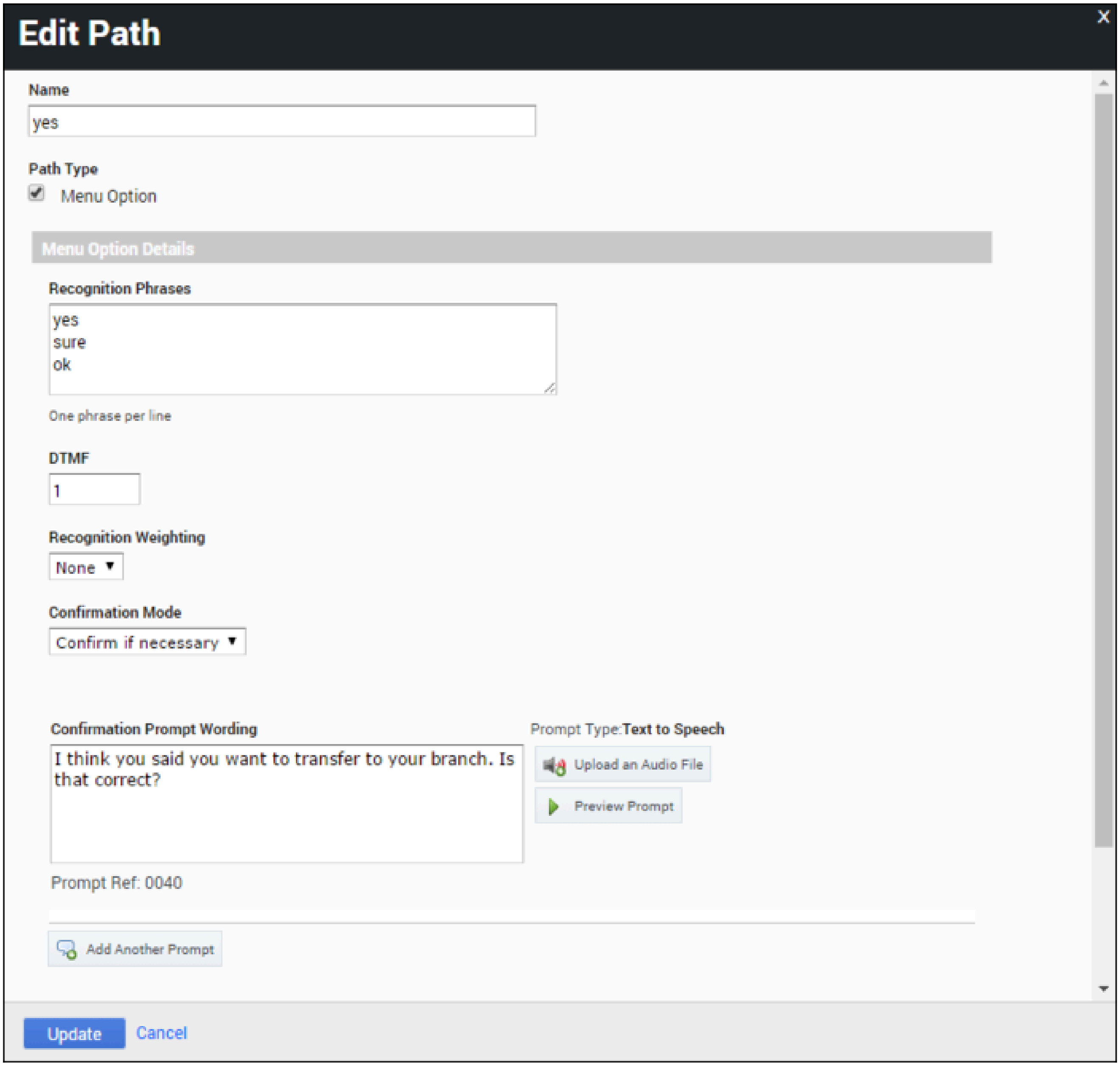
You can edit path names depending on the needs of the callflow. See Notes on names for more information on block/path names. For non-event paths, such as menu paths, you can also edit other options.

Consider this callflow:



You can do the following to configure the **"yes"** menu option path:

1. Click the **"yes"** path name in the callflow to view its properties.
2. Depending on whether you asked the caller to respond by voice or DTMF, or both, you must complete one or both of the following:
    - **Recognition Phrases** – Add phrases that the caller might say in answer to the menu question. Add a carriage return after each phrase.
    - **DTMF** – Enter the digit that the caller must press to access this menu option.
3. Click **Update**.



Edit Path

Name: yes

Path Type

☑ Menu Option

Menu Option Details

Recognition Phrases:

yes
sure
ok

One phrase per line

DTMF: 1

Recognition Weighting: None

Confirmation Mode: Confirm if necessary

Confirmation Prompt Wording: I think you said you want to transfer to your branch. Is that correct?

Prompt Type: Text to Speech

Prompt Ref: 0040

You can also apply the following settings to a menu option path:

Setting	Description
Recognition Weighting	Select the weighting to apply to the phrases you entered in the **Recognition Phrases** textbox. For example, consider you are configuring the callflow for a hospital that specializes in *neurology*, but it also has a minor *nephrology* department. Two possible paths from the preceding Menu block are **"neurology"** and **"nephrology"**. If you want to assign a higher weighting to the **"neurology"**

Setting	Description
	path, because that's what most callers will choose, then assign a higher weight rating to the **"neurology"** path and a lower weight rating to the **"nephrology"** path.
Confirmation Mode	Specify whether you want the system to confirm with the caller that the phrase or option collected is the correct one. Default confirmation prompts are inherited from the main application, but you can set the following options to override the inherited prompts: • **Never confirm** - Callers never have to confirm their choice. • **Always confirm** - Callers must always confirm their choice, regardless of the confidence score for speech recognition. • **Confirm if necessary** - Callers only confirm their choice if their response falls below the high confidence threshold, but above the low threshold, for voice recognition. Important You must set a confirmation prompt for **Always confirm** and **Confirm if necessary**. All the other confirmation prompts, such as confirmation retry prompt, are inherited from the main application.

Configuring global commands and paths

A global command refers to the commands often given by a caller during a call, such as:

- "back".
- "help".
- "repeat".

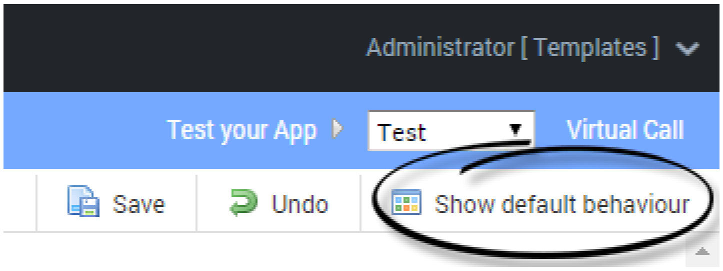
You enable global commands from your main application, using the **Enable standard menu options** option in Preferences.

The synonyms and weights for the commands are controlled by the following callflow preference settings:

- Standard ‘back’/‘help’/‘repeat’ menu option DTMF.
- Standard ‘back’/‘help’/‘repeat’ menu option synonyms.
- Standard ‘back’/‘help’/‘repeat’ menu option weight.

Configuring default behavior

Each callflow has a special type of block called **Defaults**. To view this block, click **Show default behaviour** at the top right-hand-side of the page.



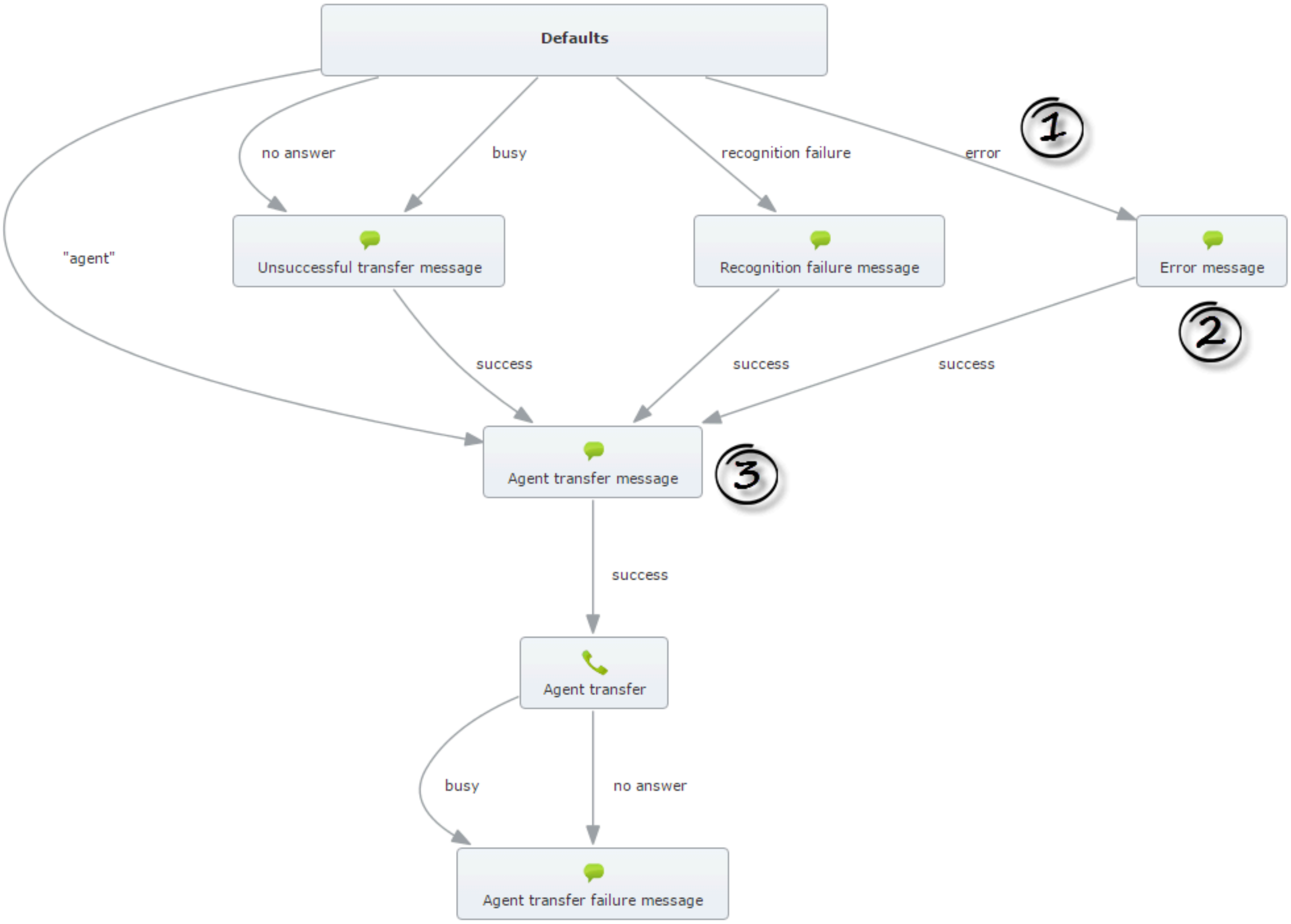
You can attach paths from this block to your main application. These paths act as global behaviors throughout the application callflow and any submodules called from the main application. For example, you can configure what happens when the caller says "agent" during any part of a call.

In your main application, you typically include default paths for the following events:

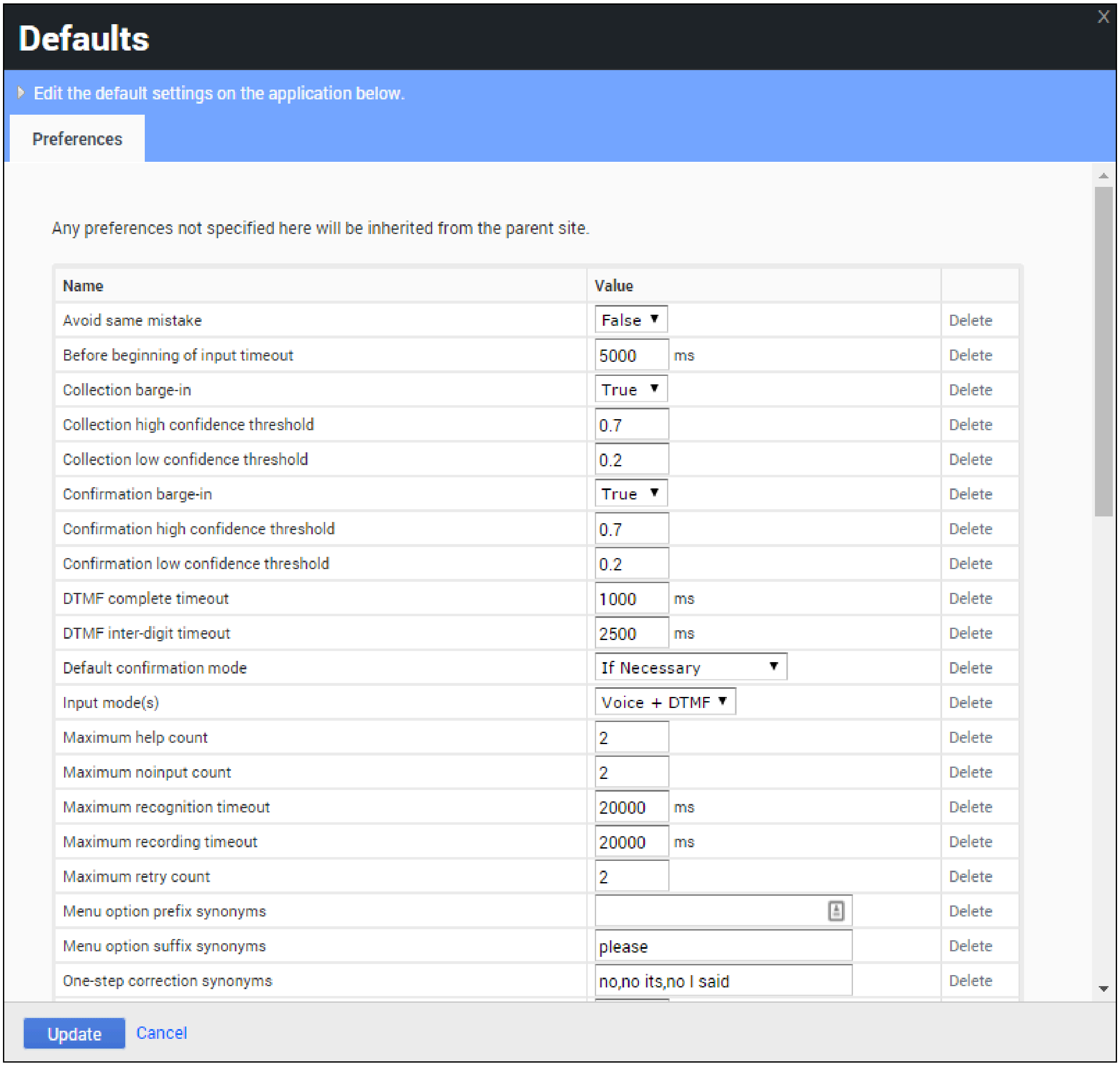
- **error**.
- **recognition failure**.
- **busy**.
- **no answer**.
- **agent**.

For example, the scenario below shows what happens if the caller encounters an error during the call:

1. The **error** path executes.
2. The *Error message* block plays a message, such as "Sorry, there is a technical problem. You are being transferred to an agent."
3. The call transfers to an agent.



You can also change default callflow preferences by clicking the **Defaults** block.



Defaults

Edit the default settings on the application below.

Any preferences not specified here will be inherited from the parent site.

Name	Value	
Avoid same mistake	False	Delete
Before beginning of input timeout	5000 ms	Delete
Collection barge-in	True	Delete
Collection high confidence threshold	0.7	Delete
Collection low confidence threshold	0.2	Delete
Confirmation barge-in	True	Delete
Confirmation high confidence threshold	0.7	Delete
Confirmation low confidence threshold	0.2	Delete
DTMF complete timeout	1000 ms	Delete
DTMF inter-digit timeout	2500 ms	Delete
Default confirmation mode	If Necessary	Delete
Input mode(s)	Voice + DTMF	Delete
Maximum help count	2	Delete
Maximum noinput count	2	Delete
Maximum recognition timeout	20000 ms	Delete
Maximum recording timeout	20000 ms	Delete
Maximum retry count	2	Delete
Menu option prefix synonyms		Delete
Menu option suffix synonyms	please	Delete
One-step correction synonyms	no,no its,no I said	Delete

Setting Callflow Preferences

The **Preferences** tab in each block dialog box allows you to apply specific rules as to how a call is handled.

A block inherits settings from the application default preferences, or from the module to which it belongs (if this is not an application module). However, specific preferences set within a block take precedence over the default preferences.

> **Important**
>
> An asterisk appears beside the **Preferences** tab label to indicate that preferences have been set up for that block.

The following table explains how to use the basic preferences:

Name	Description	Example/Notes	Typical Value(s)
ASR Language	Selects the language for the speech recognizer to use to identify speech.	Select **en-gb** for British English.	
Before beginning of speech timeout	Specifies the time, in milliseconds, after which the system times out if no input is received from the caller. The timer starts counting after the prompt has finished playing. This triggers the standard apology timeout prompt, followed by a timeout prompt.	The number of times the timeout prompt plays to the caller is determined by the **Maximum no input count** parameter. If this limit is reached, Genesys Intelligent Automation transfers the caller somewhere else.	The default value is `5000` milliseconds.
Collection barge-in	Specifies whether callers can interrupt an announcement that is being played to them in order to collect information.	For example, you might set up the initial prompt in a callflow to ask the caller to select a department. If you enable this option, the caller can make the selection before the prompt finishes.	This option is often enabled for IVR applications, especially when long announcements are used and it is not necessary for a caller to hear all of the options.
Confirmation barge-in	Specifies whether callers can interrupt a confirmation announcement.	A confirmation announcement asks callers to confirm information gathered by the system is correct.	Enable this option for efficiency and speed of use. However, consider areas within the application in which this

Name	Description	Example/Notes	Typical Value(s)
		For example, you might ask the caller to specify his or her full address, and Intelligent Automation plays back this information to confirm it was recognized correctly.	must be disabled (such as confirmation of a PIN number).
Collection high confidence threshold & Collection low confidence threshold	Specifies the upper and lower threshold, from `0.0` to `1.0`, to evaluate the quality of the caller’s input during speech recognition. The speech-recognition engine generates this confidence score as an indicator of how closely the caller's utterance matches the phrases specified in the grammar.	The speech-recognition engine awards a high confidence score when the application receives non-ambiguous input. In this case, the application usually accepts the response and continues. However, if the caller's utterance is assigned a confidence score between the high and low thresholds set, Intelligent Automation can ask for confirmation ("Is this correct?"). Important You can set a menu option to **Always confirm** to confirm inputs regardless of the value set in the **Collection high confidence** threshold. If you use the **Confirm if necessary** confirmation mode (default setting), the following rules apply: • If the confidence level is above the high threshold, the system auto-accepts the response. • If the confidence level is between the high and low thresholds, the system asks the caller to confirm the response. • If the confidence level is below the low threshold, the	Set these values depending on the type of information collected. For example, if the application asks the caller for feedback on the IVR experience, you can set a low value for the **Collection high confidence** threshold as it's not imperative that this information is verified. However, if the application asks for a credit card number, a high threshold can be used to ensure the information is correct.

Name	Description	Example/Notes	Typical Value(s)
		system rejects the response. Depending on your desired outcome, you can amend the threshold values as follows: • To confirm more responses, increase the high threshold. • To reject more responses, increase the low threshold. • To auto-accept more responses, decrease the high threshold. • To confirm more responses, decrease the low threshold. In summary, if you want to: • Confirm more responses, decrease the low threshold and increase the high threshold. • Reject more responses, increase both the low and high thresholds.	
DTMF complete timeout	Specifies the time, in milliseconds, after which the system times out after the caller has stopped entering information in response to a DTMF question.	You might ask the caller to enter a five-digit account number. If this option is set to `0`, the system proceeds to the next stage without delay after five digits have been collected.	You might set a low value to avoid a long pause after the last digit is entered.
DTMF inter-digit timeout	Specifies the amount of time, in milliseconds, the system waits between each DTMF input character before interpreting the DTMF string.	You might ask the caller to enter a 16-digit credit card number, but the caller only enters 15 digits by mistake. Intelligent Automation waits the specified number of milliseconds before timing out and prompting the caller to re-enter the number.	This depends on the complexity of the question the caller was asked. For example, if you asked the caller to "press 1 for accounts and 2 for sales", no inter-digit time delay is required. However, with requests for longer, more complex information, you can

Name	Description	Example/Notes	Typical Value(s)
			give the caller extra time to finish entering digits.
DTMF termination character	If specified, this is the DTMF character the caller must press after he or she has finished entering DTMF information. This termination character indicates the input is finished (for example #). You must inform callers to use this character in all prompts that require DTMF input. For example: "Please enter your 16-digit credit card number, followed by the # key." Important If the caller does not press the terminating character, Intelligent Automation waits for the value you specified for **DTMF complete timeout** before accepting the input. Therefore, if you are using a termination character, you must increase the **DTMF complete timeout** value.	If you want the caller to enter a five-digit account number, but you do not want a time delay after the caller has finished, you can ask the caller to enter the number, followed by the # key. This way, the system is sure the caller has finished entering data.	This depends on the nature of the data requested from the caller. If you are asking the caller to press one digit, a termination character might not be necessary. However, if you ask the caller to provide a string of digits, such as a credit card number, you can use a termination character.
Input mode(s)	Specifies which input modes to enable. • DTMF • Voice • DTMF and Voice.	You can use DTMF to collect information relating to significant numbers, such as credit card numbers and account IDs. You might also use speech recognition at times when DTMF is not feasible (for example, asking the customer for a full postal address).	Turn off voice recognition if you are expecting the caller to be calling from a noisy environment.
Maximum help count	Specifies the maximum number of times a caller can ask for help during a single **Menu** or **Question** block before they are rerouted.	You can route the caller to an agent if he or she asks for help several times during a call.	The default value is 2.
Maximum no input count	Specifies the maximum number of times a retry	If a caller is confused by a particular question, he	The default value is 2.

Name	Description	Example/Notes	Typical Value(s)
	prompt plays to a caller if no caller input is received. At this point the call transfers somewhere else.	or she might not provide any feedback. If this option is set, the call routes to another number, such as an agent.	
Maximum recognition timeout	Specifies the maximum amount of time the system wait for the speech-recognition engine to recognize an utterance after it has detected speech. If this time is exceeded, you can initiate a retry prompt.	For example, if the caller is talking in a noisy environment, he or she might stop talking but the speech-recognition engine might think the caller is still speaking. This setting allows you to allocate a set amount of time for the engine to analyze the input.	This value depends on the type of question you have asked the caller. If you anticipate a complicated answer, give the speech-recognition engine a longer time to interpret the response.
Maximum recording timeout	Specifies the maximum length of time to allow a caller to make a recording. This option is only relevant to Recording blocks.	}Intelligent Automation can detect when the caller is speaking (without speech recognition) and continues onto the next block when the caller stops speaking. However, if the background noise is too loud, Intelligent Automation might confuse this with speech. Therefore, this parameter sets the absolute maximum time the recording can last.	This value depends on the type of question you have asked the caller. If you have asked an open-ended question, you must leave enough time for the caller to give a complete answer. If you anticipate a short answer, set a quicker timeout. Be aware that if you set a higher value, the caller might stop speaking after 10 seconds but Intelligent Automation hears noise on the line and thinks the caller is still talking.
Maximum retry count	Specifies the maximum number of retries you want to allow a caller. A retry might be necessary if a digit entered was not recognized during a DTMF response, or if a low confidence is calculated by the speech-recognition engine during a spoken response.	You might ask the caller to enter an account number using the keypad. In this case, you might want to allow at least one retry in case digits are entered incorrectly. However, if you ask the caller to speak an account number, you might allow more than one retry in case speech recognition is more complicated.	This value depends on the difficulty of what you are asking the caller. The more difficult the question, the more retries you might allow. However, too many retry attempts might frustrate the caller. You can also ensure your retry prompt acknowledges the difficulty the caller might be experiencing.
Recognition complete timeout	Specifies the length of silence, in milliseconds, after which the system	If the caller provides a recognized response and then stops talking,	The longer the expected answer is, the larger this value should be.

Name	Description	Example/Notes	Typical Value(s)
	times out after the caller provides an answer that matches something in the grammar.	this value specifies how long to wait before proceeding. However, if the response is not recognized in the grammar, the **Recognition incomplete timeout** parameter determines how long the system waits before timing out.	This timeout setting can be set to the same value (or slightly shorter than) the **Recognition incomplete timeout** value. For example: • "yes/no" question – `500` • menus – `1000` • speaking credit card numbers – `3000` This could be followed up by a confirmation prompt, depending on the confidence score attained by the caller's answer.
Recognition incomplete timeout	Specifies the length of silence, in milliseconds, after which the system times out if the caller stops talking but has not yet mentioned any required words or phrases.	For example, if a caller pauses in the middle of providing an answer, Intelligent Automation cannot match the utterance against the grammar. If the caller does not continue, Intelligent Automation waits the specified number of milliseconds before timing out. This value gives the caller time to provide the final piece of the answer.	This value should scale according with the expected length of the utterance. This timeout is normally followed by a retry prompt.
Recording complete timeout	Specifies the amount of time to wait, in milliseconds, after a caller provides a recorded response to when Intelligent Automation accepts the recording and continues to the next block. This option is only relevant to **Recording** blocks.	If you asked the caller for feedback during a questionnaire, you might set an amount of time to wait before proceeding to the next block after he or she has finished answering. This wait gives the caller time to think of whether they want to add anything else.	This value depends on the nature of the question. If it is an open-ended question, you can set this value higher in case the caller decides to add more information. However, if it is a straight-forward answer with a one- or two-word response, you can set a lower value.
Recovery mode enabled	Specifies whether or not recovery mode is enabled. Recovery mode uses Recovery prompts to return the caller to the start of the current question, allowing them to	An example prompt might be: "I'm having problems understanding your response. Do you want to try again or speak to an agent?" A response of "Try again" returns the caller to the	This option is disabled by default.

Name	Description	Example/Notes	Typical Value(s)
	attempt the question again and avoid the failure path.	main menu.	
Rich Media Format	Specifies the output format for rich media message options.	Select **Text, Buttons and Videos** to use a rich media format that incorporates all of these elements. If **None** is selected, all rich media prompts are ignored. Important You can also set the following default server settings: • **Resources.Allowe** list of audio file mime types to allow users to upload. • **Resources.Allowe** list of image file mime types to allow users to upload. • **Resources.Allowe** list of video file mime types to allow users to upload.	
Rich Media try fallback if necessary	{Specifies whether to try a fallback format if the selected **Rich Media Format** is incorrectly configured (for example, if **Text, Buttons and Videos** is selected but a video is not attached). In other words, this setting determines whether Intelligent Automation validates the user's configuration or simply attempts to use what it is given.	Select **True** if you want Intelligent Automation to validate the user's rich media format. If the format is incorrect, Intelligent Automation tries to select an alternate format.	False
Sensitivity	Sets the speech-recognition engine's sensitivity to noise during input recognition. The value entered must	You can set the sensitivity level above `0.5` if you expect the majority of your callers to be in a quiet	The default value is `0.5`. If you adjust this value, do so using small increments or decrements, one day or

Name	Description	Example/Notes	Typical Value(s)
	be between `0.0` (least sensitive to noise) and `1.0` (highly sensitive to quiet input). Thus, if you set the property to a low value, the recognizer is less sensitive to noise, but the user must speak more loudly in order to be recognized.	environment, such as their own home. However, set this value below `0.5` if you expect callers to be in a noisy environment, such as a busy workplace.	week at a time, to ensure the adjustment does not have a detrimental effect on your customers.
Transfer timeout	Specifies the time, in seconds, to wait before a call exits with a result of **no answer**. If several phone numbers are provisioned, this is the maximum time to wait for each one before timing out.	If you ask the caller to select a department, and no one in the department answers and no voicemail is available, the Intelligent Automation waits for the timeout before routing the caller to the receptionist to allow them to leave a message.	Choose a value that allows people a reasonable amount of time to answer the call – but not too long that the caller loses patience.

Important

- Contact Genesys for information on advanced preferences not mentioned in the table above.
- If a call exceeds maximum values specified in these parameters (apart from **Maximum recording timeout**), the block exits with a result of **recognition failure**. This usually results in the call being routed to an agent, but this behavior can be overridden.
- Getting your timeout values right is key to making your application as pleasant to use as possible. Shorter timeouts mean a snappier response to the caller, but may also mean that the caller gets interrupted by the system before he or she finishes speaking.
- Intelligent Automation enables you to quickly make small, incremental changes to these parameters and deploy these changes to production. You can monitor the resulting statistics over time and, if no improvement is obvious (or the changes are detrimental), roll back the changes to the previous value/version. All this can be done without having to make software changes, do extensive release/testing cycles, or take the application offline.

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.

Prompts always give precedence to uploaded audio files over TTS text. However, Genesys Intelligent Automation retains the TTS text so you can easily switch back and forth between TTS and prerecorded audio, without having to retype the TTS or reupload the sound file.

> Tip
>
> If you use prerecorded audio files, Genesys recommends that you also enter equivalent TTS text so system users can easily read the prompts and quickly identify issues when troubleshooting.

Variables and playback of dynamic data

You can add variables to your prompts using square-bracket notation. Variables are containers for storing data, and they are replaced with real data when the application executes. For example, consider the following prompt: "Your current balance is [currency:PaymentBalance]." When the application executes, the variable **[currency:PaymentBalance]** is replaced with the caller's payment balance value.

Intelligent Automation uses variable data in several locations. Prebuilt modules can use them, such as in the *payment balance* example above, or they can be returned by web services, or set manually in Script blocks.

Every variable has a name and a formatter. The formatter indicates which sort of value the variable represents (for example: a digit, a date, or a currency).

Intelligent Automation sources variable data from several locations. Some prebuilt modules create variable data automatically, whereas others are returned by web services and can be set manually in Script blocks.

The following variable formatters are available:

Variable formatter	Description
[var:myvariable]	Plays the value of *myvariable* as regular TTS.
[digits:myvariable]	Plays the value of *myvariable* as a series of digits.
[letters:myvariable]	Plays the value of *myvariable* as a series of letters.

Variable formatter	Description
• [currency:myvariable]	Plays the value of *myvariable* as units of money. The variable value must be formatted accordingly, such as **GBP12.44** or **EUR3.90**, which is read out as "twelve pounds and forty four pence" or "three euro and ninety cents", respectively. You can also also provide negative amounts, such as **GBP-12.44**, which is read out as "twelve pounds and forty four pence in credit." The "in credit" portion refers to a standard prompt defined at the application level.
[date:myvariable]	Plays the value of *myvariable* as a date. Values must be formatted as *yyyy-mm-dd*. For example, **2010-01-31** is read out as "the thirty-first of January, twenty-ten." You can omit part or all of the year (**??10-01-31**) to read out only a portion of the value. For example, to only say "the thirty-first of January", use **????-01-31**.
[time:myvariable]	Plays the value of *myvariable* as time. Values must be formatted as *hh-mm*. For example, a value of **13:59** is read out as "one fifty nine p m."
[ccexp:myvariable]	Plays the value of *myvariable* in a manner consistent with credit card expiry dates. The value must be formatted in one of the following ways: • mm/yy • mm/yyyy • mmm/yy • mmm/yyyy Important The format *mmm* refers to a three-letter month name, such as **jan** for January.
[pause]	Inserts a pause. You can control the pause length by including a duration in milliseconds, such as **[pause:500]**. The default pause duration is **750**.
[prompt:myvariable]	Plays the value of *myvariable* as an audio prompt. The value can be a string that points to the name of a prompt in the **Prompt List** tab, such as **0034** or **Standard timeout apology prompt**. If the value does not match an existing prompt reference, it creates a temporary one and uses the variable value as both the name of a WAV file and as the fallback TTS.
[prompt:myvariable;fallback tts]	Plays the value of *myvariable* as an audio prompt. The value can be a string that points to the name of a prompt in the **Prompt List** tab, such as **0034** or **Standard timeout apology prompt**. If the value does not match an existing prompt reference, it creates a temporary one and uses the variable value as both the name of a WAV file and

Variable formatter	Description
	as the fallback TTS. Alternatively, you can add a semicolon and some fallback TTS text to be used in case the WAV file does not exist. Important If no variable called *myvariable* exists, it uses the actual string **myvariable**. For example, assume a variable exists called **Manufacturer** with the value **Acme**. If you use [prompt:Manufacturer], Intelligent Automation tries to use an audio file in the company's resources folder called **Acme.wav**. If the audio does not exist, it plays the word "Acme" using TTS. Alternatively, you can use [prompt:Manufacturer;an American manufacturer] in the same way, but if the file does not exist then Intelligent Automation uses TTS to say "an American manufacturer".

Uploading dynamic prompt recordings

You can upload a set of audio files (in a single ZIP file) that play back dynamic information such as digit strings, currency amounts, dates, and more. These prompts are shared across a company's applications. The server attempts to use these recordings for prompts, falling back to TTS if necessary.

See the Dynamic Prompt Uploads section on the Personas page for more information.

Prompt List

Intelligent Automation enables you to easily make and test changes to audio files by calling the test number. You can also generate a Prompt List that you can send to the studio for recording. This list is generated based on the **Prompt Ref** labels seen under each prompt. If the studio labels its recordings based on the **Prompt Ref** labels, you can upload the audio files to the system and Intelligent Automation automatically updates the prompts according to the **Prompt Ref** labels.

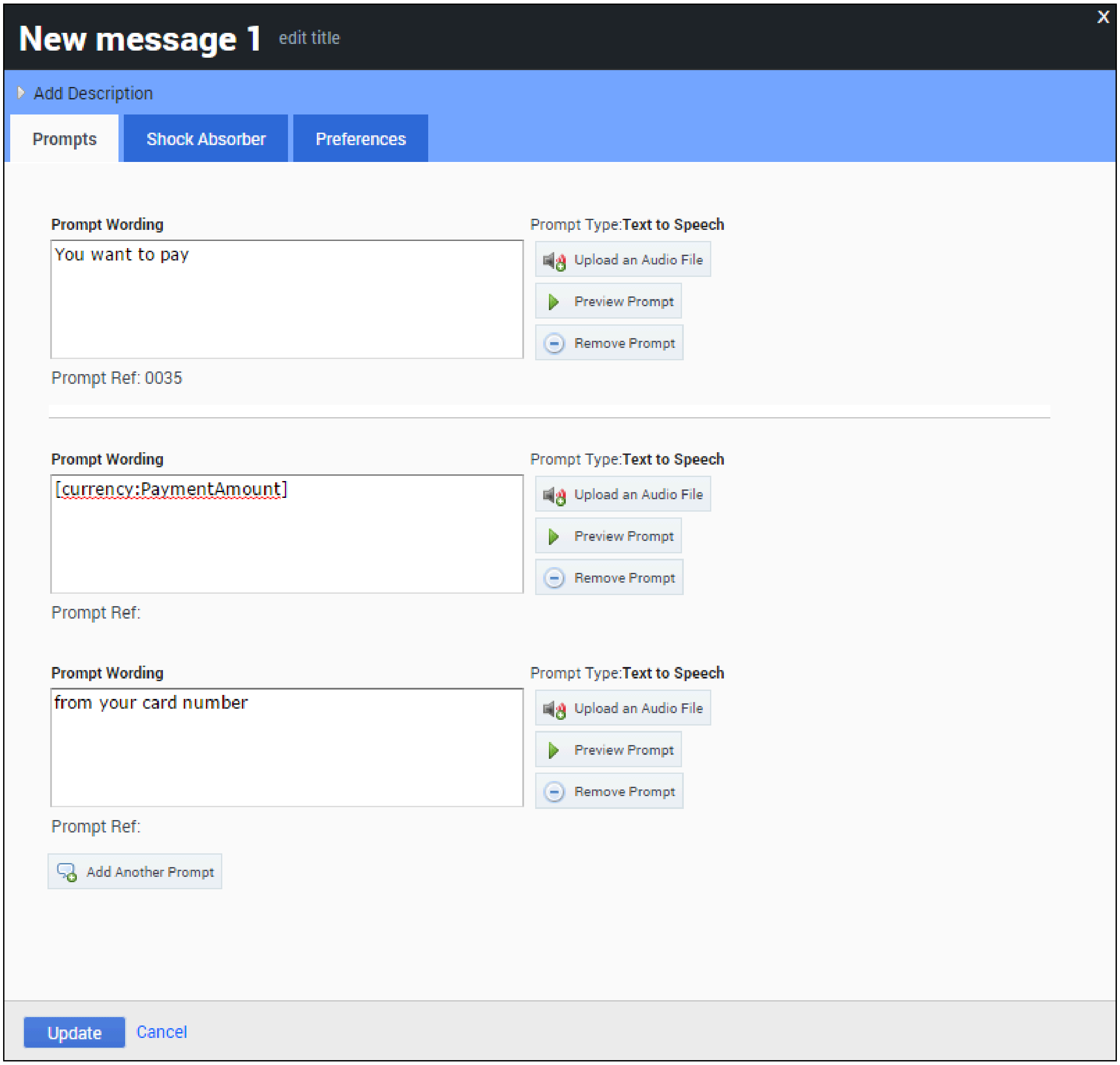
To generate a Prompt List, go to the **Prompt List** tab and click **Manage Prompts** in the top-right corner.

Important

Standard prompts for transfers, confirmations, and more are automatically set up in the main application. These are used throughout any submodules linked to the main application. You can change these prompts using the **Prompt List** tab.

Tips for writing prompts

- You can add prompts in chunks, as shown below, to play back dynamic information (such as an address) in the middle of other prompts. Or, you can upload an audio file for the first section of the prompt and use TTS for the second.



Prompt Wording

You want to pay

Prompt Type:**Text to Speech**

Prompt Ref: 0035

Prompt Wording

[currency:PaymentAmount]

Prompt Type:**Text to Speech**

Prompt Ref:

Prompt Wording

from your card number

Prompt Type:**Text to Speech**

Prompt Ref:

- Always click **Preview Prompt** after you enter a TTS prompt to confirm the speech engine reads the prompt correctly. If the speech engine has difficulty pronouncing a word, try spelling the word phonetically.
- Place your menu options in logical order. Group similar options together and place the most common ones near the top.
- When providing menu options, keep your prompt list as short as possible. Generally, three to five items

are sufficient. If necessary you can use a hierarchy, but keep it to two or three levels. Otherwise, callers might become confused, unengaged, and frustrated.

- Put the DTMF option after the name of the person or department.
  Incorrect
  `Press 1 for Sales`

  Correct
  `For Sales, Press 1`

  Callers listen for their destination first, then how to get there. If you play the DTMF option first, callers are not as likely to associate the option with the department.
- Use consistent phrasing for menu options.
  Incorrect
  `For Sales press 1. To reach the Service department, press 2.`

  Correct
  `For Sales, press 1. For Service, press 2.`
- Use consistent ordering of nouns and verbs to avoid confusing the caller.
  Incorrect
  `Which would you like: Sales, Faults, or Check an Order?`

  Correct
  `Which would you like: Sales, Faults, or Order Updates?` (nouns only)

  Correct
  `What do you want to do: buy something, report a fault, or check an order?` (verb then noun)

Configuring standard prompts

You can view a list of standard prompts in the **Prompt List** tab, under the **Other Prompts** section. Intelligent Automation uses these default apology and confirmation prompts throughout your application. You can update these prompts as required by your business needs.

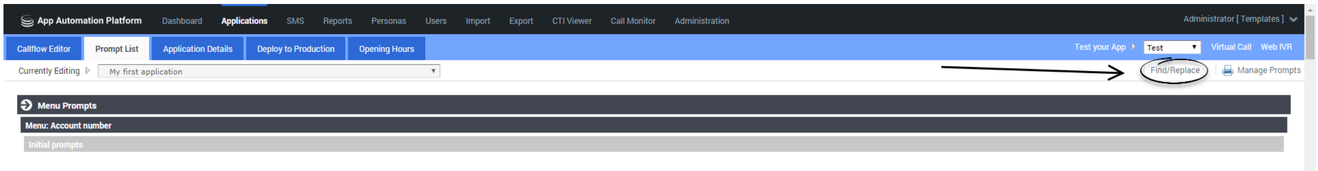
You can define the following standard prompts in your application and also have these prompts used in a sub-module.

Prompt Name	Example Wording	Further Information
Apology Prompts		
Standard apology nomatch prompt	Sorry, I didn't get that.	Played if the caller input is not recognized as part of the active grammar during collection. This is followed by a retry prompt.
Standard apology timeout prompt	Sorry, I didn't hear you.	Played if the system receives no input from the caller. The timeout value is configured using the **Before beginning of speech timeout parameter** in Preferences. This is followed by a retry prompt.

Prompt Name	Example Wording	Further Information
Standard apology wrong confirm prompt	My mistake.	Played if the caller specifies that the confirmation prompt played by the system is incorrect. This is followed by a retry prompt.
Confirmation Prompts		
Standard confirmation help prompt	Help! I need to know if I got it correctly. Please say yes or no.	Played to the caller if they invoke the global "help" command while being asked to confirm an answer.
Standard confirmation initial prompt 1	I think you said	Confirmation prompts are used to confirm that the system has correctly recognized the answer provided by the caller. The value as recognized by the system will be played between prompt 1 and prompt 2.
Standard confirmation initial prompt 2	, is that correct?	
Standard confirmation retry prompt	Sorry, please say yes or no.	Played to the caller to confirm an option if the application thought the caller said something other than "yes" or "no" (or an active default menu option, such as "agent") during a confirmation step.
Standard confirmation timeout prompt	Sorry, I didn't hear you. Please say yes or no.	Played to the caller if the system times out because the caller has either not provided an answer, or the input provided does not match anything in the grammar.
Currency Prompts		
Standard negative currency prefix prompt		Played when playing back a negative currency (for example, GBP-12.45, or "prefix twelve pounds and forty five pence suffix", which translates as "twelve pounds and forty five pence in credit."
Recovery Prompts		
Standard recovery prompt	I'm having trouble understanding you. If you want to try answering the question again press 1. Or, to speak with an advisor press 2.	This prompt is used to return the caller to the main menu and allow them to attempt the call again, therefore avoiding the failure path. Important Contact your Genesys representative if you want to use Recovery Mode but your callflow does not have a standard recovery prompt.

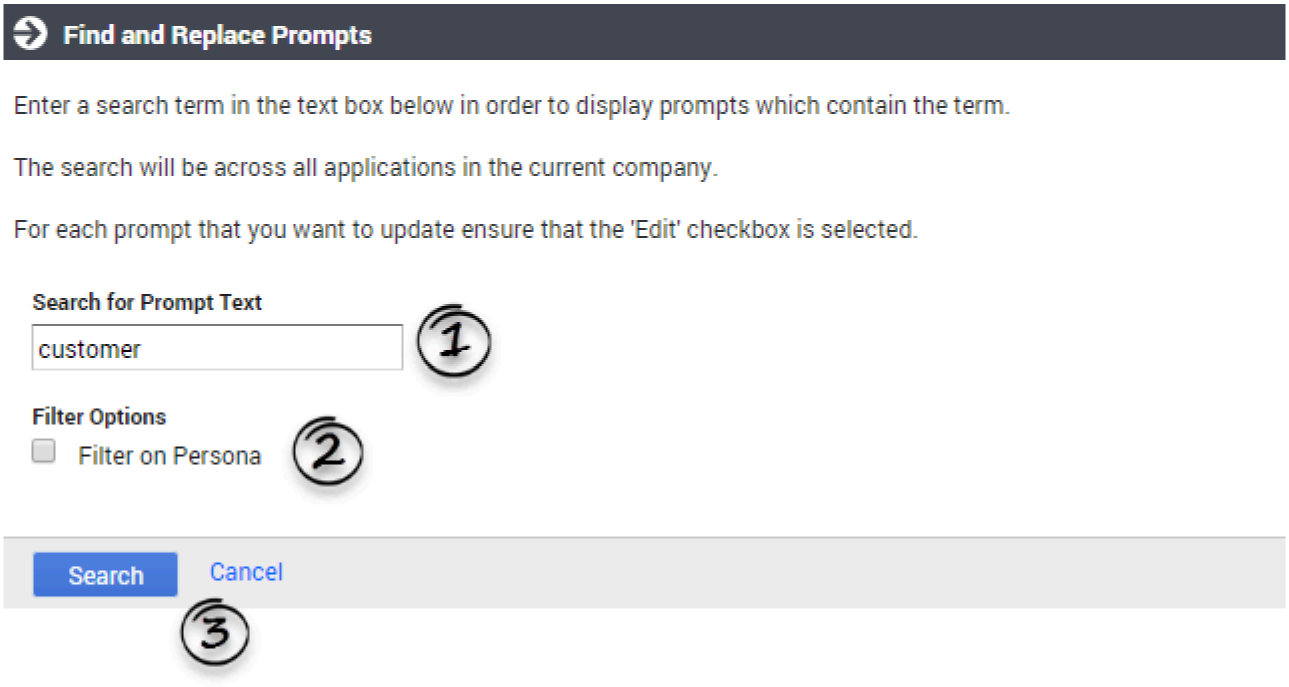
Using the Find/Replace feature

Click **Find/Replace** to search for a particular word or term in your prompts.



In the pop-up window:

1. Enter a search term in the **Search for Prompt Text** field.
2. Optionally, click **Filter on Persona** to only search within a specific persona.
3. Click **Search**.



Find and Replace Prompts

Enter a search term in the text box below in order to display prompts which contain the term.

The search will be across all applications in the current company.

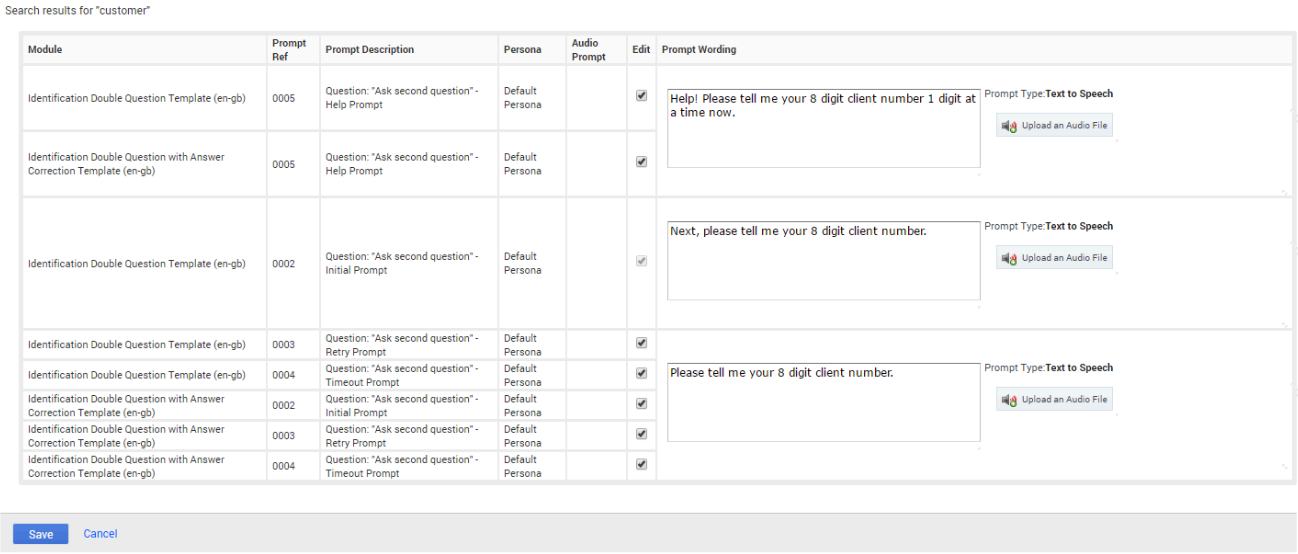
For each prompt that you want to update ensure that the 'Edit' checkbox is selected.

Search for Prompt Text: customer

Filter Options

□ Filter on Persona

Intelligent Automation displays prompts that match your search term. To edit a particular prompt, select its **Edit** check box. Next, edit the TTS text and upload a new audio file (if necessary). Click **Save** when done.



Search results for "customer"

Module	Prompt Ref	Prompt Description	Persona	Audio Prompt	Edit	Prompt Wording
Identification Double Question Template (en-gb)	0005	Question: "Ask second question" - Help Prompt	Default Persona		☑	Help! Please tell me your 8 digit client number 1 digit at a time now. Prompt Type:**Text to Speech** Upload an Audio File
Identification Double Question with Answer Correction Template (en-gb)	0005	Question: "Ask second question" - Help Prompt	Default Persona		☑	
Identification Double Question Template (en-gb)	0002	Question: "Ask second question" - Initial Prompt	Default Persona		☑	Next, please tell me your 8 digit client number. Prompt Type:**Text to Speech** Upload an Audio File
Identification Double Question Template (en-gb)	0003	Question: "Ask second question" - Retry Prompt	Default Persona		☑	Please tell me your 8 digit client number. Prompt Type:**Text to Speech** Upload an Audio File
Identification Double Question Template (en-gb)	0004	Question: "Ask second question" - Timeout Prompt	Default Persona		☑	
Identification Double Question with Answer Correction Template (en-gb)	0002	Question: "Ask second question" - Initial Prompt	Default Persona		☑	
Identification Double Question with Answer Correction Template (en-gb)	0003	Question: "Ask second question" - Retry Prompt	Default Persona		☑	
Identification Double Question with Answer Correction Template (en-gb)	0004	Question: "Ask second question" - Timeout Prompt	Default Persona		☑	

Blocks

Blocks are the individual steps in a callflow.

Each blocks performs a single basic function. For example, a Start block signals the start of the application, and an End block signals the end of the application.

The following blocks are available in Genesys Intelligent Automation:

- Start Block
- End Block
- Message Block
- Menu Block
- Question Block
- Phone Block
- Link Block
- Interceptor Block
- Script Block
- Recording Block

Notes for blocks

When creating blocks, ensure that the name for the block meets the following conditions:

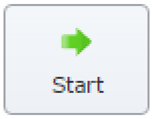
- The name must not include ASCII control characters (code 0 to 31) or characters with a code greater than 126.
- The name must also not include these symbols:

ASCII Code	Symbol
34	"
40	(
41	)
58	:
92	\

Start Block

Important

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



A **Start** block indicates the start point of an application or module. Therefore, all applications and modules have a **Start** block. This block automatically appears in the callflow of any new application or module.

A secondary purpose of a **Start** block is to read attached data fields from a CTI-enabled system. You can specify the names of the fields you want to read, and the values are stored as variables that can be accessed in your callflow using **Script** blocks.

Start blocks cannot be removed from a callflow.

End Block

The **End** block ends the call.

Sometimes there are circumstances in which a call ends naturally, such as if the caller reaches the end of the callflow in an application and there are no paths from the last block. In this case, an **End** block might seem unnecessary. However, if you want the call to end at a given point, such as in a submodule, you must add an **End** block to the callfow to ensure the caller is not looped back to the main application.

To add an **End** block to a callflow, drag and drop the **End** block onto the end of the callflow.

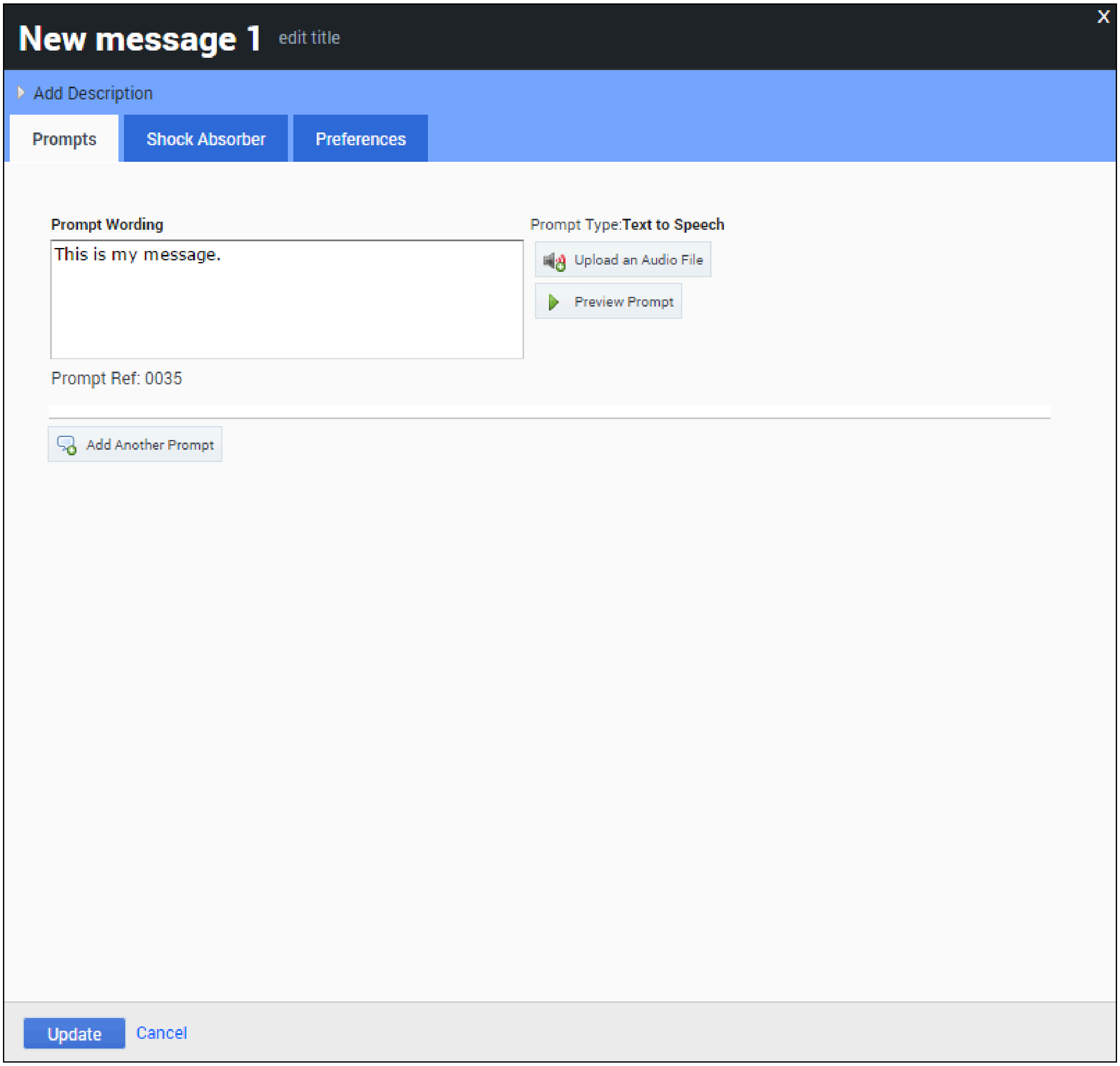
Message Block

You use **Message** blocks to play a message to the caller. These messages consist of one or more prompts, using either prerecorded audio or Text To Speech (TTS).

Adding this block to the callflow

To add and configure **Message** blocks to a callflow:

1. Drag and drop the **Message** block onto the appropriate position in the callflow.
2. Double-click the **Message** block to open its properties.
3. In the **Prompts** tab, add a message to be played to callers, either by:
    - Entering TTS text in the **Prompt Wording** text box.
    - Clicking **Upload an Audio File** to select a prerecorded audio file. After the file uploads, you can click **Preview Prompt** to listen to the audio file.
4. Click **Update**.



Prompt Wording: This is my message.

Prompt Type:Text to Speech

Prompt Ref: 0035

When configuring the prompts, enter Standard Responses, defined in UCS, in the following format:

```
StandardResponse:<text of response as defined in UCS>:Any fallback text
```

Important

The **fallback** field for the Standard Response is mandatory. See Standard Responses for configuring Standard Responses.

Using variables

You can configure prompts to reference variables that store dynamic data, such as account summaries or customer names. When you deploy the application, these variables are replaced with call-specific data.

Variables are created automatically by some prebuilt modules. For example, Payment Capture modules create variables such as **PaymentAmount** to store dynamic data about how much money the customer has chosen to pay.

You might also see variables when data is returned from a web service, or if you choose to set variables manually by using **Script** blocks.

To add a variable to a prompt, you must enter it using square-bracket notation. For example: `Your account number is [digits:AccountNumber]`.

> ### Important
> For more information on using variables, refer to the Script block page or the dynamic data section on the Prompts page.

Using shock absorbers

You can also add a Shock Absorber rule to a **Message** block, in which you state the message is only played to a particular subset of callers, or only played for a limited period of time. In other words, a Shock Absorber enables you to only play this message if the call meets specific conditions.

For example, if your contact center becomes unexpectedly busy, you can use a Shock Absorber to notify callers that wait times might be longer than usual.

> ### Important
> The ability to add a Shock Absorber depends on your license. Contact your Genesys representative for more information.

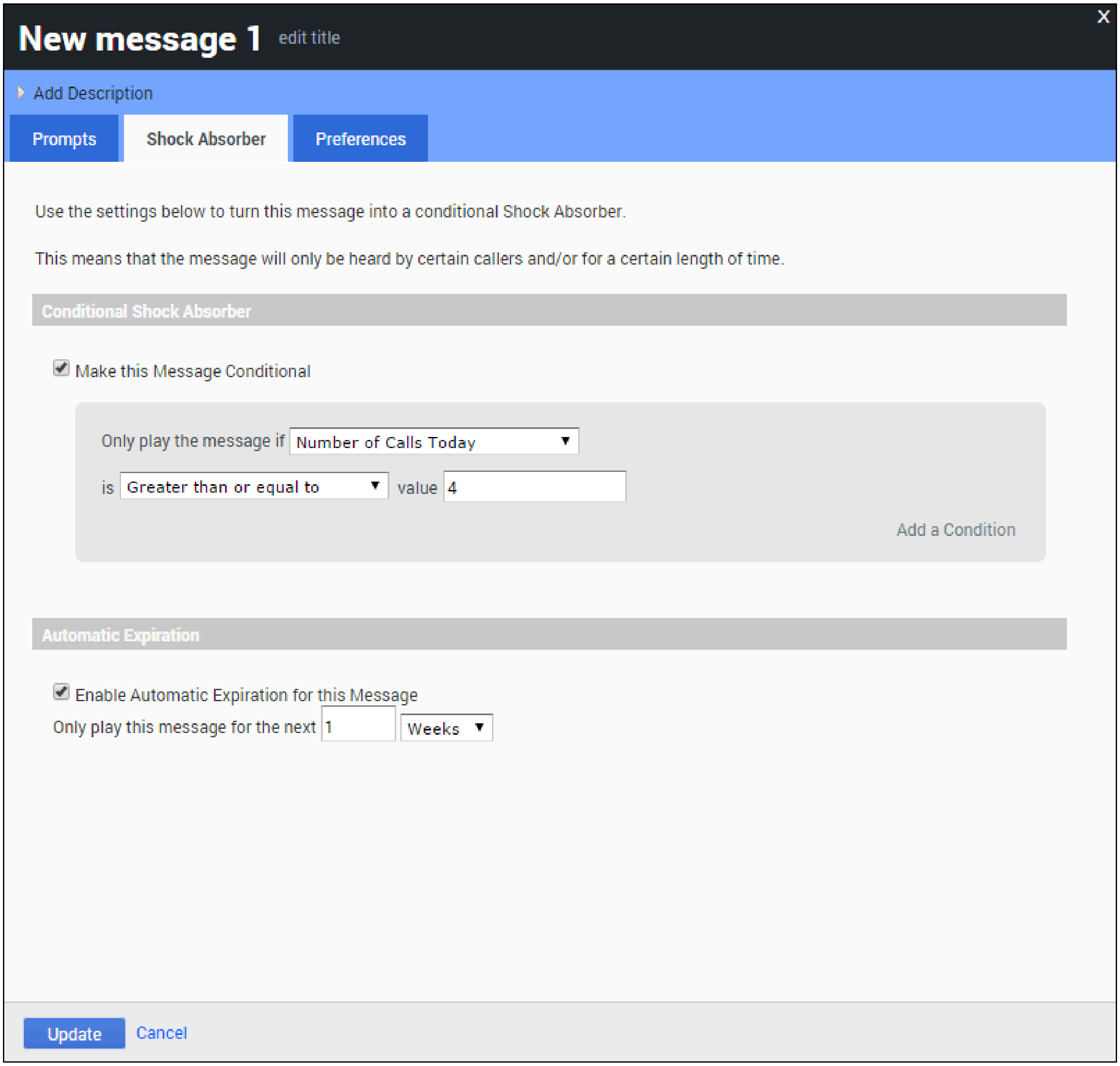
To add a Shock Absorber to a **Message** block:

1. Click a **Messsage** block to open its properties.
2. Click the **Shock Absorber** tab.
3. Enable the **Make this Message Conditional** check box to enable the Shock Absorber.
4. In the **Only play the message if** menu, select and then configure one of the following conditions:
   - **Variable** – Enable the Shock Absorber based on whether the specified variable (including attached

data fields) has a specified value.

- **Dialled Number** – Enable the Shock Absorber based on the number that the caller dialled (DNIS).
- **CLI** – Enable the Shock Absorber based on the number from which the caller has dialled.
- **Test Call Flag** – Enable the Shock Absorber based on whether this is a test call or a production call.
- **Recent Failure Flag** – Enable the Shock Absorber based on whether there has been a recent failure in the callflow.
- **Random Percentage** – Enable the Shock Absorber for a specified percentage of callers. For example, you can direct 50 percent of callers down a particular path to help split the call volume between two phone transfer points.
- **Last Result** – Enable the Shock Absorber based on the result from the block previous to this **Message** block (for example, success or error).
- **Number of Calls** ... – Enable the Shock Absorber based on the number of calls made by this CLI today or in the last one, two, or four weeks.
- **Date** - Enable the Shock Absorber for a specific date.
- **Time** - Enable the Shock Absorber for a specific time.
- **Date and Time** - Enable the Shock Absorber for a specific date and time.
- **Current Day of the Week** - Enable the Shock Absorber for a specified day of the week.

5. (Optional) Select the **Enable Automatic Expiration for this Message** check box to specify the length of time for which you want this message to be played. For example, you might configure the Shock Absorber to play this message only if this is the fourth call today from this CLI *and* to only play it for the next week. After a week, the message stops playing, regardless of how many calls come from this CLI. See the graphic below for an example.
6. Click **Update**.



Use the settings below to turn this message into a conditional Shock Absorber.

This means that the message will only be heard by certain callers and/or for a certain length of time.

Conditional Shock Absorber

☑ Make this Message Conditional

Only play the message if Number of Calls Today

is Greater than or equal to value 4

Automatic Expiration

☑ Enable Automatic Expiration for this Message

Only play this message for the next 1 Weeks

Using rich media with this block

If you are using a chat-based Persona, you can add rich media elements to this block by selecting the **Show Rich Media Editor** checkbox, which displays a new section called **Rich Media Editor** below the **Initial** prompt.

Enter the following information:

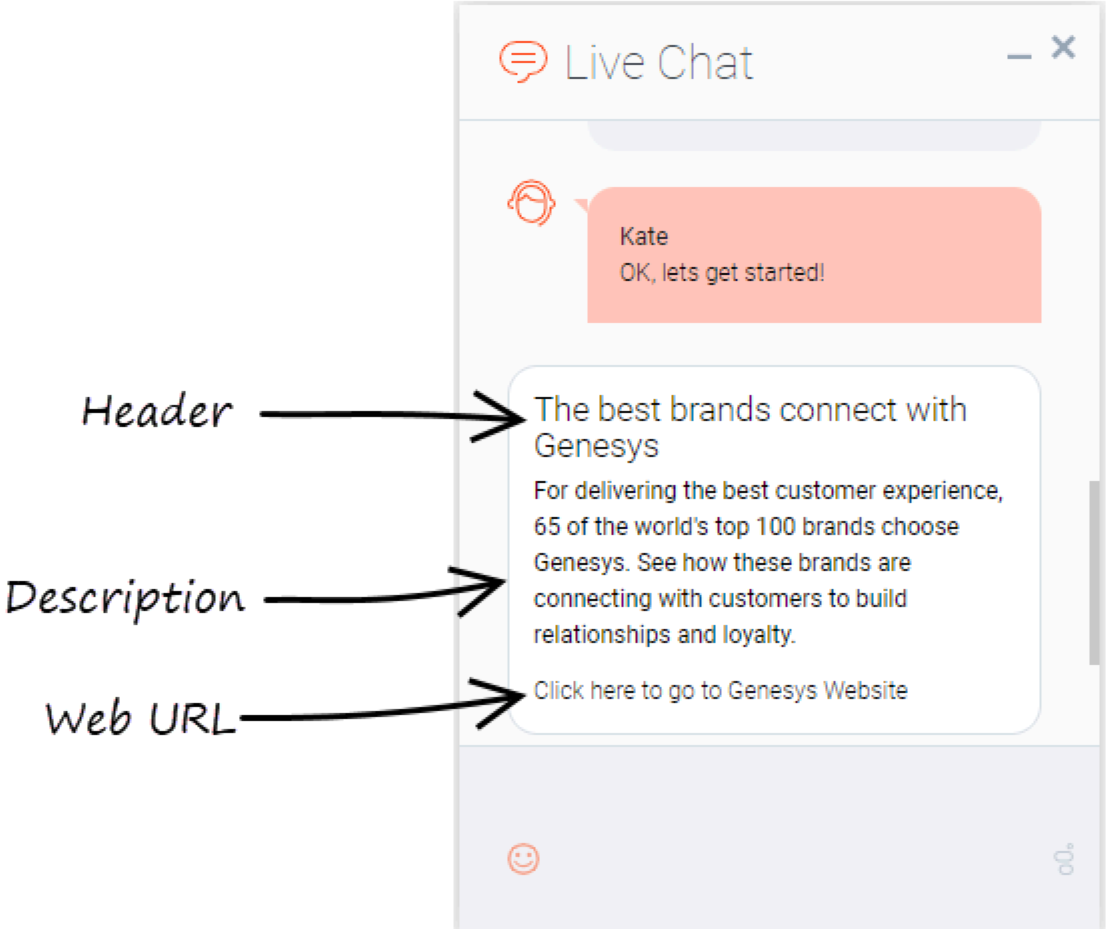
- **Image**—Enter a URL for an image, or click **Upload a File** to upload an image from your computer.
- **Video**—Enter a URL for a video, or click **Upload a File** to upload a video from your computer. To display a static image before the user plays the video, you must also enter a URL in the **Image** field.

> ## Important
> You must click **Update** to preview the image or video that you attached to this rich message.

- **Header**—Enter title text for the rich media message.
- **Description**—Enter description text that appears below the **Header** text.
- **Web URLs**—Enter a web URL that the customer can select to get more information. It must be in this format: `[link:URL;Description]`. For example: `[link:https://www.genesys.com;Genesys]`. In the preceding example, *https://www.genesys.com* is the URL that opens if the button is selected, and *Genesys* is the name of the button or link that the customer must select to open the URL.

> ## Important
> The appearance of certain elements, such as **Web URL**, depends on the **Rich Media Format** callflow preference. For example, the **Text Only** format displays links as text, but **Text and Buttons** displays links as buttons.

The following graphic shows how each element might appear in a typical chat widget. Your chat widget might appear differently, depending on your environment's configuration and style rules.



Kate
OK, lets get started!

The best brands connect with Genesys

For delivering the best customer experience, 65 of the world's top 100 brands choose Genesys. See how these brands are connecting with customers to build relationships and loyalty.

Click here to go to Genesys Website

Important

Some users can also set callflow preferences and default server settings to configure how Intelligent Automation processes rich media. See the Setting Callflow Preferences page for more information.

For more information about rich media functionality, refer to the Rich Media page in this manual.

Menu Block

You use **Menu** blocks to set up multiple-choice questions to ask callers, to which they respond using either speech or DTMF input.

A single **Menu** block asks a question several times, if necessary, until it gets an expected response from the caller. This is done using different types of prompts. If required, a **Menu** block can also ask the caller to confirm the answer. The **Menu** block uses the following prompt types:

Prompt Type	Description	Example
Initial	The first prompt played to the caller.	`Do you want to proceed?`
Retry	Played to the caller if his or her response was not recognized. Usually, the **Retry** prompt repeats the initial question and can suggest a few phrases that the caller can use in response.	`Say yes, no, or help. Do you want to proceed?`
Timeout	Played to the caller if no response was received (in other words, the caller was silent). Usually, the **Timeout** prompt repeats the initial question using slightly different phrasing to give the caller another chance to answer. A caller might encounter the **Timeout** prompt because he or she didn't understand the question, whereas a **Retry** prompt is used for when a response was received but it wasn't recognized.	`Do you want to proceed? You can say yes, no, or help.`
Help	Played to the caller if they say "help." The **Help** prompt includes additional information, such as a recap of where the caller is in the application, what they are being asked for and why, what to do if they do not want to continue, and so on. This prompt ends by asking the caller to answer the initial question.	`You have chosen to pay your bill by credit card. Do you want to continue? You can say yes or no.`

> ### Important
> **Menu** blocks inherit confirmation prompts from the main application. If necessary, you can set up different confirmation prompts for individual **Menu** blocks.

Using multiple retry and timeout prompts

> ### Important
> - You must have the **Application Designer** role, or higher, to use this feature. If you are not using standard roles, you must have the **configure_number_entry_retry_timeout_prompts** permission.
> - The availability of this feature is dependent on whether:
>   - In Default Server Settings, the setting **Prompts.RetryTimeout.MaxAllowed** has a value greater than 1 (as of the 9.0.101.00 release, the default value is 2).
>   - In the application or module details page, the **Allowed number of retry/timeout prompts** has a value greater than **1**.

In some situations, you might want to offer different iterations of a retry or timeout prompt to assist the caller. For example:

- *Caller inputs incorrect information.*
- *Retry 1*: "Please try again."
- *Caller inputs incorrect information.*
- *Retry 2*: "It looks like we are having trouble with your response. Press 0 to speak with an agent for assistance."

If properly configured (see note above), you can add multiple retry and timeout prompts to this block by clicking **Show more retry prompts** or **Show more timeout prompts** to see the additional prompts.

If the callflow requires additional retry or timeout prompts, Intelligent Automation repeats the last prompt. For example, if your environment supports three retry prompts and the caller requires a fourth retry attempt, the third retry prompt will repeat for all additional retries.

Adding this block to the callflow

To add and configure **Menu** blocks to a callflow:

1. Drag and drop the **Menu** block onto the appropriate position on the callflow.

2. Click the **Menu** block to open its properties.
3. In the **Prompts** tab, update the prompts for **Initial**, **Retry**, **Timeout**, and **Help**. As with Message blocks, the prompts can contain references to variables.
4. In the **Menu Options** tab, set the following options:
   - **Add Yes/No Answers**

     Select this option to add **Yes** and **No** menu options to your callflow. If you select this option and click **Update**, **Yes** and **No** paths are automatically added from this **Menu** block. You can drag and drop these **Yes** and **No** paths onto blocks (either in the toolbox or in the callflow diagram) to which you want them to route.

     > Important
     >
     > If you click **Update** to active this option and then later decide to remove the **Yes** and **No** answers from your callflow, you must drag and drop them onto the Recycle Bin. You cannot de-select the **Add Yes/No Answers** option after it is activated.

   - **Store Answer as a Variable**

     Select this option to store the caller's answer as a variable so it can be played back in later prompts or passed to web services. To set up the variable, enter a name for the new variable and select from these options:

     - **Attach it to the Call (CTI)** – If enabled, Genesys Intelligent Automation attaches the variable to the call for use with other third-party systems. For example, if the call later routes to a contact center, you can set up a strategy so the agent can see the information from this variable before speaking with the caller.
     - **Remember it When Calling From the Same Number** – If enabled, Intelligent Automation stores the variable against the calling number in the database. Therefore, any subsequent calls made from that number will already have the information stored against it. If you select this option, callers receive the ability to *Skip This Question Next Time*, so they are not asked this menu option again if they happen to call back. For example, if you asked the caller if he or she is an existing customer, and the response is "yes," the caller is not asked this question again in subsequent calls.
5. Click **Update**.
6. In the callflow, attach appropriate blocks to the **Menu** block to represent the menu options available.

> Important
>
> - Attaching data to calls consumes system resources, so use this option sparingly.
> - Consider the privacy implications when using the **Remember it When Calling From the Same Number** option. Intelligent Automation can recognize some shared numbers, but not all. In these cases, it might be unwise to attach sensitive information, such as account numbers.
> - While the CLI helps to indicate the identity of a caller, Genesys recommends you confirm the caller's identity before authorizing account changes.

Using rich media with this block

If you are using a chat-based Persona, you can add rich media elements to this block by selecting the **Show Rich Media Editor** checkbox, which displays a new section called **Rich Media Editor** below the **Initial** prompt.

Enter the following information:

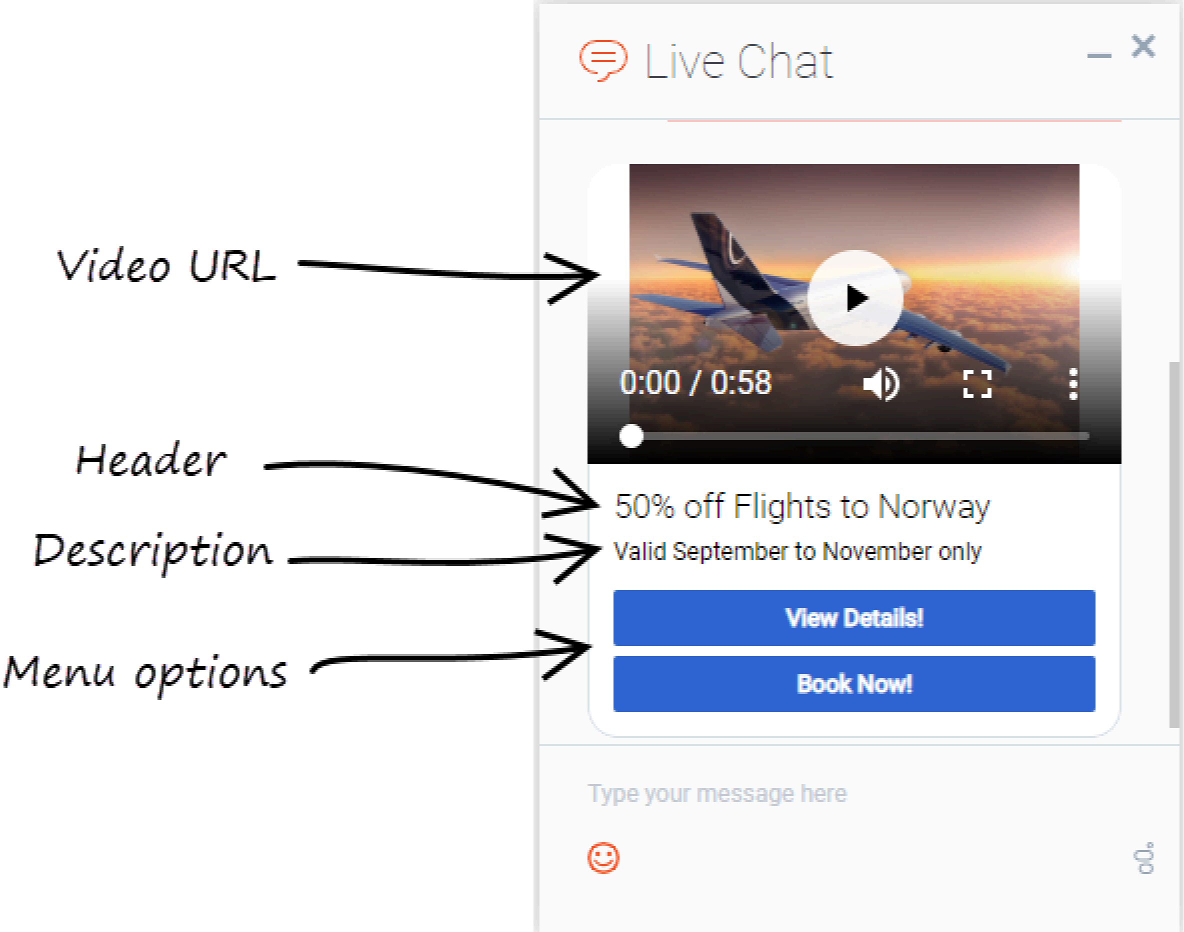
- **Image**—Enter a URL for an image, or click **Upload a File** to upload an image from your computer.
- **Video**—Enter a URL for a video, or click **Upload a File** to upload a video from your computer. To display a static image before the user plays the video, you must also enter a URL in the **Image** field.

> ### Important
> You must click **Update** to preview the image or video that you attached to this rich message.

- **Header**—Enter title text for the rich media message.
- **Description**—Enter description text that appears below the **Header** text.
- **Submit Button Text**—This field is not currently supported.
- **Web URLs**—Enter a web URL that the customer can select to get more information. It must be in this format: `[link:URL;Description]`. For example: `[link:https://www.genesys.com;Genesys]`. In the preceding example, *https://www.genesys.com* is the URL that opens if the button is selected, and *Genesys* is the name of the button or link that the customer must select to open the URL.

> ### Important
> The appearance of certain elements, such as **Web URL**, depends on the **Rich Media Format** callflow preference. For example, the **Text Only** format displays links as text, but **Text and Buttons** displays links as buttons.

The following graphic shows how each element might appear in a typical chat widget. Your chat widget might appear differently, depending on your environment's configuration and style rules.



50% off Flights to Norway

Valid September to November only

Important

Some users can also set callflow preferences and default server settings to configure how Intelligent Automation processes rich media. See the Setting Callflow Preferences page for more information.

Question Block

Important

Only users with the role **Application Designer** can add **Question** blocks to a callflow. However, both **Application Designer** and **Application Maintainer** roles can view and update **Question** block properties.

You can use **Question** blocks to ask complex questions to callers, such as credit card numbers, dates, or currency amounts. Whereas Menu blocks tend to ask simple questions with limited answers (for example "yes" or "no"), **Question** blocks can use grammars (standard or customized) to specify the format of expected answers. A grammar defines the acceptable words, phrases, commands, or even sentences that the **Question** block accepts as valid input.

Important

Contact your Genesys representative for more information on grammars written in SRGS.

As it might not be possible to create paths from your **Question** block to cover every distinct answer from a caller, Genesys recommends you use a default **success** path to accommodate all successful recognitions (unless a path exactly matches the recognized answer).

Adding this block to the callflow

To add and configure Question blocks on a callflow:

1. Drag and drop the **Question** block onto the appropriate position in the callflow.
2. Click the **Question** block to open its properties.
3. In the **Vocabulary** tab, click the **Mode** menu and select the type of grammar you want to use. You can choose from:
    - **Standard Grammar** – Use a built-in grammar. You can select options such as **date** or **credit card number**.
    - **Grammar Builder** – Define a custom simple grammar format. For example, if the **Standard Grammar** option does not have an option to cover your particular business need, you can use the **Grammar Builder** option to build your own solution. For example, you can define a grammar that

accepts two digits followed by two to five letters.

> ## Important
> For more details, refer to Using the Grammar Builder, below.

- **Upload Custom Grammar** – Upload a SRGS XML grammar file(s). For example, you could set up a **Question** block that asks callers for a complex account number that contains a check digit or a small number of possible alpha characters. This can help to ensure better recognition accuracy. You can upload a new voice grammar as well as a DTMF grammar.

> ## Important
> - See the Standard Grammar settings section below for more information on standard grammars.
> - Contact your Genesys representative for information on the exact format of the SRGS XML grammar file to use.

You might choose to upload both a DTMF grammar and a voice grammar to ensure either response is properly recognized. For example, if the caller speaks their account number and the speech recognizer does not understand what was said, the caller can choose to enter DTMF digits instead.

4. In the **Prompts** tab, add prompts for **Initial**, **Retry**, **Timeout**, and **Help**. These prompts can contain references to variables. See the Menu block page for more information on prompts.
5. (Optional) In the **Questions Options** tab, select the **Store Answer as a Variable** check box if you want to store any answer provided by the caller as a variable. See the Menu block page for more information on working with variables.
6. Click **Update**.

Standard Grammar settings

This section explains the settings available for the Standard Grammars.

> ## Important
> Intelligent Automation supports masking user input entries in Visual IVR applications from 9.0.111.x version onwards. If the **Confidential Mode** setting is enabled, Visual IVR masks the strings instead of displaying the actual information. This setting does not mask the currencies, dates, numbers, and phone number

> grammar types.

Credit Card Expiry

Parameter	Description
MinAllowed	The minimum number of months for which a credit card expiry date can be out of date. The default is **-36**, which is three years ago.
MaxAllowed	The maximum number of months for which a credit card expiry date can be in the future. The default is **36**, which is three years from the current date.

Credit Card Number

This grammar allows you to specify the types of credit cards that can be accepted from a caller. Setting a type of credit card to **true** allows it to be used; setting a type to **false** disables it.

Currency EUR and Currency GBP

Parameter	Description
MinAllowed	The absolute minimum amount in Euros/Pounds that can be accepted from a caller.
MinExpected	The minimum amount of Euros/Pounds expected from a caller. If an amount does not fall between this value and the **MaxExpected** amount, the speech-recognition engine assigns it a higher confidence score.
MaxExpected	The maximum amount of Euros/Pounds expected from a caller. If an amount does not fall between this value and the **MinExpected** amount, the speech-recognition engine assigns it a lower confidence score.
MaxAllowed	The absolute maximum amount in Euros/Pounds that can be accepted from a caller.

Date

Parameter	Description
MinAllowed	The absolute minimum number of days before the current date that a date given by a caller can be accepted (for example, input -365 for a year before the current date).
MinExpected	The earliest date expected from a caller. Any date between this value and the **MaxExpected** date

Parameter	Description
	results in the speech-recognition engine assigning a high confidence score.
MaxExpected	The latest date expected from a caller. Any date between this value and the **MinExpected** date results in the speech-recognition engine assigning a high confidence score.
MaxAllowed	The absolute maximum number of days after the current date that a date given by a caller can be accepted.
AllowAmbiguousCentury	Specify whether to accept the century portion of a date from a caller. For example, for the year 2012, a caller may say "twenty-twelve", "two-thousand and twelve," or omit the century completely.
AllowAmbiguousYear	Specify whether to accept the year portion of a date from a caller. This allows a caller to use a two-digit year or four-digit year for dates after the year 2000. For example, a caller might say "twenty twelve" or "two thousand and twelve."
DateFormat	Specify the date format. Starting from 9.0.110.23 release, Question blocks now support the YMD, DMY, and MDY date formats.

Natural Numbers

Natural numbers are positive integers, such as 1, 2, 3, and so on.

Parameter	Description
MinAllowed	The absolute smallest natural number that can be accepted from a caller.
MinExpected	The smallest natural number expected from a caller. Any amount between this value and the **MaxExpected** number results in the speech-recognition engine assigning a high confidence score.
MaxExpected	The maximum natural number expected from a caller. Any amount between this value and the **MinExpected** number results in the speech-recognition engine assigning a high confidence score.
MaxAllowed	The absolute highest natural number that can be accepted from a caller.

Phone Number Republic of Ireland (RoI) and Phone Number UK

Parameter	Description
AllowMobiles	Specifies whether a RoI/UK mobile number provided by the caller can be accepted.

Parameter	Description
AllowLandlines	Specifies whether a RoI/UK landline number provided by the caller can be accepted.

Postcode UK

This grammar allows you to specify that only UK-based postcodes can be accepted from the caller.

Time

Parameter	Description
Assume24Hour	Specifies that all times given by callers will be in 24-hour-clock time. For example, when set to **true**, if a caller says "eight thirty", the speech-recognition engine assume this to mean 8:30 a.m. rather than 8:30 p.m.
MinAllowed	The earliest time that will be accepted from a caller.
MinExpected	The earliest time expected from a caller. Any time between this and the **MaxExpected** time results in the speech-recognition engine assigning a high confidence score.
MaxExpected	The latest time expected from a caller. Any time between this and the **MinExpected** time results in the speech-recognition engine assigning a high confidence score.
MaxAllowed	The latest time that will be accepted from a caller.

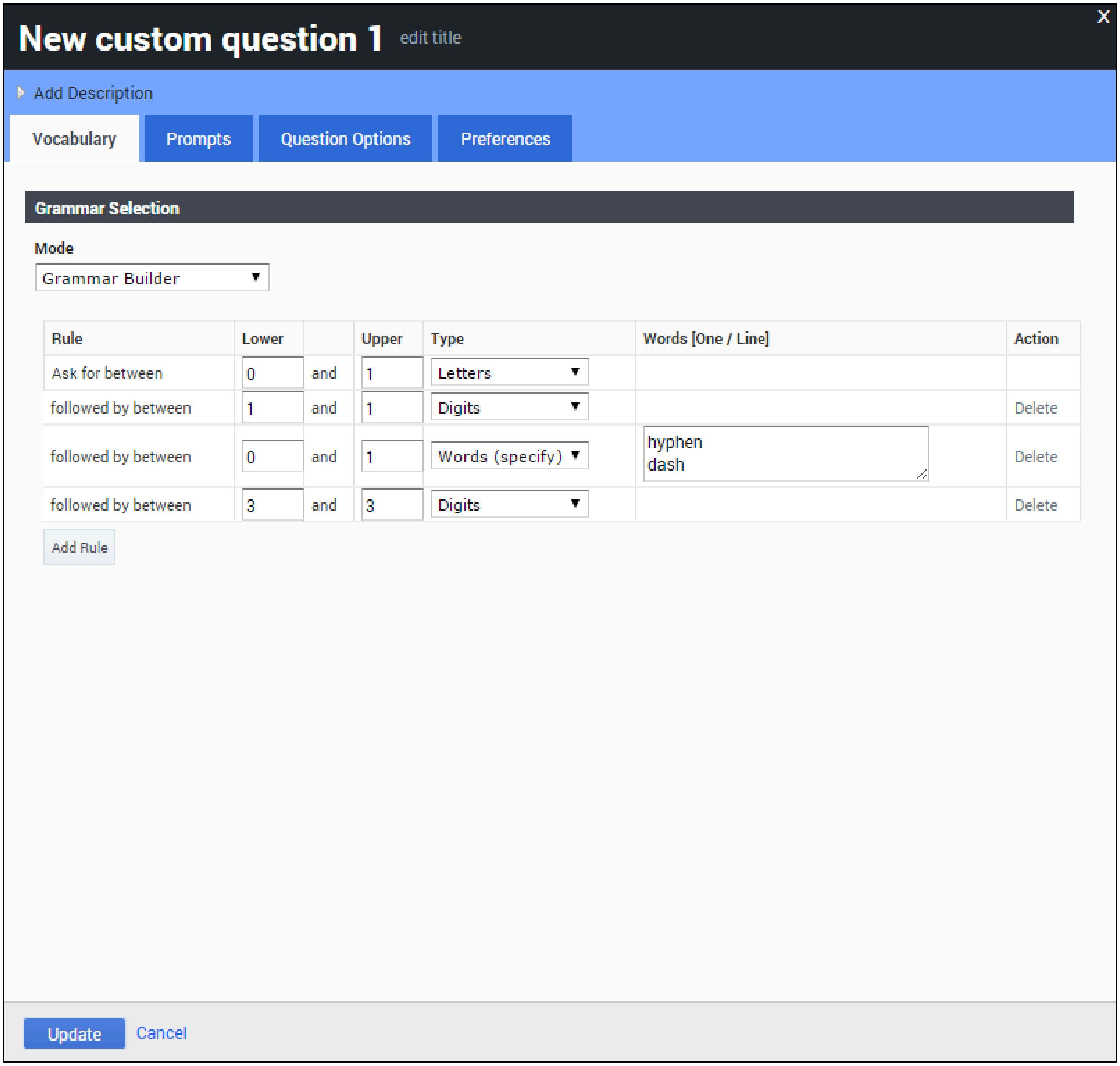
Sample Grammar Snippet

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar
xml:lang="en-us"
mode="voice"
tag-format="semantics/1.0"
version="1.0"
xmlns="http://www.w3.org/2001/06/grammar"
root="root">
<rule id="root" scope="public">
        <one-of>
            <item>
                <one-of>
                    <item>1</item>
                </one-of>
                <tag>
                    out.MEANING = 'mango';
                    out.dm_confirmation_mode = 'NEVER';
                    out.dm_confirm_string = '';
                    out.IsFullConfirmationPrompt = true;
                    out.SlotData = '';
                </tag>
            </item>
            </one-of>
    </rule>
</grammar>
```

Parameter	Description
xml:lang	The language code for the XML. This could be 'en-us' or 'en-gb'.
mode	The type of input that the user agent should be detecting. The default mode is "voice" for speech recognition grammars. An alternative is 'dtmf'.
tag-format	The tag-format identifier that indicates the content type of all rule tags and header tags contained within a grammar. Set it to 'semantics/1.0'.
version	The version number of the grammar element indicates which version of the grammar specification is being used — '1.0'. The grammar version is a required attribute.
xmlns	The XML namespace being used. This is a required parameter.
root	The root rule of the grammar. This must be in lowercase.
out.MEANING	The output generated based on the user input.
out.dm_confirmation_mode	Specify if a user confirmation is required for the phrase or option collected. The possible options are: • 'NEVER' would not ask for a confirmation. • 'ALWAYS' would always ask for a confirmation regardless of the confidence levels. • 'IF_NECESSARY' would ask for a confirmation if the response falls below the high confidence threshold, but above the low threshold, for voice recognition.
out.dm_confirm_string	This message is played dependent on the **confirmation mode** setting.
out.IsFullConfirmationPrompt	Used to confirm that the system has correctly recognized the answer provided by the caller
out.SlotData	Defines the slot that matches the phrase. For example, 'expiry date' could be defined as the slot data when requesting credit card information.

Using the Grammar Builder

The **Grammar Builder** option allows you to define your own simple grammar format. For example, you can set up a grammar for a customer account number that contains a specific mixture of letters, dashes and numbers. Consider the following grammar:



Mode: Grammar Builder

Rule	Lower		Upper	Type	Words [One / Line]	Action
Ask for between	0	and	1	Letters		
followed by between	1	and	1	Digits		Delete
followed by between	0	and	1	Words (specify)	hyphen dash	Delete
followed by between	3	and	3	Digits		Delete

Using the rules in the previous example, the custom grammar accepts any of the following phrases from a caller:

- "B nine four five six"
- "B nine dash four five six"
- "B nine hyphen four five six"
- "nine four five six"
- "nine dash four five six"

- "nine hyphen four five six"

To use the **Grammar Builder** option:

1. In the **Lower** field, enter the minimum amount of digits, letters, or words that a caller must provide. If caller does not need to provide a response for this rule, enter `0` in this field.
2. In the **Upper** field, enter the maximum number of digits, letters, or words that the caller must provide.

   In the example above, the rule specified that the caller must provide between **0** and **1** letters. In other words, the caller might provide a letter, or not. If you do require the caller to provide one letter only, set the **Lower** and **Upper** values to 1 and 1, respectively.
3. In the **Type** menu, select from the following:
   - **Digits**
   - **Non-Zero Digits**
   - **Letters**
   - **Digits or Letters**
   - **Words (specify)** – If selected, enter the words that this grammar accepts. For example, dash or hyphen. The grammar might recognize these words, but they are not included in the final result. Therefore, if the caller says, "one two dash four", a result of "124" is recognized.
4. (Optional) To add another rule, click **Add Rule**.
5. Click **Update** to save the grammar.

Using multiple retry and timeout prompts

Important

- You must have the **Application Designer** role, or higher, to use this feature. If you are not using standard roles, you must have the **configure_number_entry_retry_timeout_prompts** permission.
- The availability of this feature is dependent on whether:
  - In Default Server Settings, the setting **Prompts.RetryTimeout.MaxAllowed** has a value greater than 1 (as of the 9.0.101.00 release, the default value is 2).
  - In the application or module details page, the **Allowed number of retry/timeout prompts** has a value greater than **1**.

In some situations, you might want to offer different iterations of a retry or timeout prompt to assist the caller. For example:

- *Caller inputs incorrect information.*

- *Retry 1*: "Please try again."
- *Caller inputs incorrect information.*
- *Retry 2*: "It looks like we are having trouble with your response. Press 0 to speak with an agent for assistance."

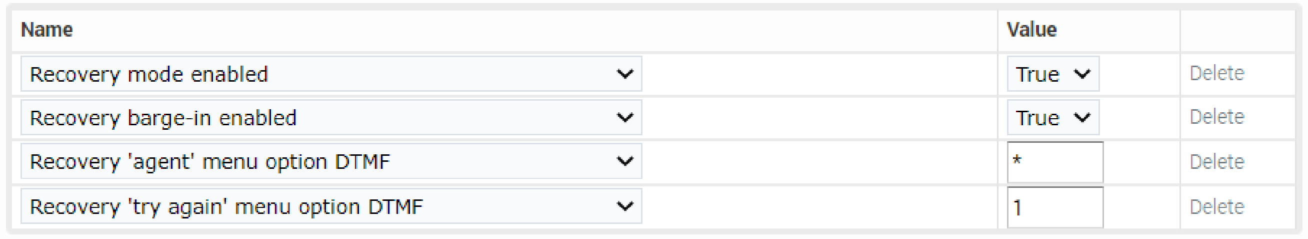
If properly configured (see note above), you can add multiple retry and timeout prompts to this block by clicking **Show more retry prompts** or **Show more timeout prompts** to see the additional prompts.

If the callflow requires additional retry or timeout prompts, Intelligent Automation repeats the last prompt. For example, if your environment supports three retry prompts and the caller requires a fourth retry attempt, the third retry prompt will repeat for all additional retries.

Using recovery modes

When the recovery options are configured in the **VUI Preferences** page and the block's maximum tries count is reached, the recovery mode is enabled and a recovery prompt is played asking the user to confirm if the callflow is to be transferred to an agent (`DTMF *`) or if the block is to be reprocessed (`DTMF 1`).

- **Recovery mode enabled** - This setting enables or disables the recovery option mechanism on the specific block.
- **Recovery barge-in enabled** - This setting enables or disables the barge-in option for the upcoming recovery option prompt.
- **Recovery 'try again' menu option DTMF** - This setting allows configuring a different DTMF value. The default value, `DTMF 1` is used for the 'try again' option.
- **Recovery 'agent' menu option DTMF** - This setting allows configuring a different DTMF value. The default value, `DTMF *` is used for the agent transfer option.



Name	Value	
Recovery mode enabled	True	Delete
Recovery barge-in enabled	True	Delete
Recovery 'agent' menu option DTMF	*	Delete
Recovery 'try again' menu option DTMF	1	Delete

Using rich media with this block

If you are using a chat-based Persona, you can add rich media elements to this block by selecting the **Show Rich Media Editor** checkbox, which displays a new section called **Rich Media Editor** below the **Initial** prompt.

Enter the following information:

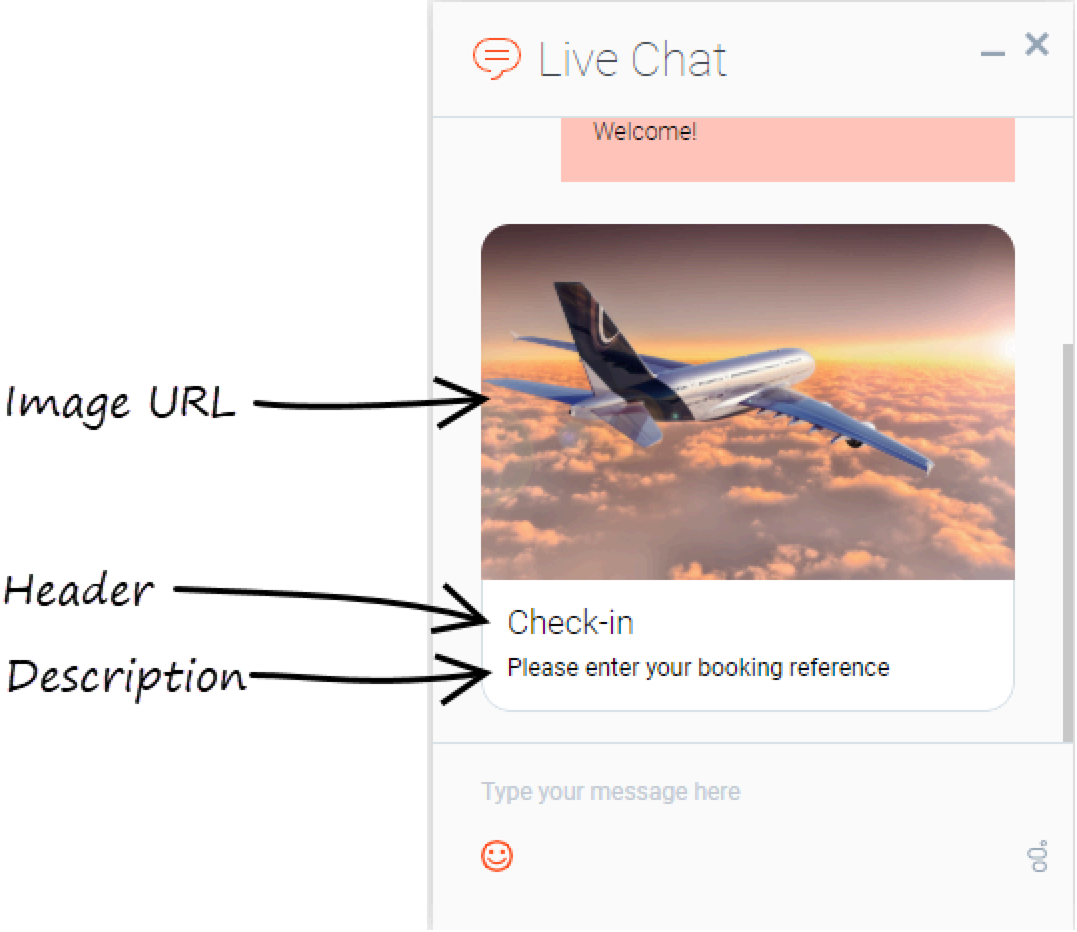
- **Image**—Enter a URL for an image, or click **Upload a File** to upload an image from your computer.
- **Video**—Enter a URL for a video, or click **Upload a File** to upload a video from your computer. To display a static image before the user plays the video, you must also enter a URL in the **Image** field.

> Important
>
> You must click **Update** to preview the image or video that you attached to this rich message.

- **Header**—Enter title text for the rich media message.
- **Description**—Enter description text that appears below the **Header** text.
- **Submit Button Text**—This field is not currently supported.
- **Web URLs**—Enter a web URL that the customer can select to get more information. It must be in this format: `[link:URL;Description]`. For example: `[link:https://www.genesys.com;Genesys]`. In the preceding example, *https://www.genesys.com* is the URL that opens if the button is selected, and *Genesys* is the name of the button or link that the customer must select to open the URL.

> Important
>
> The appearance of certain elements, such as **Web URL**, depends on the **Rich Media Format** callflow preference. For example, the **Text Only** format displays links as text, but **Text and Buttons** displays links as buttons.

The following graphic shows how each element might appear in a typical chat widget. Your chat widget might appear differently, depending on your environment's configuration and style rules.



Check-in

Please enter your booking reference

Important

Some users can also set callflow preferences and default server settings to configure how Intelligent Automation processes rich media. See the Setting Callflow Preferences page for more information.

For more information about rich media functionality, refer to the Rich Media page in this manual.

Phone Block

You can use **Phone** blocks to transfer a call to an external phone number or SIP address. The following results are possible:

- **success**
- **no answer**
- **busy**
- **hangup**
- **error**

Adding this block to the callflow

To add and configure **Phone** blocks on a callflow:

1. Drag and drop the **Phone** block onto the appropriate position in the callflow.
2. Click the **Phone** block to open its properties.
3. In the **Transfer Details** tab:
    - Enter the number to which you want to transfer the caller. You can enter:
        - A phone number.
        - A SIP address (for example, `sip:12345@myserver.com`).
        - Multiple numbers or SIP addresses, each separated by a comma. The **Phone** block will try each one until it gets an answer or they all fail.
        - A phone number or SIP address in the form of a variable name (for example, `[var:MyDestinationNumber]`).

      Genesys Intelligent Automation validates entered phone numbers to ensure calls are not transferred to any disallowed numbers (for example, premium-rate numbers). Disallowed numbers can be changed by your system administrator.
    - (Optional) Select the **Use Whisper Transfer** checkbox and configure the prompt you want to play to the agent during a Whisper Transfer. The prompt can include variables, such as `[digits:AccountNumber]` or `[digits:WhisperID]`. See the Whisper Transfer section, below, for more information.
4. (Optional) Apply an Opening Hours rule to this call to specify at which times and which days calls can be transferred to a specific number. For example, you might not want to transfer to a number during weekends. If a call is made outside of opening hours, a specific prompt plays and the call can transfer to a different phone number or another result as determined by the Opening Hours rule.

> ### Important
>
> At least one Opening Hours rule must be set up in the current module for the Opening Hours rule list to be visible.

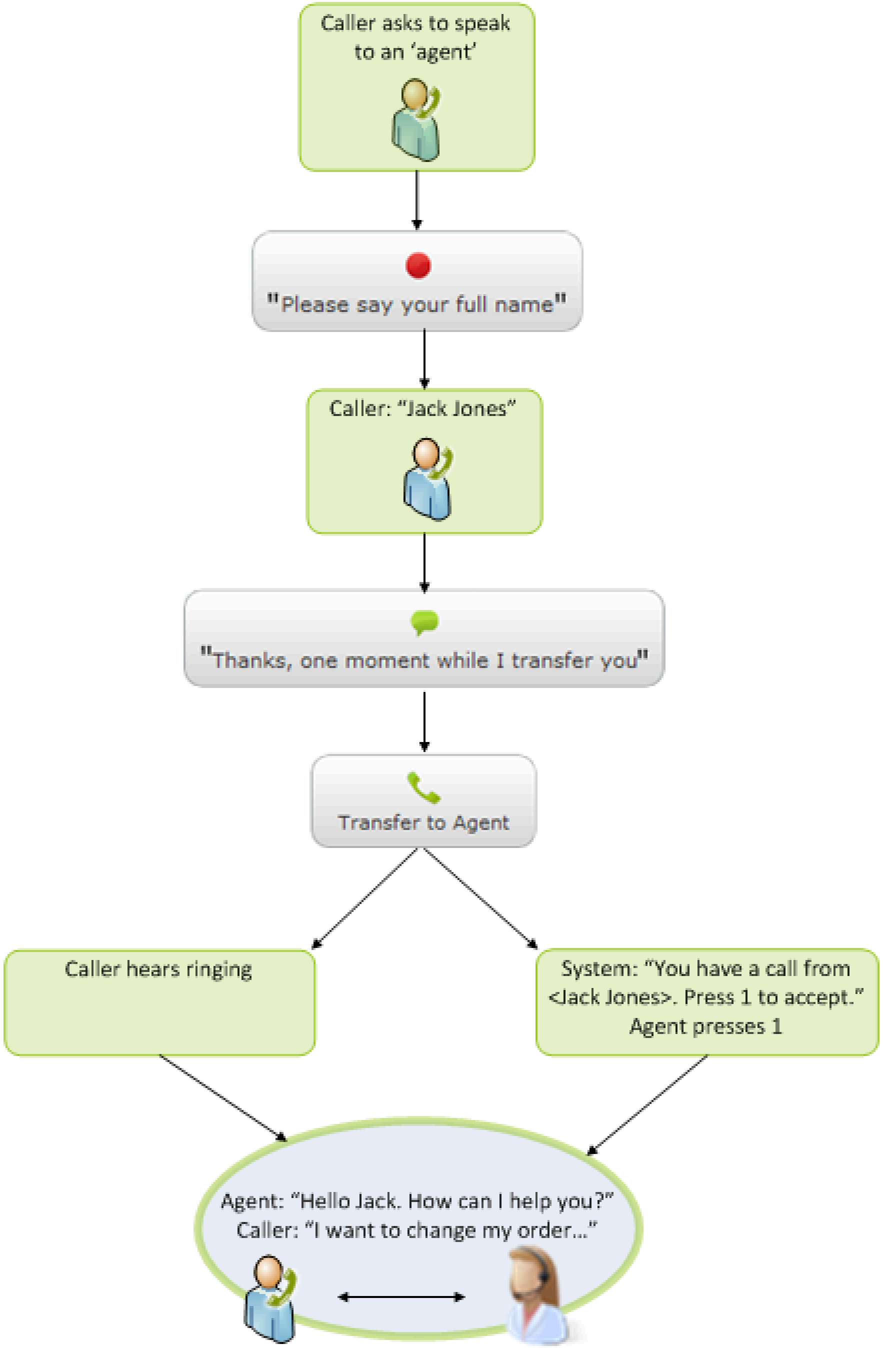
5. Click **Update**.

Whisper Transfer

This feature allows you to supply the agent with basic details, such as the caller's full name, before the agent speaks to the caller. You can turn on Whisper Transfers for all transfers or for individual transfers.

The agent must provide a DTMF response when provided with the whisper prompt. For example, you might use: `Press 1 to accept the call or 2 to decline it`. Or, you can add the account number: `You have a call from [digits:AccountNumber]. Press 1 to accept or 2 to decline.`

If you have provided the **WhisperID** variable as part of the prompt, the agent can use the CTI Viewer to view the details of the call. For example, you might use: `The code is [digits:WhisperID]. Press 1 to accept or 2 to decline.`


Caller asks to speak to an 'agent'
"Please say your full name"
Caller: "Jack Jones"
"Thanks, one moment while I transfer you"
Transfer to Agent
Caller hears ringing
System: "You have a call from <Jack Jones>. Press 1 to accept." Agent presses 1
Agent: "Hello Jack. How can I help you?"
Caller: "I want to change my order..."

The following parameters determine the maximum number of times the Whisper Transfer prompt is played to the agent:

- **Maximum no input count** – Defines the maximum number of times a prompt is played to an agent before it times out due to lack of input from the agent.
- **Maximum retry count** – Defines the maximum number of times the retry prompt is played to the agent. The retry prompt may be played to the agent if he or she has selected an invalid option.

If one of these maximums is reached and you still want the call to connect to an agent, you can add the **Connect on whisper transfer timeout** option in the **Preferences** tab. Otherwise, the call returns a **no answer** result.

If the Whisper Transfer is in a queue, you can add the **Two-step whisper transfer** parameter in the **Preferences** tab to specify what happens if the Whisper Transfer is in a queue. When enabled, this parameter plays an introductory message to the agent ("Press **9** to hear the message") in a loop until the agent is ready to hear the actual Whisper Transfer. At this point, the agent can press **9** to hear the message or **8** to reject the call.

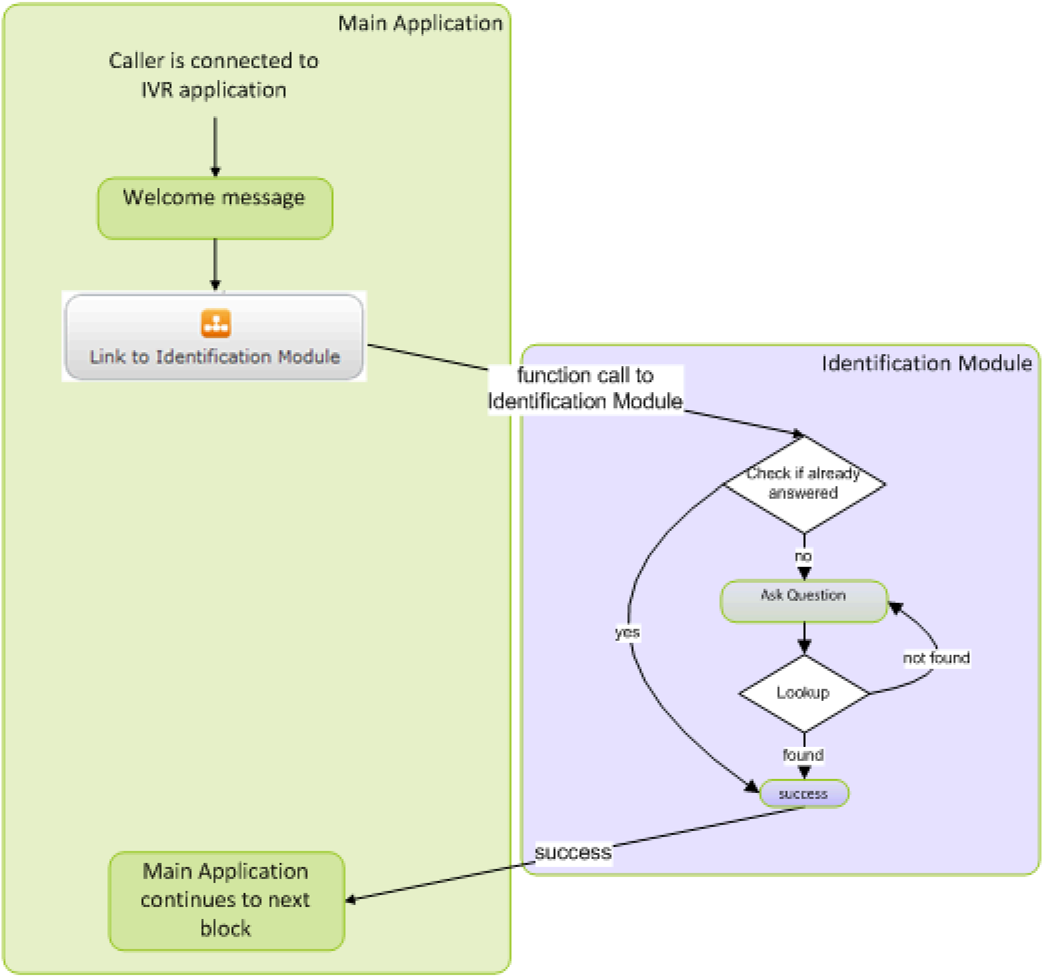
Important

- The maximum number of times the introductory message can be played to the agent is dictated by the length of time specified in the **Transfer timeout** parameter — not the number of iterations. Once the agent presses **9** to hear the actual whisper message, the **Maximum no input count** and **Maximum retry count** parameters determine how many times the message is played.
- By default, Whisper Transfer uses a built-in WAV file (**whispertransfer_intro.wav**) for the introductory message, but you can override this by uploading a custom prompt named `Standard whisper transfer introduction.wav`.

Link Block

You can use **Link** blocks to send a call within your callflow to another application, module, or external VXML application. Once the external callflow finishes, the call returns to the point at which it left in the parent callflow.

For example, the main application might start with a Message block to welcome the caller and then use a **Link** block to call into an Identification module, which allows the caller to identify himself or herself. When the module finishes, the call routes back to the main application with a **success** result and the rest of the application callflow executes.


Main Application
Caller is connected to IVR application
Welcome message
Link to Identification Module
function call to Identification Module
Identification Module
Check if already answered
no
yes
Ask Question
Lookup
not found
found
success
success
Main Application continues to next block

You can also assign values to variables in the Link Block. See Working with module parameters.

Adding this block to the callflow

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

1. Drag and drop a **Link** block onto the appropriate position in the callflow.
2. Click the **Link** block to view its properties.
3. In the **Link Details** tab, select one of the following options:
   - **Transfer to Another Genesys App Automation Platform Module** – Call a Genesys Intelligent Automation module within the same Company. This links to the Start block within the module.

     > ### Tip
     > You can click **Jump to Module** to open the linked module in the Callflow Editor. If you have unsaved changes in your current application, Intelligent Automation asks you to leave the page without saving.

   - **Transfer to an External VXML Application** – Use an HTTP URL to call into a VXML application. After this application finishes processing the call, it must return specific information back to the Intelligent Automation application (as defined in the third-party application contract), as well as the path that must be followed as a result of the call. You can use variables when specifying the URL, such as: `http://[var:server]:[var:port]/MyPage.vxml`.

     > ### Important
     > Contact your Genesys representative for more information on configuring the results from a VXML application.

     If the caller hangs up while in the VXML application portion of the call, or the VXML application wants to end the call, the VXML application must return a **hangup** or **system hangup** result, respectively.

Interceptor Block

You can use **Interceptor** blocks to add branching to a callflow. Although not as powerful as a Script block, **Interceptor** blocks are easier to read and change.

You must set up one or more rules within an **Interceptor** block in order to branch the callflow. You can also call a web service before applying these rules, which allows you to gather some back-end data mid-call without having to resort to a Script block or separate prebuilt callflow module.

> **Important**
>
> Do not use the Interceptor block for catching hangups. Use the `Scripting.EndCallNotification.WebServiceURL` option to configure the end-of-call service.

Adding this block to the callflow

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

1. Drag and drop an **Interceptor** block onto the appropriate position in the callflow.
2. Click the **Interceptor** block to view its properties.
3. In the **Interceptor Rules** tab, configure one or more rules to apply to the **Interceptor** block. The block applies these rules in the order they are listed. To configure a rule:
    a. Select one of the following conditions to check:
        - **Variable** – Select any variable (including attached data fields).
        - **Dialled Number** – The number the caller dialled (DNIS).
        - **CLI** – The number from which the caller has called, if available.
        - **Test Call Flag** – Check whether this is a test or production call.
        - **Recent Failure Flag** – Check whether there has been a recent failure in the callflow (for example, a block that exited with **recognition failure** or **error**.
        - **Random Percentage** – Specify a percentage of callers to follow the path you are configuring. For example, you can specify a value of `50` to direct 50 percent of callers down a particular path, to help split the volume of traffic between two phone transfer points.
        - **Last Result** – Check the result from the last block, typically **success**, **error**, or **agent**.
        - **Number of Calls** – Specify the number of calls by this CLI today or in the last one, two, or four weeks.

- **Date** - Check for a specific date.
- **Time** - Check for a specific time.
- **Date and Time** - Check for a specific date and time.
- **Current Day of the Week** - Check for a specified day of the week.

b. Apply a rule to the condition. For example, if you selected **Random Percentage**, you must configure a value or range or values to apply. Or, if you selected **Date**, you must specify a date.

c. Specify a path name to follow if the condition is met.

d. Specify a path name to follow if the condition is not met.

4. (Optional) Click **Add Another Rule** to configure another rule, following the same process documented above.
5. In the **Integration** tab, you can specify a web service to call before applying the rules in the **Interceptor Rules** tab. Any variables in the call session, as well as the CLI and call session ID, are passed to the web service. The web service then returns information to Genesys Intelligent Automation to allow it to set variables in the callflow.

> **Important**
>
> You can pass URLs as an absolute path (e.g., http://URL/path) or as a session variable (e.g., [var:URLvariable]).

To specify a web service:

- **Web Service for Test Calls** - Enter the URL for the web service to use for test calls.
- **Web Service for Production Calls** - Enter the URL for the web service to use for production calls.
- **Web Service Timeout** - Enter the number of milliseconds to wait before the call to the web service times out due to a lack of response from the service.
- **Web Service Behavior** - If enabled, the **Interceptor** block ignores any errors returned by the web service.

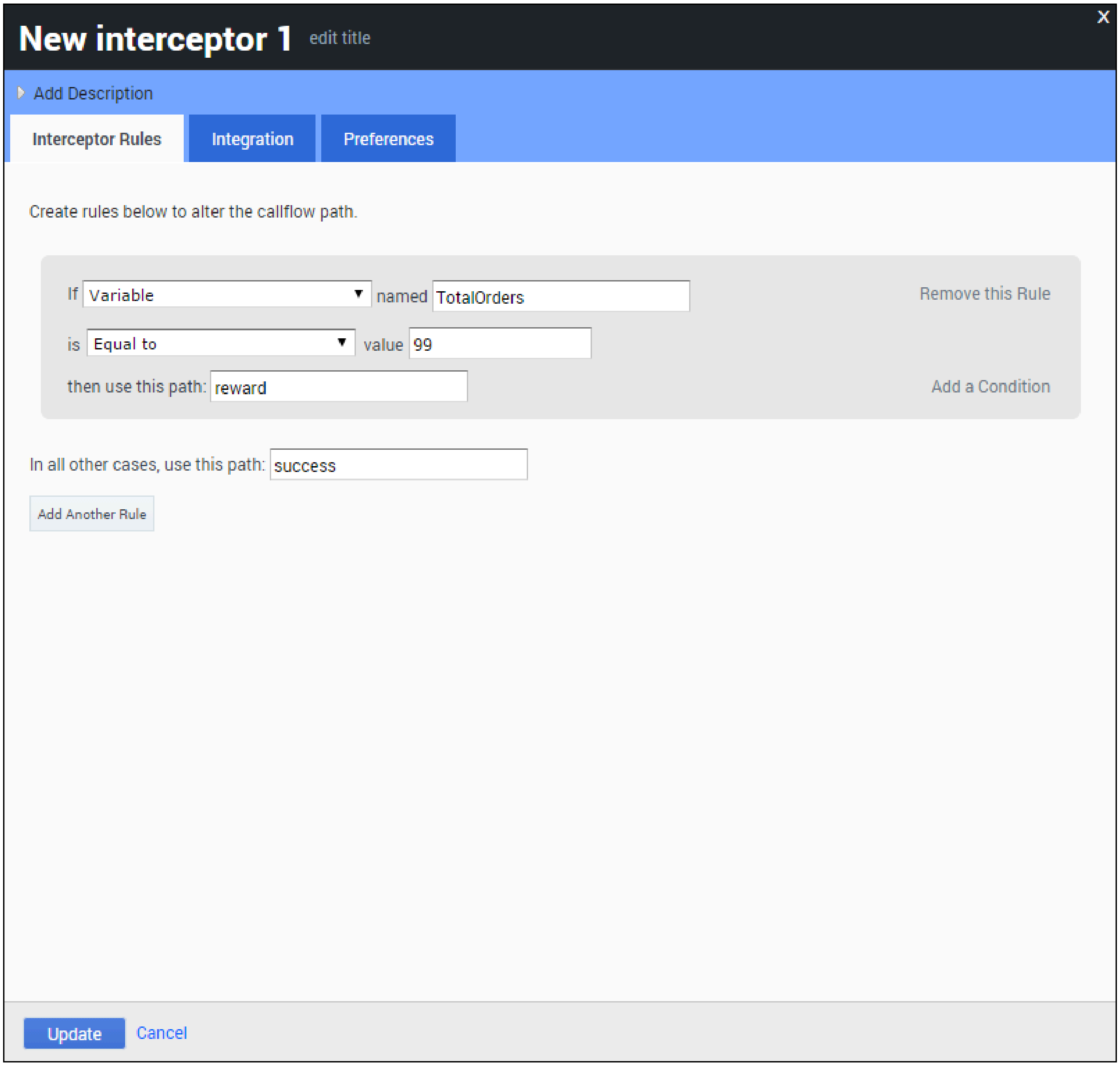
6. Click **Update**.

Example

You can set up an **Interceptor** block to do a lookup at the beginning of a call based on the CLI (which is automatically passed in), plus any information that has been stored against the CLI from a previous call (for example, a customer account number). You can then use the result to find out how many orders the caller has previously made (using a variable passed back by the web service), and route them to a reward line if they are calling to make their 100th order.

1. Drag and drop an **Interceptor** block onto the callflow.
2. Click the **Interceptor** block to view its properties.
3. Open the **Integration** tab and enter the URL for a Sales Order web service.

4. Any variables already gathered from the call up are passed into the web service, such as the CLI and customer account number. Therefore, you can open the **Interceptor Rules** tab and add the following rule:



New interceptor 1 edit title

Add Description

Interceptor Rules | Integration | Preferences

Create rules below to alter the callflow path.

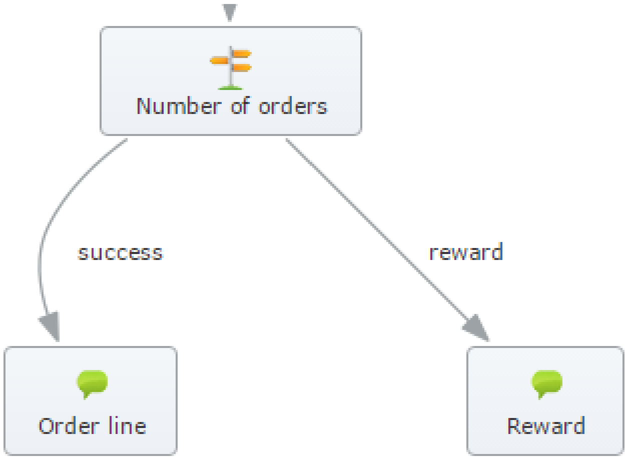
If Variable named TotalOrders

is Equal to value 99

then use this path: reward

In all other cases, use this path: success

5. Click **Update.**
6. The **Interceptor** block automatically creates the **reward** and **success** paths and accompanying Message blocks. Name these blocks `Reward` and `Order line`, respectively.



7. Open the **reward** path Message block.
8. In the **Prompts** tab, enter a message according the result of the web service call. For example: `Looking at your account, I see that today's order means you have ordered [number:TotalOrders] times with us. As a reward, you will receive a 50 percent discount on today's order. Now onto the order menu.`
9. Any calls that do not meet the interceptor rule follow the **success** path. Open the **success** path Message block and configure it accordingly.

Sample web service data contract

This section describes the flow of data between an **Interceptor** block and a customer's web service. This web service allows a back-end system to look up known information about the current caller.

Depending on the call steps, this information might include:

- Basic call information - Passed with all web service requests in Intelligent Automation (for example, **cli** and **sessionid**).
- Variables:
  - Captured during the call so far, such as an account number spoken by the caller, or variables set within Script blocks.
  - Information previously stored against the caller's CLI.

Request

You can make web service requests using HTTP POST, passing any string variables in the call session, plus the **cli** and **sessionid** fields.

A typical URL might be:

```
http://localhost:8080/fish-services/test/GetAccountDetails.jsp
```

A typical HTTP POST body might be:

```
cli=02890571100&sessionid=1234%2D3AAF%2D3372&favouriteMenu=Sales
```

Response

The XML response specifies zero or more variables to be set in the call session.

When specifying variables in the response, you can indicate that you want some or all key-value pairs to be attached to the call via the CTI (where the platform supports it) by doing one of the following:

- Include an optional **attach** attribute with a value of **true**
- Set as the CLI data by including an optional **remember** attribute with a value of **true**.

A typical XML response might be:

```
<interceptorResults>
       <variables>
              <variable name="AccountNumber" value="12345" attach="true"/>
              <variable name="PIN" value="1234"/>
              <variable name="Segment" value="Gold" attach="true" remember="true"/>
       </variables>
</interceptorResults>
```

A lookup in which you do not want to set any variables might be:

```
<interceptorResults/>
```

> **Important**
>
> Intelligent Automation treats HTTP response codes other than **200** as errors.

Script Block

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

Script blocks are great for adding ‘presentation’ logic to your app - comparing values and branching out to different blocks, performing date calculations, combining prompts. It’s possible to invoke RESTful web services directly from a Script block, and this may be sufficient if the services are simple. It is advised not to use the Script block for tasks like handling security certificates, or calling onto databases or SOAP-based web services.

Integration Hub is a simple and powerful way to do this, and brings many other benefits such as support for multiple environments’ endpoints (dev/test/production, for example) and the ability to create automated test scripts. Using Integration Hub is a good way to keep your Intelligent Automation apps and their assorted presentation logic separate from the details of how to call onto your enterprise’s backend systems.

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**.

To learn how to use the Script block to add custom rich media, refer to the Rich Media page in this manual.

Scripting methods

Click the link below to download a zip file that documents the scripting methods that you can call from the **Script** block.

- Script API Reference

Example

```
    //
    //  Call the Web Service (note that session variables can be included in the URL and we
then use context.expandVariables() to replace the variable with whatever is in session at
runtime)
    //
    def iTimeout = 5000;
    def sURL = "https://testwebservice.com/[var:WebServiceName]"

    sURL = context.expandVariables(sURL);
    def dataResult = context.getRemoteHttpData(sURL, "POST", params, iTimeout);

    def xml = dataResult.responseXml;


    //
    //  Parse the response
    //
    assert null != xml.status;
    assert "" != xml.status.text();

        def sStatus = xml.status.text();

    for (variableDeclaration in xml.variables?.variable)
    {
        def sName = variableDeclaration.@name;
        def sValue = variableDeclaration.@value;

        context.setVariable(sName, sValue);
    }

    if ("success" == sStatus)
    {
        context.log("Found");
        return "success";
    }
    else if ("not found" == sStatus)
    {
        context.log("Not found");

        return "not found";
    }
```

Unit Tests

Sample Script

```
if (context.getVariable(“AuthMethod”) == “AccountNumber”)
{
      return “account number”;
}
else
{
      return “social security number”;
}
```

Sample Unit Test

```
context.setVariable(“AuthMethod”, “AccountNumber”);
def result = callScript();
assert “account number” == result;

context.log(“Passed test”);
```

Splitting results based on recognition failures

To handle different behaviors in case of a recognition failure, you can use a script to split the results based on `maxnomatches` and `maxnoinputs`.

```
if (context.getLastResultDetail().contains("maxnomatches"))
{ return "max retries"
}
else {
return "max timeouts"
}
```

Recording Block

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

You can use **Recording** blocks to record free speech during a call. For example, you can record the caller's name for later use in a Whisper Transfer, or to record feedback at the end of a questionnaire call.

If you want to use the recorded audio later in the call, Genesys Intelligent Automation automatically creates a **LastRecording** variable for you to use in prompts. For example, if you record the caller's name and you want to play it back during a Whisper Transfer, use the following variable: `[var:LastRecording]`.

> **Important**
>
> **Recording** blocks do not perform voice recognition.

Adding this block to the callflow

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

1. Drag and drop a **Recording** block onto the appropriate position in the callflow.
2. Click the **Recording** block to view its properties.
3. In the **Prompts** tab, update the prompts for **Initial** and **Timeout**. The latter is used when no sound is detected.
4. (Optional) In the **Recording Options** tab, you can enable the **Save Recording After Call is Complete** check box to save the recording after the call is complete. Intelligent Automation saves this recording in the **resources** folder of the Tomcat VUI server.

   For example, you can save recordings to allow support staff to listen to a caller's response at a later time, as with questionnaire responses. However, you might not want to save recordings if you are only using them for Whisper Transfers. In this case, Intelligent Automation deletes the recording after it finishes handling the call.
5. (Optional) In the **Preferences** tab, you can configure **Maximum recording timeout** and **Recording**

complete timeout to specify the maximum duration of the recording.

6. Click **Update**.

Using multiple retry and timeout prompts

Important

- You must have the **Application Designer** role, or higher, to use this feature. If you are not using standard roles, you must have the **configure_number_entry_retry_timeout_prompts** permission.
- The availability of this feature is dependent on whether:
  - In Default Server Settings, the setting **Prompts.RetryTimeout.MaxAllowed** has a value greater than 1 (as of the 9.0.101.00 release, the default value is 2).
  - In the application or module details page, the **Allowed number of retry/timeout prompts** has a value greater than **1**.

In some situations, you might want to offer different iterations of a timeout prompt to assist the caller. For example:

- *Caller does not provide recording.*
- *Retry 1*: "Please try again."
- *Caller does not provide recording.*
- *Retry 2*: "It looks like we are having trouble with your response. Press 0 to speak with an agent for assistance."

If properly configured (see note above), you can add multiple timeout prompts to this block by clicking **Show more timeout prompts** to see the additional prompts.

If the callflow requires additional timeout prompts, Intelligent Automation repeats the last prompt. For example, if your environment supports two timeout prompts and the caller requires a third timeout attempt, the second timeout prompt will repeat for all additional retries.

Rich Media

If you're using a chat-based persona, you can add rich media elements (an image or a video, for example) to chat interactions so that users can interact and engage with message content.

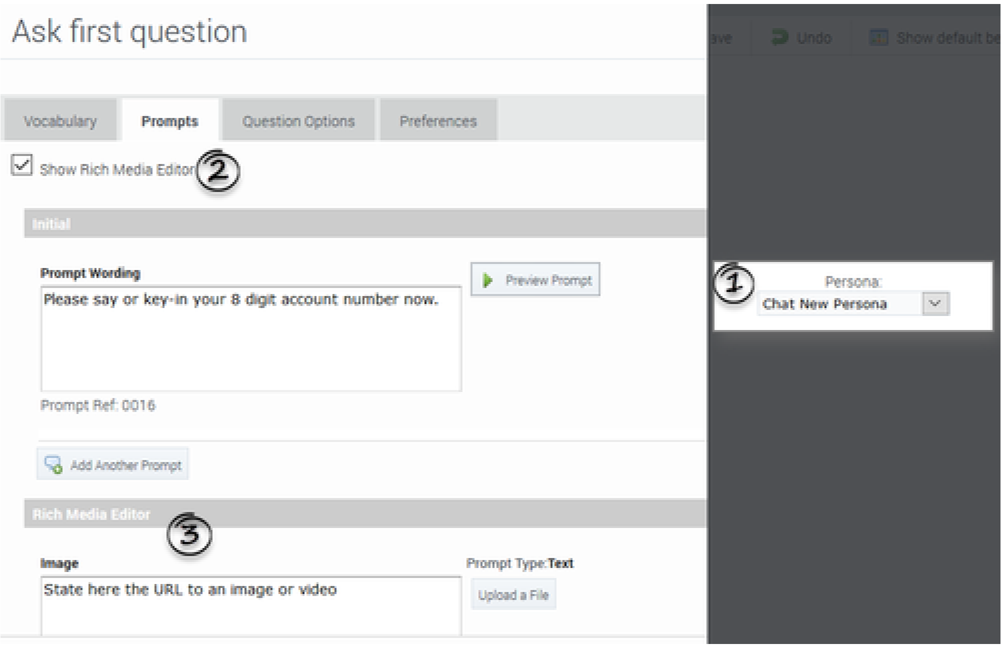
In Intelligent Automation you can add and configure rich media using the rich media editor for any Message block, Menu block, or Question block, or you can create custom rich media using a Script block. Both options are described below.

> **Important**
>
> An application can contain both custom rich media and natively-generated rich media (rich media created using the rich media editor), but a single block can only contain one or the other - not both.

Adding rich media using the rich media editor

To add rich media to a Message block, Menu block, or Question block:

1. Go to the Prompts tab for the block and ensure you're using a chat persona.
2. Click the **Show Rich Media Editor** checkbox.
3. In the new Rich Media Editor section, fill in the fields described in the table below.



Ask first question

☑ Show Rich Media Editor

Prompt Wording: Please say or key-in your 8 digit account number now.

Prompt Ref: 0016

Persona: Chat New Persona

Image: State here the URL to an image or video

Prompt Type: Text

Option	Description
Image	Enter a URL for an image, or click **Upload a File** to upload an image from your computer.
Video	Enter a URL for an image, or click **Upload a File** to upload an image from your computer.
Header	Enter title text for the rich media message.
Description	Enter description text that appears below the **Header** text.
Submit Button	This field is not currently supported.
Web URLs	Enter a web URL that the customer can select to get more information. It must be in this format: `[link:URL;Description]`. For example: `[link:https://www.genesys.com;Genesys]`. In the preceding example, *https://www.genesys.com* is the URL that opens if the button is selected, and *Genesys* is the name of the button or link that the customer must select to open the URL. Important The appearance of certain elements, such as **Web URL**, depends on the **Rich Media Format** callflow preference. For example, the **Text Only** format displays links as text, but **Text and Buttons** displays links as buttons.

Next, set the following rich media callflow preferences described below. Note that these callflow preferences only apply if you're adding rich media using the rich media editor. They do not apply if you add custom rich media.

Callflow Preference	Desription	Example/Notes	Typical Values
Rich Media Format	Specifies the output format for rich media message options.	Select **Text, Buttons and Videos** to use a rich media format that incorporates all of these elements. If **None** is selected, all rich media prompts are ignored. Important You can also set the following default server settings: • **Resources.Allowe** list of audio file mime types to allow users to upload. • **Resources.Allowe** list of image file mime types to allow users to upload. • **Resources.Allowe** list of video file mime types to allow users to upload.	
Rich Media try fallback if necessary	Specifies whether to try a fallback format if the selected **Rich Media Format** is incorrectly configured (for example, if **Text, Buttons and Videos** is selected but a video is not attached). In other words, this setting determines whether Intelligent Automation validates the user's configuration or simply attempts to use what it is given.	Select **True** if you want Intelligent Automation to validate the user's rich media format. If the format is incorrect, Intelligent Automation tries to select an alternate format.	False

Adding custom rich media

Creating custom rich media is an alternative to using the rich media editor in a Message, Menu, or Question block.

Important

When adding custom rich media, the callflow preferences described above do not apply. If these preferences are set as callflow defaults, override them with **None**.

To add custom rich media:

1. In a Script block, create an object based on the definition described on the Rich Messaging page in the *Genesys Widgets Deployment Guide*.
2. Convert that object to a JSON string using the **context.toJsonString()** method.
3. Use that JSON string to create a native data prompt with the context.**createNativePrompt()** method
4. Store the native data prompt in a variable.
5. Reference that variable within a chat persona fallback prompt (for example, **[var:MyNativePrompt]**)

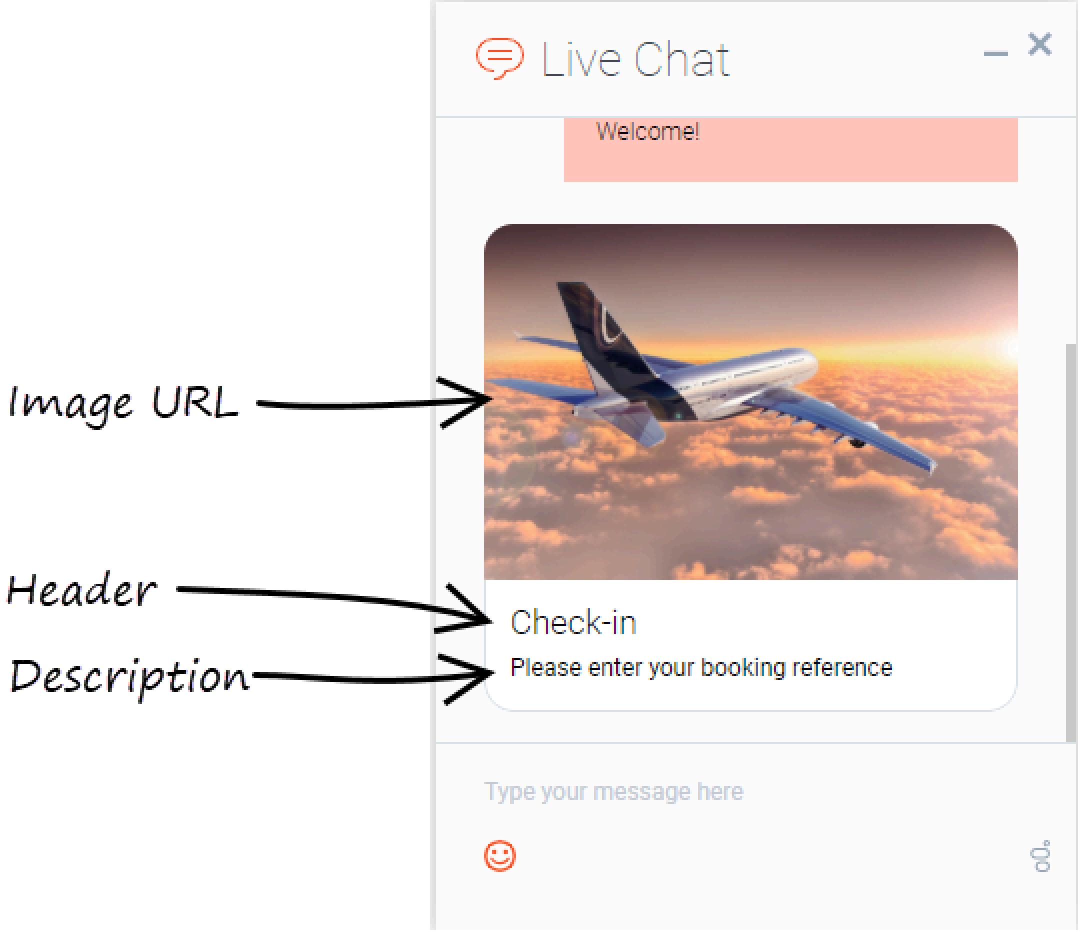
You can send also rich media related properties as optional parameters (JSON key-value pairs) as part of the message using the **createNativePrompt()** method:

```
optionField = ["type" : "Button"];
optionFieldJSON = context.toJsonString(optionField);
custom = context.createNativePrompt( "QuickReply3", sJson, "Genesys EngageStructuredMessage",
"facebook-messenger",optionFieldJSON);
context.setVariable( "MyNativePrompt", custom);
```

Rich media examples

The following examples show how each element might appear in a typical chat widget. Your chat widget might appear differently, depending on your environment's configuration and style rules.

[+] Example chat with image element

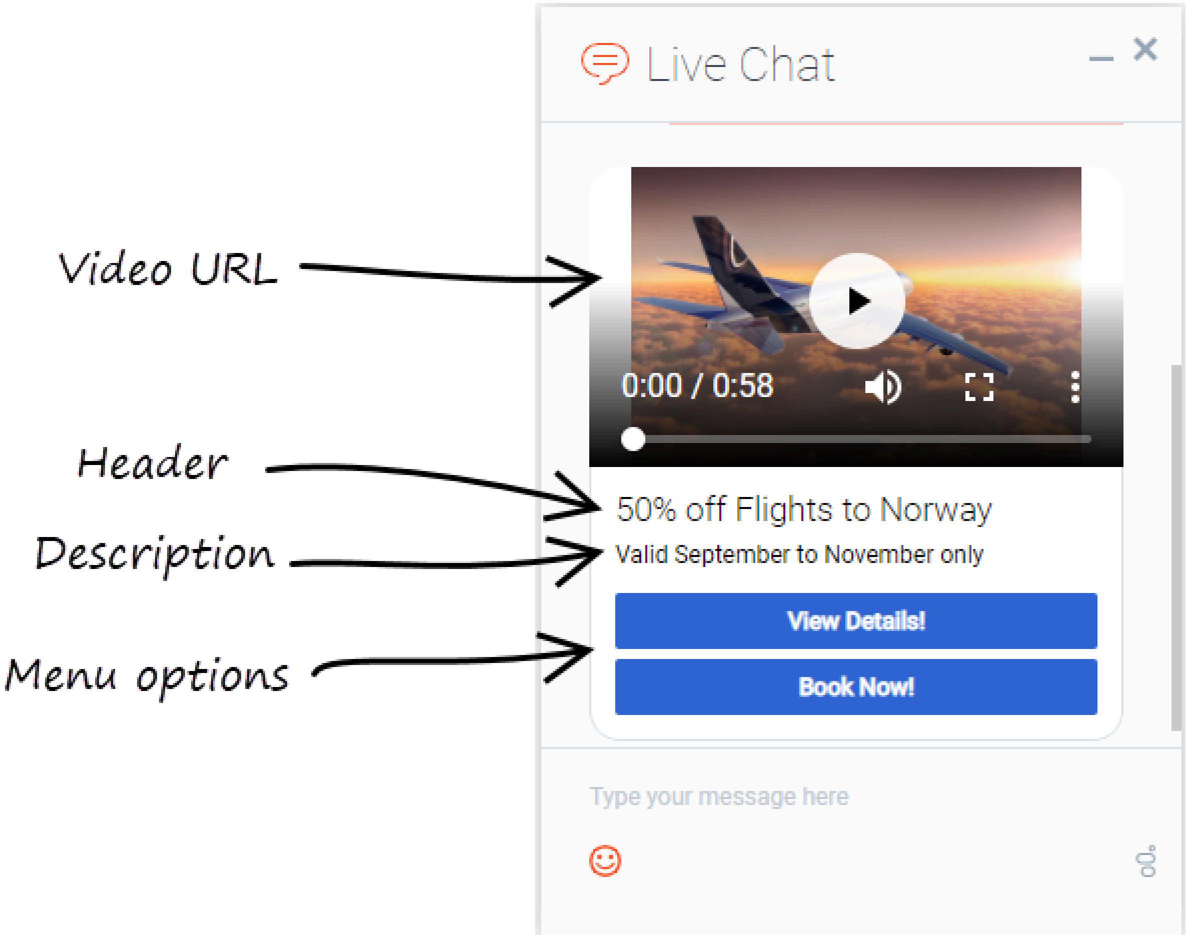


Welcome!

Check-in

Please enter your booking reference

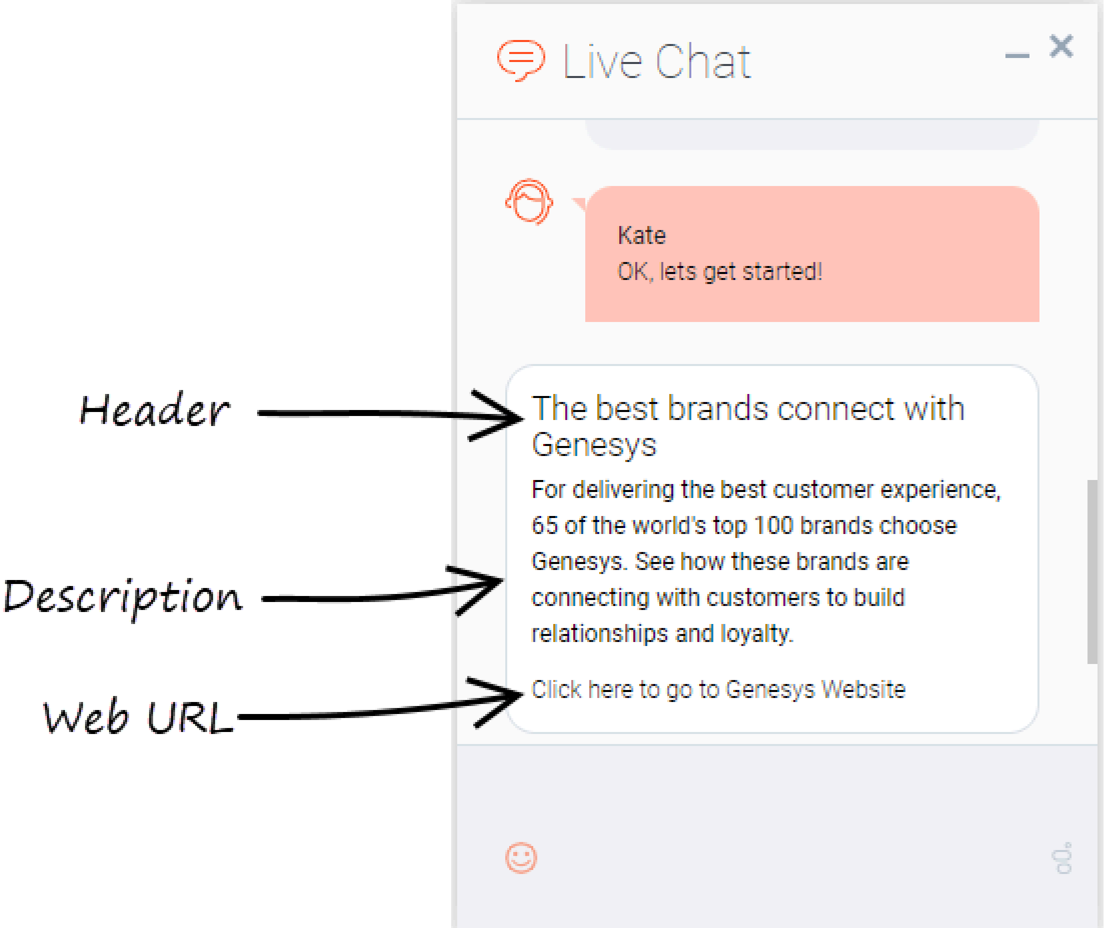
[+] Example chat with video element



50% off Flights to Norway

Valid September to November only

[+] Example chat with header and description prompt



Kate
OK, lets get started!

The best brands connect with Genesys

For delivering the best customer experience, 65 of the world's top 100 brands choose Genesys. See how these brands are connecting with customers to build relationships and loyalty.

Click here to go to Genesys Website

Related Topics

- Message Block
- Menu Block
- Question Block
- Script Block
- Setting Callflow Preferences
- Rich Messaging (Genesys Widgets Deployment Guide)

Natural Language AI Integration

Intelligent Automation can integrate with external artificial intelligence (AI) services to provide natural language understanding (NLU) capabilities.

> **Important**
>
> The Natural Language Menu App and Custom Natural Language Menu are shipped separately and only allowed in conjunction with the purchase of either the bot bundle/orchestration either Voicebot or Chatbot bundle. Get in touch with your AE or Sales.

When Intelligent Automation connects to a natural language engine, Intelligent Automation allows a user to use natural language when responding, even entering multiple pieces of information in a single response. The natural language engine infers the relevant information contained in a user's response, enabling some subsequent Menu and Question blocks in the call flow to be skipped, because the information has already been captured. The result is a successful conversation that is shorter than that of the directed-dialog approach.

> **Warning**
>
> Please note that the customer is responsible for ensuring that the environment and bot applications they build are properly configured and secured according to PII and HIPAA requirements.

How it works

The natural language engine works in the background while Intelligent Automation communicates directly with the user. Here's how:

1. Intelligent Automation asks an open-ended question (For example, *What can I help you with?*)
2. The user provides a natural language response (For example, *What will the weather be like in London on Saturday*?).
3. Intelligent Automation receives the response and passes it along to the natural language engine.
4. The natural language engine responds, as follows:
    - If the natural language engine needs more information to understand the request, it responds to Intelligent Automation with follow-up questions. (For example, *For what city?*).
    - If the natural language engine has all the information it needs, it sends Intelligent Automation the Intent and associated Slots (In the Weather example, the Intent could be Weather and the Slots could be weatherLocation and weatherDate).

5. When Intelligent Automation receives and reads the Intents and Slots, it directs the interaction according to the configured application.

The following demonstrates a chat session with a natural language engine running in the background:

Link to video

Intelligent Automation currently has built-in connections to Genesys Dialog Engine and Google Dialogflow (the v2 API) . If you want to use other natural language AI services, contact your Genesys representative for help with a custom integration.

How to integrate with natural language AI services

To integrate with a natural language AI service, complete the following steps:

1. Configure Default Server Settings (applies to Genesys Dialog Engine only)
2. Configure NLU Settings
3. Map Intents to modules
4. Map Slots to questions

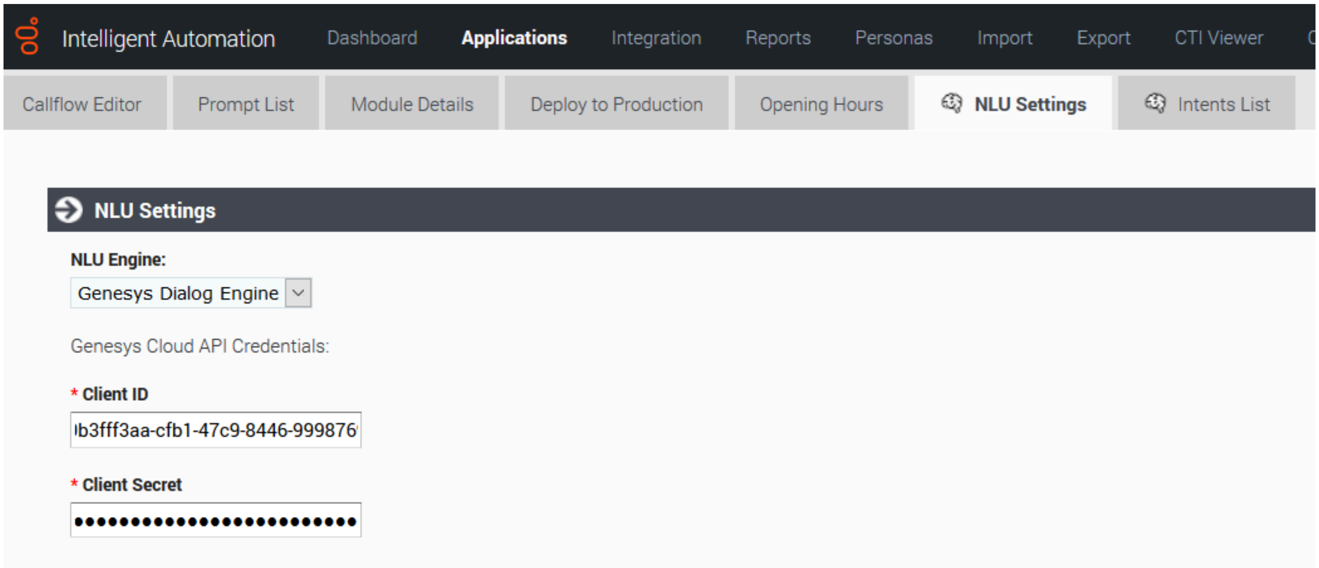
Configure Default Server Settings

If you're using Genesys Dialog Engine, configure the following server settings:

- **DialogEngine.JOPv3.AuthBaseURL** - The URL to the Genesys Cloud Authentication API.
- **DialogEngine.JOPv3.FetchTimeoutMillis** - The length of time (in milliseconds) that Intelligent Automation waits for a response from Dialog Engine before throwing an error.
- **DialogEngine.JOPv3.ClientID** - The Dialog Engine Client Identifier.
- **DialogEngine.JOPv3.ClientSecret** - The Dialog Engine Client Secret password.

> **Important**
> If you do not have access to these settings, contact your Genesys representative.

Configure NLU Settings



NLU Settings

NLU Engine: Genesys Dialog Engine

Genesys Cloud API Credentials:

* Client ID: b3fff3aa-cfb1-47c9-8446-999876

* Client Secret: ••••••••••••••••••••••••

Once you have configured the Default Server Settings, open a **Natural Language Menu** module and click the **NLU Settings** tab.

From the **NLU Engine** menu, select the natural language engine you're using - either Genesys Dialog Engine, Google Dialogflow, or Microsoft Bot Framework.

- If you select Genesys Dialog Engine, the **Client ID** and **Client Secret** password configured on the **Default Server Settings** page will display. Leave the **Use default credentials** box checked to use these credentials or uncheck the box to override them for this particular application.
- If you select Google Dialogflow, you'll need to enter the Google API Service Account JSON. Google provides this JSON when you set up your Google services account.

> **Important**
>
> Ensure that you have the Dialogflow API Admin role configured to your Google services account.
>
> - Ensure that the value of the **DialogEngine.Provider** setting (**Administration** > **Default Server Settings**) is set to `SlotBucket`.

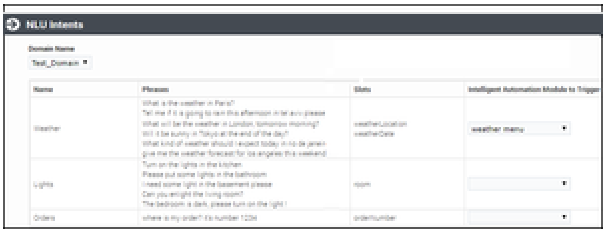
Support for Proxies

To use Google DialogFlow through a proxy, create an environment variable called `GRPC_PROXY_EXP` and set the value in the *host:port* format.

```
GRPC_PROXY_EXP=localhost:31138
```

Map Intents to modules

> **Important**
>
> The Map Intents to modules is supported for Genesys Dialog Engine, Google Dialogflow, and Microsoft Bot Framework.



Domain Name
Test_Domain

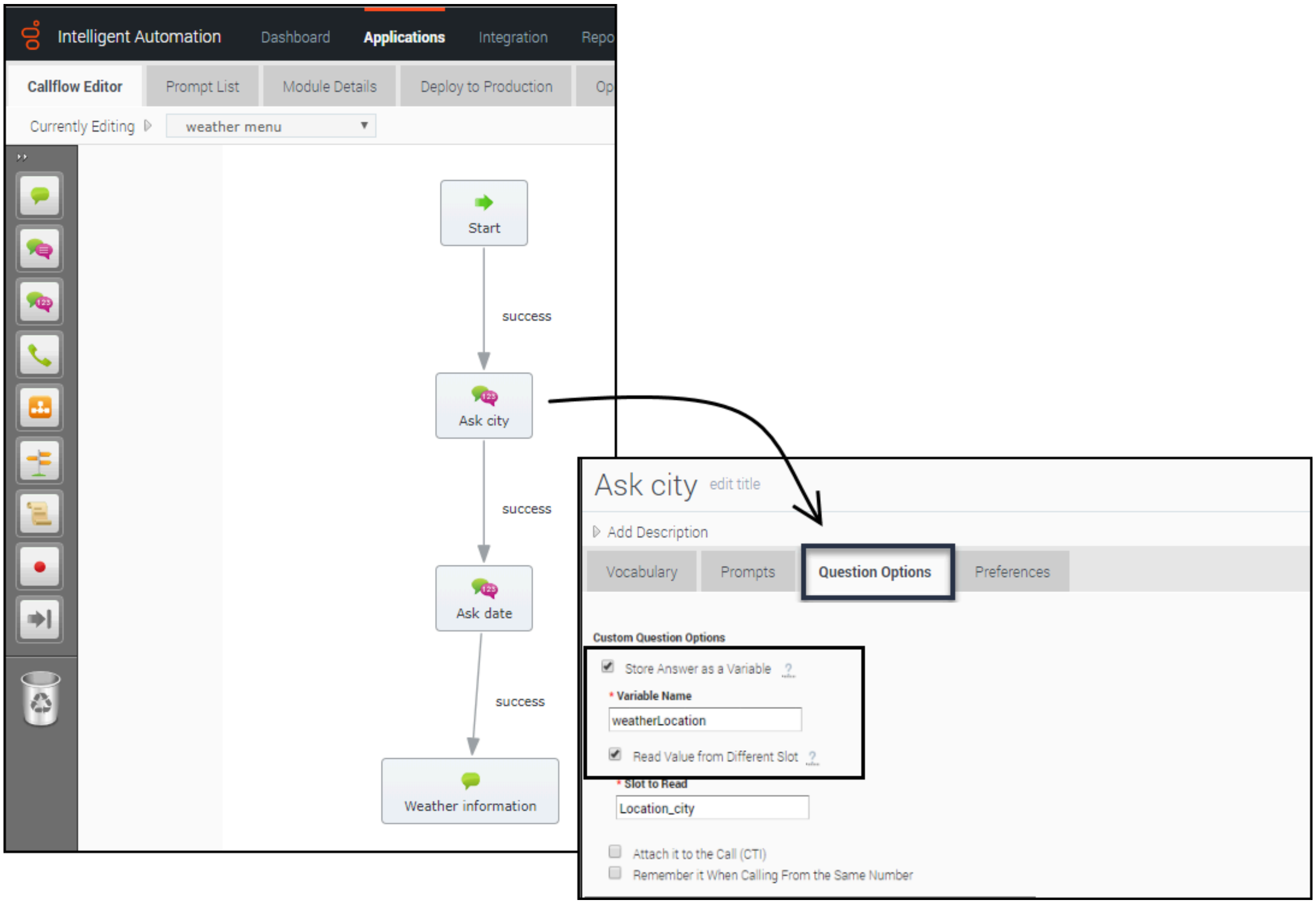
Name	Phrases	Slots	Intelligent Automation Module to Trigger
Weather	What is the weather in Paris? Tell me if it is going to rain this afternoon in tel aviv please What will be the weather in London, tomorrow morning? Will it be sunny in Tokyo at the end of the day? What kind of weather should I expect today in rio de janeir give me the weather forecast for los angeles this weekend	weatherLocation weatherDate	weather menu
Lights	Turn on the lights in the kitchen Please put some lights in the bathroom I need some light in the basement please Can you enlight the living room? The bedroom is dark, please turn on the light !	room	
Orders	where is my order? it's number 1234	orderNumber	

Intelligent Automation reads all Intents, Utterances, and Slots associated with a domain and then displays that information on the **Intents List** page for the **Natural Language Menu** module. This is where you'll map each Intent to a module.

To map an Intent to a module, simply select a module from the **Intelligent Automation Module to Trigger** menu for the Intent. Intelligent Automation will then automatically update the application's callflow accordingly.

> **Important**
>
> If the fulfilment is being performed by the bot, you can select *None* from the **Intelligent Automation Module to Trigger** menu and a generic “success” result will be returned after the intent has been completed.

Map Slots to questions



☑ Store Answer as a Variable ?

* Variable Name: weatherLocation

☑ Read Value from Different Slot ?

* Slot to Read: Location_city

☐ Attach it to the Call (CTI)

☐ Remember it When Calling From the Same Number

In the natural language engine, an Intent contains Slots, which are key pieces of information you need to extract from the end user to process a request. For example, if a user wants the weather forecast, you would need to know two key pieces of information before providing a weather forecast - city and date. These would be considered Slots.

For each Intent, you should have an associated module containing questions that will extract the right information from the customer. For the Weather example, you would have a Weather Questions module, which contains two questions: *For what city* and *For what date*? Both of these questions relate to the weatherLocation and weatherDate Slots.

Once you have created this module in Intelligent Automation, you need to map it to its associated Slots, as follows:

1. Open the module that is linked to Intent.
2. For each Question block in that module for which you would expect to have a slot, open the block and go to **Question Options**.
3. Check the **Store Answer as a Variable** checkbox and enter the Slot name (weatherLocation).

When a chat or voice session reaches one of those Question blocks, it first checks if that slot has been sent to Intelligent Automation from the natural language engine. In that case, Intelligent Automation uses that as the answer to the question. Otherwise, Intelligent Automation asks the question and waits for a response.

Support for Live Agent Hand-off

> **Important**
>
> This feature is supported only for Google Dialogflow.

When the live agent hand-off flag is returned as part of the DialogFlow intent, you can configure IA to follow a special path, usually a hand-off to a live agent.

When a value is provided in the **Path to take When Agent Requested** setting, the call is transferred to a live agent defined in the callflow.

In Dialogflow, the liveAgentHandoff flag should be enabled using a Custom Payload response within the intent.

```
{ “liveAgentHandoff” : true }
```

Support for Maximum Attempts

You can set the maximum number of retries to fill a slot and the maximum number of attempts to identify intents in a chat session.

The **Enable Maximum Attempts Counters** setting allows configuring the number of retry attempts and the action to be performed when the limit is reached.

The **Enforce Maximum Attempts for filling Mandatory Slot** will enable or disable the slot fill failure timeout feature. The number of unsuccessful tries can be configured after which the call is either transferred to a different module or return a result.

For example, if the number of retries is specified as 2, the bot will look if it can fill up any slots based on the user input two times. If the slot is not filled after both attempts, the call flow will be transferred to either a different module or return a special value.

When a value is provided in the **Maximum Attempts to Fill Mandatory Slot Values** option, the bot will try to slot-fill the user input. If the bot cannot perform a slot-fill successfully, the counter is increased. When the counter value exceeds the maximum retry values, Intelligent Automation either transfers the callflow to a different module (configured in the **Exit Module** field) or return a special result.

> **Important**

> When a slot is filled successfully or if the intent is switched, the counter is reset to zero.

When the **Enforce Maximum Attempts for Intent Disambiguation** option is configured, the bot will try to identify an intent from the conversation. When the maximum number of unsuccessful attempts is reached, Intelligent Automation either transfers the callflow to a different module (configured in the **Exit Module** field) or return a special result.

Pass Context Settings

> Important
>
> The Map Intents to modules is supported for Genesys Dialog Engine, Google Dialogflow, and Microsoft Bot Framework.

Intelligent Automation allows you to pass variables to an NLU engine. The NLU engine can use the variables as part of the slot filling capabilities when the **Send Intelligent Automation variables as context to NLU Engine** setting is enabled.

For DialogFlow, the parameter's default value must be mapped to the variable defined in Intelligent Automation. You can configure the Default Value option from the contextual menu for a parameter in DialogFlow and it should be use the following format: *#fish_context.<VariableName>*.

For Dialog Engine, all variables are passed and Dialog Engine will map the variables to slots if the variable names from Intelligent Automation match the slot names in Dialog Engine.

For Microsoft Bot Framework, the context settings are passed in the `slots` section and Microsoft LUIS can use this to extract information. The context is passed only from Intelligent Automation to Microsoft Bot Framework. Context is **not** passed from the Bot Framework to Intelligent Automation.

Bypass Opening Question

You can allow users to bypass the initial question from the bot by passing the utterance as a variable to the NLU. The NLU then skips the initial question and proceeds to the next question. You can use any previously collected data or create and define a variable in the Script block and pass the utterance as a parameter in the **Link** block to the Natural Language Menu. The utterance is then passed to the NLU menu as an input in the callflow.

Support for System entities

> **Important**
>
> This feature is supported only for Google Dialogflow.

DialogFlow supports composite entities and returns them as part of the response. To allow use of these entities in a callflow, Intelligent Automation can use the values as a whole object or split the object into simpler components.

For example, in DialogFlow, we have a slot called *Duration* and the data type is `@sys.duration`, DialogFlow returns the following object as a result:

```
{"amount":10,"unit":"min"}
```

Intelligent Automation will store the object as a slot which can be accessed using a **Script** block.

```
Duration  = {"amount":10,"unit":"min"}
```

Intelligent Automation will also split the object as separate sub-slots which can be referenced in a **Question** block (see section Map Slots to questions). You can access each entity as `<slotName_type> = value`.

```
Duration_amount = 10
Duration_unit = min
```

Sample script

This script will parse the value of the variable, NLSlots and log the slot name, value and confidence.

```
def slots = context.getVariable("NLSlots")

slots.entrySet().each
{
    entry ->
    sSlotName = entry.key
    sSlotValue = entry.value
    sSlotValue.entrySet().each
    {
        valueEntry ->
        value = valueEntry.key
        fConfidence = valueEntry.value
        context.log("Slot info: Name: " + sSlotName + " Value: " + value.getValue() + "
Confidence " + fConfidence)
    }
}
```

Support for Follow-up Intents

> **Important**
>
> This feature is supported only for Google Dialogflow.

Intelligent Automation now supports follow-ups intents from Google Dialogflow. A follow-up intent is an intent that is tied to another earlier intent. Think of it as a child intent of an existing parent intent. When you create a follow-up intents in Dialogflow, Intelligent Automation can read and work with contexts for the intents.

Set up a module with a **Question** block. The Question block stores the intent in the **Prompt Wording** field. Ensure that this field has `var:NLTextResponse` as its value. In the **Question Options** tab, enable the **Store Answer as a variable** option. This holds the values received from DialogFlow. Ensure that the variable name is `utterance`.

Custom Natural Language Menu Microapp

> **Important**
>
> The Custom Natural Language Module is a licensed product in Intelligent Automation. Be sure to acquire and install the updated license prior to beginning the installation process. Directions for installing a license file in Flex License Manager can be found here.

The Custom Natural Language Menu microapp allows you to perform custom integrations to bot frameworks like Google Dialogflow or Genesys Dialog Engine. These custom integrations are configured through scripting with special scripting screens provided to configure bot authentication and processing of utterances.

Differences between the Custom Natural Language Menu microapp and the existing Natural Language Menu microapp

- Custom Natural Language Menu is not restricted to bots supported by Intelligent Automation. You can connect to any bots which have an HTTP REST API. However, this requires scripting knowledge and is more suited to technical users. IA provides example scripts for connecting to Genesys Dialog Engine and Google Dialogflow but non-technical users are still advised to use the existing Natural Language Menu microapp.
- The flow for Custom Natural Language Menu is unlocked. This means that if any bot systems require additional scripting or processing outside of authentication and utterance capture, you can add more blocks to the flow.
- Custom Natural Language Menu has no options to configure paths for intents. Instead, you should manually add the required paths to the flow or to the Link block which has invoked the microapp.

Configuring Dialogflow Prerequisites

Google Dialogflow Agent Credentials

The Google Cloud **Project ID** for the Dialogflow Agent can be found on the **General** tab of the agent settings in your Google Console.

Creating a Google Cloud Service Account

Intelligent Automation connects to Google Cloud Project for your Dialogflow Agent using a Service Account P12 service account key. This key file can be generated once you have created a Service Account within the Google Cloud Project.

- Open the Google Cloud Console for your Dialogflow Agent Project and select **Service Accounts** and select **CREATE SERVICE ACCOUNT**.

- Create a new Service Account.
- Add the **Dialogflow API Admin** role to the new service account.

Generating the P12 Service Account Key

Once the Service Account is created,

- Generate the P12 Service Account Key file by selecting the Service Account from the **Service Account** list and select **ADD KEY**.
- Select the **P12** format and select **Create**.
- Save the key file and also the **Private Key Password** that is displayed.

Configuring the Service Account Email

The CNLM feature also requires the email for the Service Account. This can be found on the Service Accounts list page. This email account is used to configure the feature.

Configuring Custom Natural Language menu microapp

If you are connecting to Dialogflow, you must provide additional information to connect to your Google Apps account. In addition, you have to provide a PK12 certificate and credentials if you are using Intelligent Automation 9.0.100 or later.

- Navigate to **Administration** > **Default Server Settings**.
- Edit the **Resources.AllowedUploadContentTypes** and add "*,application/pkcs-12,application/x-pkcs12*" to the end of the list.

Installing the microapp

- Navigate to **Administration** > **Products**.
- Scroll to the bottom of the page and select **Import a Product**.
- In the **Import Products** page, select **Choose File** and navigate to the **Custom Natural Language Menu.product** file.
- Select **Import Product** to import the microapp.

Importing the template

- Navigate to **Templates** > **Import**.
- Select the **Custom Natural Language Menu Template.zip** file and select Import.

Once the microapp and the template has been installed, you can create a new module using the template.

Configuring the Service Account Key Certificate Alias

Use the `keytool` command to integrate the Custom Natural Language Menu module with Google Dialogflow.

`keytool -list -v -keystore <key file name>` where the key file name is the P12 Service Account key created earlier.

Configuring a new Custom Natural Language Menu Module

Creating a new Custom Natural Language Menu Module

- Navigate to **Applications** and select **Create a new Menu**.
- Select the **Custom Natural Language Menu Template.**
- Enter a name for the new module and select **Create**.
- Select the new module.

The following video (from 01:30 to 02:00) shows how to configure a new CNLM module with Google Dialogflow.

Link to video

Custom Natural Language Menu Dialogflow CX Chatbot Integration

Important

The Custom Natural Language Module is a licensed product in Intelligent Automation. Be sure to acquire and install the updated license prior to beginning the installation process. Directions for installing a license file in Flex License Manager can be found here.

Important

The Custom Natural Language Module supports Dialogflow CX edition for chat features only.

The endpoint URL for Dialogflow CX is https://<region-prefix-if other than US>.dialogflow.googleapis.com/<api-version>/projects/my-project-id/locations/<my region>/agents/my-agent-id/environments/<my environment id>/sessions/<my-session-id>:detectIntent, where:

- <API version> is v3 or v3beta1 (Check the API reference at https://cloud.google.com/dialogflow/cx/docs/reference/rest.)
- <my-project-id> is your project ID in Dialogflow CX Console.
- <my region> is your region in Dialogflow CX Console.
- <my-agent-id> is the agent ID from Dialogflow CX Console.
- <my environment id> is your environment id and also an optional value.
- <my-session-id> is the session ID that is created during runtime in the **Start dialog engine session** script block in **Custom Natural Language Menu**.

Important

You can get the endpoint URL information from the Project Overview screen in Dialogflow console. Select the project from the **Project** dropdown, then select the agent from the **Menu** icon and click **Copy ID**.

You can specify an environment for runtime session calls, like *detectIntent*, *streamingDetectIntent*, *matchIntent*, �and session entity calls. To specify an environment, alter the endpoint URL by inserting environments/environment-id between the agent and sessions path parameters.

For example, the following endpoint uses the 6db409d7-57ac-41d7-83bd-89b8768e2745 environment ID: https://dialogflow.googleapis.com/v3beta1/projects/my-project-id/locations/us/agents/my-agent-id/environments/6db409d7-57ac-41d7-83bd-89b8768e2745/sessions/my-session-id:detectIntent

If an environment is not specified, the default environment is used.

> ## Important
>
> You can get the environment ID form the following path in Dialogflow Console: **Project > Agent > Manage > Environments**. See versions and environments in Google Dialogflow CX documentation.

You can also include these values as module parameters and build the URL to access the endpoint. You can then include a script that reads the parameters and constructs the Base URL:

```
def slanguageCode = context.getVariable("LanguageCode");
def sAPIVersion = context.getVariable("GoogleCXAPIVersion");
def sGoogleCXProjectID = context.getVariable("GoogleCXProjectID");
def sLocations = context.getVariable("GoogleCXRegion");
def sAgentID =context.getVariable("GoogleCXAgentID");
def sEnvironmentID = context.getVariable("GoogleCXEnvironmentID");

def sBASE_API_URL = "https://europe-west1-dialogflow.googleapis.com/" + sAPIVersion +
"/projects/" + sGoogleCXProjectID + "/locations/" + sLocations + "/agents/" + sAgentID +
"/environments/" + sEnvironmentID+ "/sessions/" + sSessionID +
":detectIntent?access_token="+sToken;
```

Sample scripts

These scripts can be used as a starting point to build your custom scripts:

- Trigger the default welcome for the default intent
- Perform natural language interpretation

Natural Language Audio Streaming Menu

Important

The Natural Language Audio Streaming feature is available only for Early Adaptor Customers and availability is under Product Manager Control. Contact your Account Executive for further information on using this feature.

The Natural Language Audio Streaming Menu microapp allows direct streaming of audio to Bot Engines using Nexus, GWS, and Google Dialogflow. Currently, Google Dialogflow Essentials is supported.

You can import the Natural Language Audio Streaming as a new product into Intelligent Automation and use it in your callflows.

Prerequisites for Audio Streaming

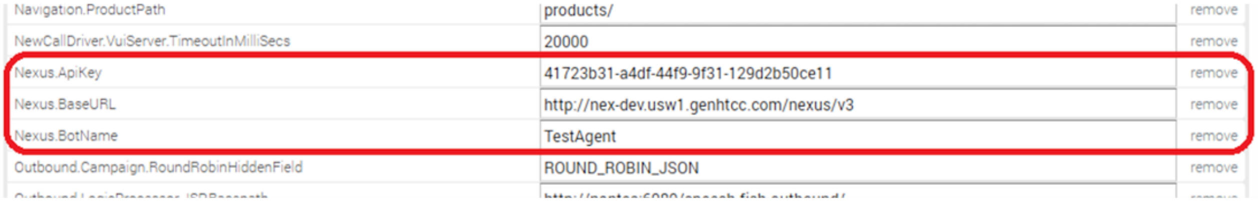
You must have the following prerequisites available and configured before you can use the app:

- Genesys Intelligent Automation with Conversation AI Orchestration
- Genesys Digital (Nexus)
- Genesys Web Services and Applications 9
- Genesys Voice Portal
    - The latest MCP Linux release (Windows is not supported)
    - Voice Self Service Applications
    - vXML Interpreter
- Google Dialogflow

Configuring the Natural Language Audio Streaming Menu Microapp

To enable this feature, configure the following Server Settings:

- **Nexus Api Key** - Enter the API Key to access the Nexus API.
- **Nexus BaseURL** - Enter the URL of the Nexus server.

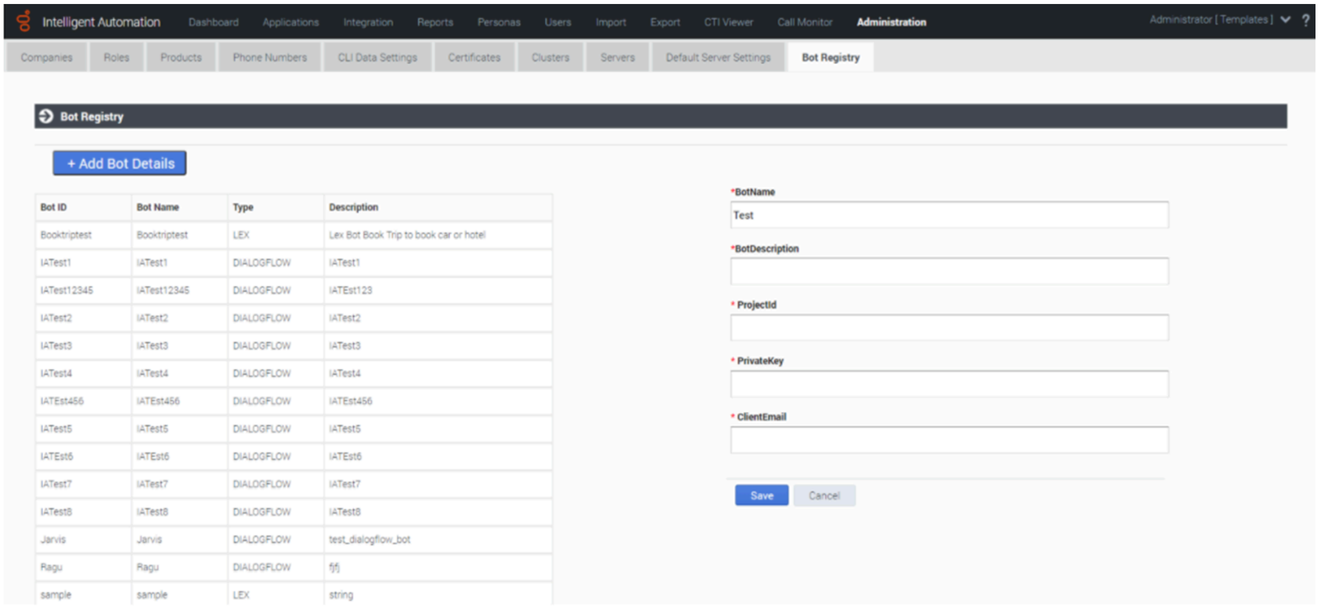


Navigation.ProductPath	products/	remove
NewCallDriver.VuiServer.TimeoutInMilliSecs	20000	remove
Nexus.ApiKey	41723b31-a4df-44f9-9f31-129d2b50ce11	remove
Nexus.BaseURL	http://nex-dev.usw1.genhtcc.com/nexus/v3	remove
Nexus.BotName	TestAgent	remove
Outbound.Campaign.RoundRobinHiddenField	ROUND_ROBIN_JSON	remove

To allow GVP to invoke Nexus for audio streaming, Intelligent Automation uses the following VXML properties:

- `<property name="com.genesyslab.asrengine" value="nexus"/>`
- `<property name="com.genesyslab.asr.botName" value="<bot-name>"/>` - This property can be configured in the bot or intent settings.
- `<property name="com.genesyslab.asr.sessionid" gvp:expr="sessionid"/>`
- `<property name="com.genesyslab.asr.x-api-key" gvp:expr="<apikey>"/>`
- `<property name="com.genesyslab.asr.contexts" gvp:expr="var1"/>`
- `<property name="com.genesyslab.asr.contextsPolicy" value ="<contextPolicy>"/>`

After the options are configured, a new **Bot Registry** tab is available. the Bot Registry tab lists all bots that are available in the Nexus server.



Bot ID	Bot Name	Type	Description
Booktriptest	Booktriptest	LEX	Lex Bot Book Trip to book car or hotel
IATest1	IATest1	DIALOGFLOW	IATest1
IATest12345	IATest12345	DIALOGFLOW	IATEst123
IATest2	IATest2	DIALOGFLOW	IATest2
IATest3	IATest3	DIALOGFLOW	IATest3
IATest4	IATest4	DIALOGFLOW	IATest4
IATEst456	IATEst456	DIALOGFLOW	IATEst456
IATest5	IATest5	DIALOGFLOW	IATest5
IATEst6	IATEst6	DIALOGFLOW	IATEst6
IATest7	IATest7	DIALOGFLOW	IATest7
IATest8	IATest8	DIALOGFLOW	IATest8
Jarvis	Jarvis	DIALOGFLOW	test_dialogflow_bot
Ragu	Ragu	DIALOGFLOW	fjfj
sample	sample	LEX	string

*BotName: Test
*BotDescription:
* ProjectId:
* PrivateKey:
* ClientEmail:

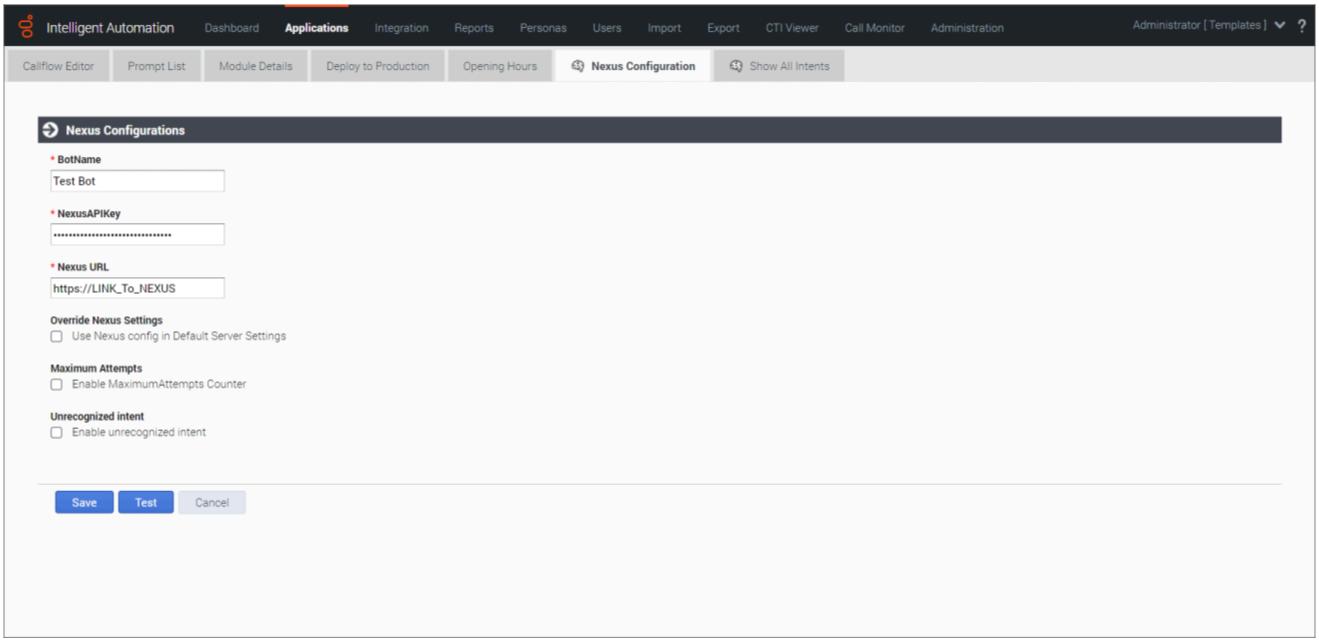
You can also add a new bot using the **+ Add Bot Details** option.

Using the Natural Language Audio Streaming Menu Microapp

- Navigate to **Applications** and select **Create a new Menu**.
- Select the **Natural Language Audio Streaming Menu Template** option.
- Enter a name for the new module and select **Create**.
- Select the new module.

The audio streaming module has two additional tabs, **Nexus Configuration** and **Show All Intents**.

The **Nexus Configuration** tab displays the current Nexus configuration setting for the call flow.



Nexus Configurations

* BotName
Test Bot

* NexusAPIKey
••••••••••••••••••••••••••••••

* Nexus URL
https://LINK_To_NEXUS

Override Nexus Settings
□ Use Nexus config in Default Server Settings

Maximum Attempts
□ Enable MaximumAttempts Counter

Unrecognized intent
□ Enable unrecognized intent

If your callflow requires any additional configuration, you can override the settings configured in the Default Server Settings:

- **BotName** - Enter the name for the Bot.
- **NexusAPIKey** - Enter the API Key to access the Nexus API.
- **Nexus URL** - Enter the URL of the Nexus server.

When enabled, the **Use Nexus config in Default Server Settings** setting will use the information configured in the Default Server Settings options.

You can also specify the maximum attempts values by enabling the **Enable MaximumAttempts Counter** field.

To allow unrecognized intents to be handled, enable the **Enable unrecognized intent** option. You can configure the module that will be triggered when an intent is unrecognized.

The **Context Setting** option supports passing audio context to Nexus as part of Audio Streaming application from GIA 9.0.112.12 onwards. The context values are specified by a variable defined in the **Enter Variable Name** field. The context policy selected in **Select Context Policy** field will pass on the corresponding context policy on how to use the context to Dialogflow.

Currently the following policies are supported:

- `merge_soft` – This policy will merge existing parameters (returned from Dialogflow and preserved by Nexus) with the request data. If there is a conflict, the existing context is retained.
- `merge_hard` – This policy merges the existing parameters (returned from Dialogflow and preserved by Nexus) with the request data. If there is a conflict, the values from the request (the ones provided by Nexus client) will take precedence.

The contexts are passed through the **asr.contexts** property and the context policies using the **asr.contextsPolicy** property in the VXML.

The **Show All Intents** tab lists all intents available for the bot.

> Important
>
> Intents cannot be configured from within Intelligent Automation. The intents are fetched from Nexus and available for use within GIA.

System Variables

```
<assign name="NLIntent" expr="nexus_form_Response.data.intent" />
<assign name="NLSlots" expr="JSON.stringify(nexus_form_Response.data.slots)" />
<assign name="NLTextResponse" expr="nexus_form_Response.data.message" />
<assign name="NLInputText" expr="nexus_form_Response.data.inputTranscript" />
```

MicroApps

MicroApps are pre-built collections of callflow templates, configuration screens, and scripts built to industry best practices. They allow for rapid deployment of commonly-required functions within a self-service system and can be used across all channels supported by Intelligent Automation.

All available MicroApps are listed in the table below.

Name	Description
Account Barring	Allows a caller to bar and unbar their account.
Account Selector	Allows the caller to select an account from a list of accounts fetched from the back-end system.
Add-Ons Selector	Plays the list of available add-ons configured, and then provides an option to order one.
Address Capture UK	Captures an address based on a UK postcode and first line of address, verified using third-party address software from Experian QAS www.qas.co.uk.
Address Selector	Dynamically builds and presents a list of the caller's addresses, and then prompts the caller to choose the address for mail or parcel delivery.
Airline Flight Status Checker	Checks the status of a flight based on either the departure date with flight number or on the departure date with departure and arrival airport.
Automatic Query	A version of a query that does not prompt the caller for input.
Balance Enquiry	Performs a real-time lookup to a back-end system and plays the balance to the caller, with an offer to repeat the information.
Bill Copy Request	Calls a back-end system to request a copy of the caller’s most recent bill, which can be posted or emailed to the customer at a pre-registered address.
Bill Information	Calls a back-end system to retrieve bill information, including multiple balances. It plays this information to the caller, with an offer to repeat the information.
Billed and Unbilled Balance	Performs a real-time lookup to a back-end system and plays the billed and unbilled balance to the caller, with an offer to repeat the information.
Call Divert	Provides a range of functionality for call diverting. For example, divert to voicemail, another number, and when unavailable. It also allows the caller to cancel all call diverts.
Call Waiting Activation	Allows the caller to activate or de-activate call waiting on their account.

Name	Description
Caller ID Activation	Allows the caller to activate or de-activate caller ID on their account.
Cancel Contract	Allows the caller to cancel their contract. Calls a back-end system to request the cancellation.
Continuous Card Payment	Collects the debit or credit card details required for setting up continuous payment and passes these details on to the back-end system for continuous payment processing.
Custom Natural Language Menu	The Custom Natural Language Menu allows older versions of Intelligent Automation to support bot capabilities. Important The Natural Language Menu App and Custom Natural Language Menu are shipped separately and only allowed in conjunction with the purchase of either the bot bundle/orchestration either Voicebot or Chatbot bundle. Get in touch with your AE or Sales.
Date Selector	Dynamically builds and presents a list of dates to the caller, and then prompts the caller to choose a date for delivery.
Direct Debit Request	Captures the bank sort code and account number required to set up a new Direct Debit instruction on the caller's account.
Dynamic Grammar	Defines a dynamic grammar that is populated from a web service.
Dynamic Menu	Allows you to build a dynamic menu.
Easy Opening Hours	An easy way to configure opening hours from one place, with extended functionality for special dates.
Flight Status Checker	Checks the status of a flight based on either the departure date with flight number or on the departure date with departure and arrival airport.
Foreign Exchange	Dynamically builds and presents a list of currencies to the caller and then prompts the caller to specify a currency and dollar amount for the order.
Genesys Smart Transfer	Presents call waiting options to the caller, depending on the results returned by Genesys.
Handset Unlock Request	Calls a back-end system to request a handset unlock for the caller. Can ask the caller for information such as IMEI before making the request.
Hold	For outbound dialing - enables the receiving party to place the call on hold. The call resumes once the receiving party presses a key on the keypad.
Identification	Identifies the caller's Caller Line Identification (CLI), account number, or other unique identifier. Allows progressive levels of identification to be used within the same callflow.

Name	Description
Incident Alert	Allows easy management of incidents. Active incidents are played to the caller, and then the caller is given the option to subscribe to the incident.
IT Help Desk	Define a multi-synonym question with disambiguation. When the caller answers the question, the call moves to the next module.
Itinerary	Calls a back-end system to retrieve a flight, hotel, limo, rental car, or rail itinerary for the caller. Can be used in conjunction with the Itinerary Selector app.
Itinerary Selector	Dynamically presents flight, hotel, limo, rental car, or rail itineraries to the caller, depending on what has been booked. Calls a back-end system to retrieve all itineraries for the caller's booking. Used in conjunction with the Itinerary app.
International Direct Debit Request	Captures an international bank account number (IBAN), and then calls a web service to process the Direct Debit request.
Loan Request	Allows the caller to request a loan.
Local Identification and Verification	Identifies a caller against the users configured within the product.
Locate and Connect	Used to locate a local branch, ATM, or office. Plays a list of locales and branches configured in the application, and then prompts the caller to select from the list. This application requires speech recognition.
Locate and Connect Postcode	An enhanced version of the Locate and Connect MicroApp. It uses the full UK postcodes to locate branches and partial area postcodes to establish relationships between them. It also uses latitude/longitude calculations and weighting of local dialing codes to find the nearest branches. Customization for other postcode systems is available.
Marketing Promotions	Uses the dialed number (DNIS) and other available customer information to dynamically select and present a promotional offer to the caller, based on eligibility criteria configured in the application. The caller is given the option to subscribe to the offer, which can be updated to his/her account using a back-end system call. Includes the ability to transfer the call with its context to a specific agent group, based on selection.
Module Transfer Helper	This is the base flow for the Module Transfer Helper product. Do not modify the callflow, it will be updated any time its Settings page is saved.
Money Transfer	Allows the customer to transfer money from a selected *From* account to a previously used bill payment destination. The *From* account uses Account Selector. The list of payment destinations

Name	Description
	is constructed dynamically. Can handle scenarios involving insufficient funds.
Natural Language Menu	Allows you to connect to a natural language engine for any chat interaction. Important The Natural Language Menu App and Custom Natural Language Menu are shipped separately and only allowed in conjunction with the purchase of either the bot bundle/orchestration either Voicebot or Chatbot bundle. Get in touch with your AE or Sales.
Offer Direct Debit Setup	Captures the bank sort code and account number required to set up a new Direct Debit instruction on the customer's account.
Operator	Automated attendant for connecting callers to contacts and departments.
Order Add-Ons	Allows callers to order add-ons.
Order Phone Services	Presents a list of phone services for the caller to choose from (fully configured within the application) and calls a back-end system to request the order. Handles functionality such as eligibility, insufficient funds, and duplicate orders.
PAC Port In	Guides the caller through the process of porting in a PAC number from a previous provider. Captures the information and calls a back-end system to make the request. Can also offer the caller the opportunity to specify the date they wish the port in to occur.
PAC Request	Calls a back-end system to request and present a PAC for the caller. Usually used in conjunction with the SMS product to send the PAC to the caller.
Payment Capture	Secure payment functionality offers PCI Compliant payments. These applications can be used as part of a self-service or agent-assisted interaction and are compliant with the strict requirements of the Payment Card Industry Security Standards Council (PCISSC). Important • The Payment Capture App is available only as a separate package. It is provided to customers on request only. Get in touch with your AE or Sales. • The existing Payment Capture App can be used when upgrading Intelligent Automation.
Pay by Bank Account	Captures bank account details, and then processes

Name	Description
	a payment using those details.
Payment by Registered Card	A two-step process: 1. Calls a back-end system to retrieve any cards registered against the caller's account. The caller then chooses the card they wish to use (by entering the last four digits), the amount they wish to pay, and then the three-digit security code for the card. 2. Sends a back-end request to process the payment. Offers "friendly" functionality such a 'card expiring soon' warning.
PIN Change	Allows callers to set a new Personal Identification Number (PIN) by first requiring the old PIN to be entered. The caller must enter the old PIN and then enter the new PIN twice, as part of the validation process. A web service is used to update the PIN in the back-end system.
PIN Request	Requests the callers PIN from a back-end system. Can be presented to the caller or used for verification within the application.
Plan Details	Calls a back-end system to retrieve call plan information, including monthly charges and the plan name. It then plays this information back to the customer, with an offer to repeat the information.
Postcode Outage	Allows the caller to enter a numeric postcode to find out if there is an outage in their area.
Postal Flat Rates	Presents the caller with a series of questions to determine the fixed postal rate.
Postal Rates Calculator	Requests information from the caller about the weight, dimensions, and destination of the item to be posted, and the calls a web service to calculate the postal rate.
Price Plan Change	Presents a list of price plans for the caller to choose from (fully configured within the application) and calls a back-end system to request the change. Handles functionality such as eligibility, insufficient funds, and duplicate orders.
Product Order	Allows the caller to order products on account. Dynamically builds and presents the available inventory, and then prompts the caller to choose a product and quantity. If the specified quantity doesn't exceed the available inventory, the order is placed.
Prompt Recorder	Quickly and easily produces new prompt WAVs for use in IVRs. Call a direct dial number to record your prompts, and then downloads it via the Intelligent Automation application.

Name	Description
PUK Request	Calls a back-end system to request and present a PUK for the caller. Usually used in conjunction with the SMS product to send the PUK code to the caller.
Query List	Returns a list of information from a back-end, and then presents the information to the caller.
Query	Uses input from the caller, such as application reference number, to collect the status from a back-end system and then play the status to the caller. Includes variations, with an option to provide an SMS status confirmation to the caller.
Query Account Add-Ons	Calls a back-end system to retrieve add-on information currently associated with the caller's account. Information presented to the caller is driven by data fully configured within the application. This includes an add-on description.
Query Account Price Plan	Calls a back-end system to retrieve price plan information currently associated with the caller's account. Information presented to the caller is driven by data fully configured within the application. This includes the price plan description.
Questionnaire Builder	Powerful and easy-to-use application suitable for customer and employee surveys, as well as other questionnaires or checklists (for example, insurance renewals). Supports yes/no, multiple choice, and numeric range responses, with options to record verbatim comments, re-order questions, and configurable conditional question sequence logic.
Random Questionnaire Builder	Includes the Questionnaire Builder functionality, with added question sets. The caller hears a randomly-selected question set at the time of the call.
Register Card	Offers the caller the option to register a credit card or debit card against their account for ease of use in the future. Calls a back-end system to store the card information. Usually used after a successful Payment Capture call and usually required for customers wishing to use the Payment by Registered Card product.
	Presents basic remaining allowance information to the caller. This includes minutes, texts, and data. Can also present fair use information, if necessary.
Remaining Allowances v2	An enhanced version of the original Remaining Allowances product. Supports a fully configurable list of allowances (numeric, decimal, currency, and simple on/off). Prompting is customizable and can play back conditional information.
Responder	Allow emergency personnel to respond to pager alerts to acknowledge incidents.

Name	Description
Roaming Activation	Allows the caller to activate or de-activate roaming on their account.
Semafone Payment Capture	Includes all Payment Capture features, with an added pre-built integration to Semafone (www.semafone.com, which enables DTMF tone masking.
SIM Card Activation	Allows the caller to activate a SIM card currently pending against their account.
SMS	Dynamically constructs an SMS message using information available from the call and then submits the message for delivery to an SMS gateway. It automatically prompts the caller to enter a mobile number if the caller’s CLI is not recognized as a mobile or if is withheld.
Statement Request	Requests the account's statement by fax.
Transaction List	Plays a list of recent transactions for a current account, savings account, or credit card statement retrieved from a back-end system. The account from which to retrieve transactions is determined by Account Selector.
Top Up By Voucher	Allows the caller the option to top up an account by entering a pre-purchased voucher number. Handles scenarios such as account locked, expired, already activated, and not found.
Top Shelf Activation	Activates top shelf content on the callers account. Usually used in conjunction with the Payment Capture product in order to confirm age.
Two Factor Authentication	Identification with additional verification by registered phone number, email address or authentication app.
Unbilled Amount	Performs a real-time lookup to a back-end system and plays the unbilled amount to the caller, with an offer to repeat the information.
Voicemail Activation	Allows the caller to activate or de-activate voicemail on their account.
Yes No Selector	Allows the caller to provide a yes/no response to a question.

Two-factor Authentication

The two-factor authentication MicroApp adds an extra layer of security to a self-service application.

Use case: After customers provide an account number as a form of identification, they're asked to enter a code that was delivered to them via email or SMS, or one they accessed via an external authentication application, such as Google Authenticator.

Intelligent Automation supports two types of two-factor authentication:

- Internal authentication - Intelligent Automation generates a code internally and delivers the code to the user's verified email address or phone number (SMS).
- External authentication - Customers use their own systems to authenticate a user.

Internal authentication

With internal authentication, Intelligent Automation generates a code internally and delivers the code to the user's verified email address or phone number (SMS).

The process is as follows:

1. If the user has more than one verified contact method, Intelligent Automation prompts the user to select an authentication method. For example, Press 1 to receive the code via email. Press 2 to receive the code via SMS. If the user has only one verified contact method, Intelligent Automation skips to Step 3 in this process.
2. The user enters a response.
3. Intelligent Automation generates the code and delivers it to the user's preferred contact method (SMS, for example).
4. Intelligent Automation prompts the user to enter the code.
5. When the user enters the code, Intelligent Automation checks the code and its timestamp.
6. If the code and timestamp are valid, Intelligent Automation grants access. If the code or timestamp is invalid, Intelligent Automation generates and send another code and prompts the user to enter the new code. It does this until the **Maximum Attempts at Sending an Authentication Code** value is reached.

Configuring contact methods

Email

Configure the following server settings to enable Email communication:

- Email.SMTP.Host – The hostname of the customers SMTP server.
- Email.SMTP.Port – The port number on which the SMTP host is running.

SMS

Configure the following server settings to enable SMS communication:

- SMS.Method – The type of request required by the customers SMS Gateway e.g. POST
- SMS.URL – The URL for the customer’s SMS gateway
- SMS.RequestHeaders – any headers which need to be sent in the format “HeaderName1:HeaderValue1”,”HeaderName2:HeaderValue2” etc. Can be blank
- SMS.RequestBody – anything which needs to be included in the request body. This is specific to each customer’s own setup and can be blank.
- SMS.Timeout – Time in milliseconds to wait before declaring an error in contacting the SMS gateway
- SMS.PlusSymbolBeforeRecipientNumber – true or false. Determines whether we need to add a “+” symbol to the recipient number before attempting to send

External authentication

With external authentication, Intelligent Automation hands the code generation and verification process over to an external authentication application, such as Google Authenticator. In this case, Intelligent Automation skips the **send code** web service.

The process is as follows:

1. If the user has more than one verified contact method, Intelligent Automation prompts the user to select an authentication method. For example, Press 1 to receive the code via email. Press 2 to receive the code via Google Authenticator. If the user has only one verified contact method (in this example Google Authenticator), Intelligent Automation skips to Step 3 in this process.
2. The user enters a response.
3. The user retrieves the code from the external application and enters it into the Intelligent Automation application
4. Intelligent Automation checks with the external application that the code is valid.
5. The external application returns a the validation result (for example **success**).
6. If the result is successful (i.e. the code is valid), Intelligent Automation grants access. If the result is unsuccessful (i.e. the code is invalid), Intelligent Automation prompts the user to enter another code, until the **Maximum Attempts at Sending an Authentication Code** value is reached.

Web service details

For both internal and external authentication, web service calls are made to either verify a contact method or to send and verify an authentication code. This section describes the available web services. **Note**: You can write your own wrapper web service or use Intelligent Automation Integration Hub.

Check Verified Contact Methods Web Service

This web service applies to both internal and external authentication.

The web service should return XML in the following format:

```
<checkVerifiedContactMethods>
                    <status>success</status>
                    <methods>
                           <method name="email" value="youremail@genesys.com"/>
                           <method name="authenticator"/>
                           <method name="SMS"/>
                    </methods>
</checkVerifiedContactMethods>
```

- **Status** - The values **success** and **not found** indicate whether the information was found.
- **Methods** - Describes which valid contact methods are available to the customer. The options are **email**, **SMS** or **authenticator**. You can assign an optional value to either **email** or **SMS** and those variables will be configured with that value on the **Settings** page. Using the example above, if **Variable Name for Customer Email Address** is set to **CustomerEmail** on the Settings page, then Intelligent Automation sets a variable called **CustomerEmail** to **youremail@genesys.com**. If you don’t assign a value, Intelligent Automation assumes the variable is already set.

Send Authentication Code Web Service

This web service applies to external authentication only.

Use the request parameter **TwoFactorAuthSelectedMethod**. This parameter will tell the web service which method the customer selected (Email, SMS, or Authenticator).
Expected XML response:

```
<sendAuthenticationCode>
                <status>success</status>
</sendAuthenticationCode>
```

Intelligent Automation only checks for a **success** or **incorrect code** status, but the <status> parameter can contain any valid path, such as **error** to indicate that an attempt to send an authentication code has failed.

Code Authentication Web Service

This web service applies to external authentication only.

Use the request parameter **authCode_nbest1**. This parameter contains the code that the customer entered. The web service checks this against the code sent out previously.
Expected XML response:

```
<codeAuthentication>
                <status>success</status>
</codeAuthentication>
```

Intelligent Automation only checks for a **success** or **incorrect code** status, but the <status> parameter can contain any valid path, such as **error** to indicate that an attempt to send an authentication code has failed.

Enabling Two-factor Authentication

Two-factor Authentication settings are enabled on the **Applications > Two Factor Authentication Settings** page in the Intelligent Automation user interface.

Setting	Description
Contact Methods and Behavior	
Variable Name for Customer Email Address	The user's verified email address.
Variable Name for Customer Phone Number	The user's verified phone number.
Behaviour if No Verified Contact Method Available	The name of a valid path in the callflow (for example, **agent**). This can be either a default path or a path coming from the Link block that calls the MicroApp.
Maximum Attempts at Sending an Authentication Code	The maximum number of times that Intelligent Automation generates and delivers a new code.
Behaviour After Exceeding Maximum Code Send Attempts	The name of a valid path in the callflow (for example, **agent**). This can be either a default path or a path coming from the Link block that calls the MicroApp.
Maximum Attempts at Verifying an Authentication Code	The maximum number of times that Intelligent Automation will process a verification request for a single authentication code.
Behaviour After Exceeding Maximum Code Verification Attempts	The name of a valid path in the callflow (for example, **agent**). This can be either a default path or a path coming from the Link block that calls the MicroApp.
Skip Options	If a user has passed authentication and the application returns to the module at any point, the application will not attempt to go through the verification process again.
Code Generation Options	
Select an option for the code generation	Options are Internal and External.
External Code Generation Options	
Prompt Wording	Enter the name of the external authenticator application (for example, Google Authenticator).
Web Service details	Provide the Web Service URL for test calls and production calls. For external authentication, the web services are as follows: • Check Verified Contact Methods Web Service • Send Authentication Code Web Service

Setting	Description
	- Code Authentication Web Service Refer to the Web service details section above for more information.
Web Service Timeout (milliseconds)	When this value is reached, the code is no longer valid. When setting this value, consider the time it takes for the user to retrieve the code and the time it takes to enter it.
Internal Code Generation Options	
Number of digits for the Generated Code	Intelligent Automation automatically updates the callflow according to the value specified here. Important For external authentication, Intelligent Automation isn't privy to this information, so you need to manually change the grammar settings for **Enter Code** in the same way you would for other MicroApps.
Web Service details	Provide the Web Service URL for test calls and production calls for the following web service: - Check Verified Contact Methods Web Service Refer to the Web service details section above for more information.
Internal Code Expiration Time (minutes)	When this value is reached, the code is no longer valid. When setting this value, consider the time it takes for the user to receive the code and the time it takes to enter it.
- Email Address 'From' - Email Subject - SMS Number 'From'	Sender information that appears in the email or text message to the use

Smart Transfer

When a caller requests to speak to an agent, the Smart Transfer MicroApp allows you to either place the caller in the queue or offer a callback.

How it works

- If the estimated wait time is considered short, Intelligent Automation places the call in the queue.
- For a moderate wait time, Intelligent Automation plays the estimated wait time (in minutes) to the caller and then offers the callback option.
- For a long wait time, Intelligent Automation offers a callback option but doesn't play the estimated wait time.
- If the caller accepts the callback offer, Intelligent Automation offers an available time slot for the callback.
- If the caller accepts the time slot, the caller can specify the callback number - either the number they called in with or a different number.
- If the caller doesn’t accept the callback offer or the available time slot, Intelligent Automation places the call in the queue.

Smart Transfer set up and configuration

Add the Genesys Smart Transfer module

To add the Genesys Smart Transfer module:

1. Navigate to **Applications -> Utility Modules -> Create a new utility module**.
2. Select **Genesys Smart Transfer Template (en-gb**).

Configure Smart Transfer settings

This section describes all available Smart Transfer settings.

Main Settings

Option	Description
Account Variable Name	The name of a mandatory variable (for example, AccountType). Refer to the Declaring variables section below for more information.
Automatic Transfer Threshold	This value, in seconds, defines a *short* wait time. If the expected wait time is equal to or lower than the value specified here, Intelligent Automation automatically places the call in the queue. It does not play back the estimated wait time or offer a callback option. The default value is 60.
Upper Wait Time Play Back Threshold	This value, in seconds, defines a *long* wait time. If the expected wait time is equal to or greater than the value specified here, Intelligent Automation offers a callback option but does not play the estimated wait time. The default value is 600. If the expected wait time falls between the **Automatic Transfer Threshold** value and the **Upper Wait Time Play Back Threshold** value, Intelligent Automation plays the estimated wait time to the caller and offers the callback option.

Web Service Details Settings

To fetch the estimated wait time and available time slots, and to schedule the callback appointment, Intelligent Automation sends API requests to Genesys Mobile Services (GMS) (the platform Callback is built on). It relies on the fish-services Web application to format and translate requests and responses transmitted between Intelligent Automation and GMS.

Use the Web Services Details Settings section to specify the URLs that call on the fish-services Web application and specify a value for the **Web Service Timeout** option, which defines the amount of time that Intelligent Automation waits for a response from the web service. If the response doesn't return before this value is reached, Intelligent Automation treats the request as being in an error state.

The table below lists the following information:

- The name of each web service
- The URL used to call on to the fish-services Web application
- The API that the fish-services Web application calls on to retrieve the web service information
- The URL format for the API call

Web service name	URL to call fish-services Web application	API used by fish-services	URL format for API call
EstimatedWaitTime	http://<host>:<port>/fish-services/callback/EstimatedWaitTime.jsp	Stat Server API	http://<hostname>:<port>/genesys/1/internal_statistic

Web service name	URL to call fish-services Web application	API used by fish-services	URL format for API call
CallbackAvailabilityTimes	http://<host>:<port>/fish-services/callback/CallbackAvailabilityTimes.jsp	Callback Services API	http://<hostname>:<port>/genesys/1/service/callback/{callback-execution-name}/availability **Note:** Replace **{callback-execution-name}** with the name of the web service.
RequestCallback	http://<host>:<port>/fish-services/callback/RequestCallback.jsp	Callback Services API	http://<hostname>:<port>/genesys/1/service/callback/{callback-execution-name} **Note:** Replace **{callback-execution-name}** with the name of the web service.

Declaring variables

Within the fish-services Web application, you can use the **genesys_callback.properties** file to specify information such as the URLs and timeout values.

Within the **genesys_callback.properties** file located in fish-services, each web service call has a **ParametersPassThrough** field which you can use to specify the names of variables you need to send through from Intelligent Automation to GMS (including the one specified in the **Account Variable Name** field on the Smart Transfer Settings page in Intelligent Automation). All variables listed here must be in comma separated format.

Identification

The Identification MicroApp allows callers to identify themselves using their Caller Line Identification (CLI), account number, or other unique identifier. The MicroApp also allows progressive levels of identification to be used within the same callflow.

When a caller dials in, the Identification MicroApp will help the caller identify themselves and proceed further through the callflow.

The Identification module comes with three different templates that can be used in callflows:

- Identification Single Function - Identify using a single question.
- Identification Double Question - Identify using two questions.
- Identification Double Question with Answer Correction - A variant of the Double Question with an option to change the answers.

All three templates share the same logic but their workflows may differ.

Single Question Identification Module

The **Prerequisite Identification Module** option allows you to set another module as a prerequisite before this module is triggered.

The **Fall-back Identification Module** option allows you to set another module as a fallback module. If the user fails the current module, the fallback module is triggered.

The **Maximum Attempts at This Identification Level** option allows you to define how many times that IA will try to identify a caller. This indicated the number of times the identification web services is called, not the number of times that a caller tries to identify themselves.

The **Behaviour for the 'Don't Know" Option** allows triggering a specific path for scenarios such as when the caller responds "I don't know." to the question, "What is your account number?" The option can either trigger a specific path for such queries or trigger the fallback identification module.

The **Behaviour when Maximum Attempts is Reached** option allows triggering a specific path for scenarios when the value specified in the **Maximum Attempts at This Identification Level** option is reached. The option can either trigger a specific path for such queries or trigger the fallback identification module.

The **Skip if Already Completed** option allows an identified caller to skip the Identification module. In scenarios where there are repeat transactions like multiple payments, this option when selected would allow the caller to bypass the Identification module for the second and subsequent payments in the same call.

The **Skip 'Ask ID question" step** option when selected would store the answers provided by the customer. The next time the caller dials in, IA checks if the answer is available and allows the caller to skip the identification module.

The **Web Service Details** section defines the web service URLs to be used for validation. You can define different URLs for test and production environments.

Double Question Identification Module

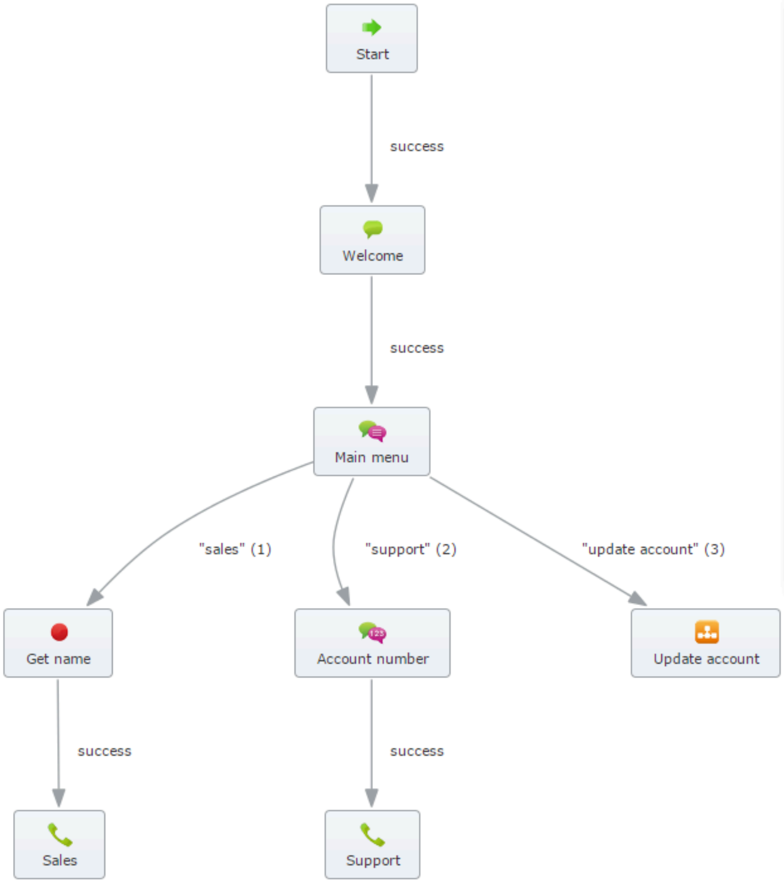
For the Double Question template, the flow is similar except that there are an additional identification question. The logic is similar to the Single Question template.

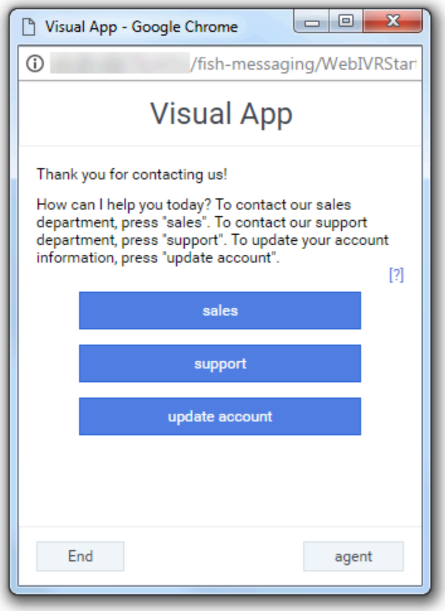
Double Question with Answer Correction Identification Module

The Double Question with Answer Correction template follows the same logic as a Double Question template but includes an additional **Skip 'Ask ID question" step** option. The callflow can allow a user to skip one or both questions as part of the identification flow.

WebIVR Applications

You can use Genesys Intelligent Automation to create WebIVR versions of your voice applications. WebIVR applications are visual and web-based, allowing you to use the same routing logic as your voice application in a web or mobile setting.





Visual App

Thank you for contacting us!

How can I help you today? To contact our sales department, press "sales". To contact our support department, press "support". To update your account information, press "update account".

In the example above, a phone-based callflow was quickly adapted for use as a visual application by quickly updating prompt text using visual personas. Also notice how:

- IVR menu options became clickable buttons. When clicked, each button follows the path set in the Callflow Editor.
- If you created help text, it displays when the customer clicks **?**.
- Global paths are respected. For example, the **agent** button is based on the **agent** global default path that is part of the standard application template.

Visual personas

WebIVR applications are based on visual personas that you enable in the Personas view. In the **Persona** tab, go to the persona you are using for your application and select the **Has visual alternative** check box. This allows you to use the same persona to serve both your voice application and your WebIVR application. You can customize the WebIVR persona to use prompts that are more relevant for visual interactions (for example, instead of saying "Thank you for calling," you can set your WebIVR application to say "Thank you for contacting us.")

Themes

You can set the appearance of a WebIVR application by choosing a theme for your visual persona in the Personas view.

Block behavior

All blocks in a WebIVR application function similarly to their role in a voice application. For example, a Message block plays a message in a voice application, whereas in a WebIVR application this block simply states a message on-screen (using the visual persona). The Phone block does not transfer the interaction directly to a phone number; it asks the customer to dial the transfer number (and on a mobile device, customers can usually click the displayed phone number to open the device's dialler).

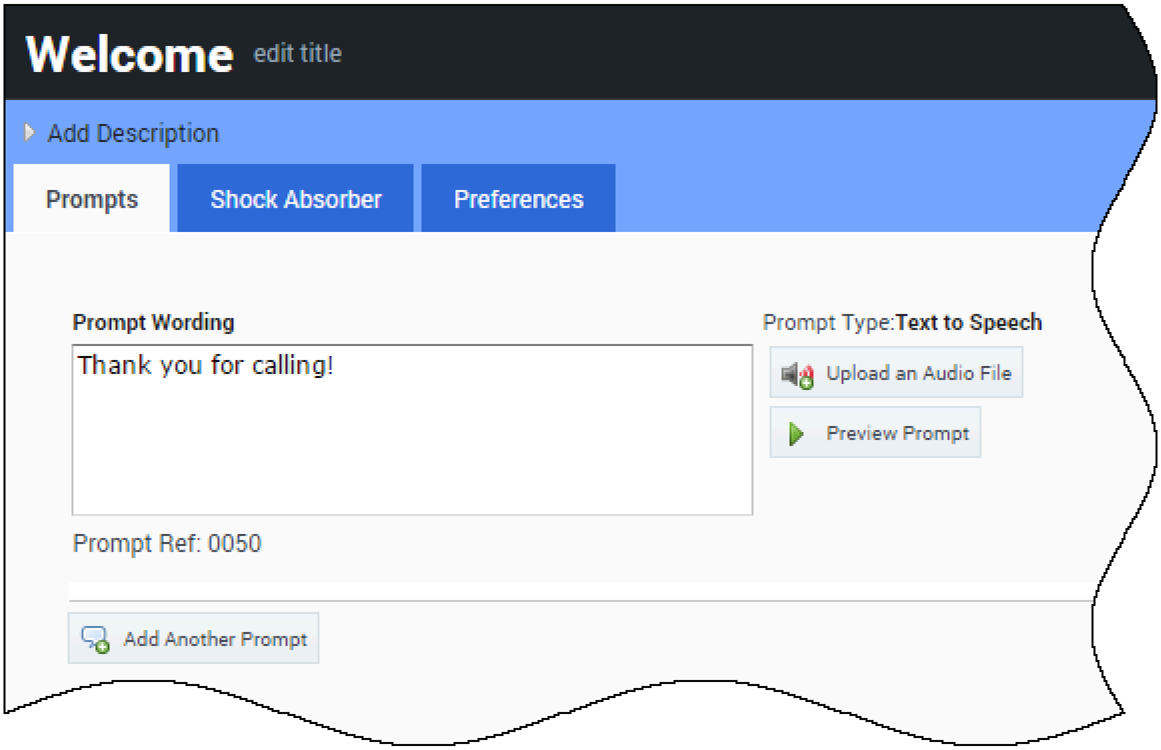
You can insert hyperlinks into certain blocks, such as Message and Menu blocks, by using the following format in a prompt: `[link:URL;Description]`. For example: `[link:https://www.genesys.com;Genesys]`.

> **Warning**
>
> You cannot use custom grammars with WebIVR applications. Only standard grammars are supported.

Getting started

A good first step to prepare your WebIVR application is to go through each block in the Callflow Editor to check and define prompts and settings to ensure the visual persona is properly configured for a WebIVR application.

For example, the Welcome Message prompt below says, "Thank you for calling." However, this message does not make sense in a WebIVR application, as the customer has not dialled the voice application.



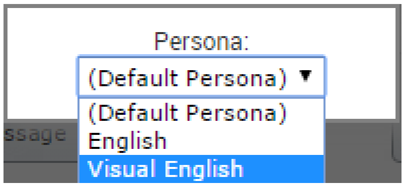
Prompt Wording

Thank you for calling!

Prompt Type:**Text to Speech**

Prompt Ref: 0050

Go to the right of the Message screen and there is a persona selector. Select the visual persona.



Persona:

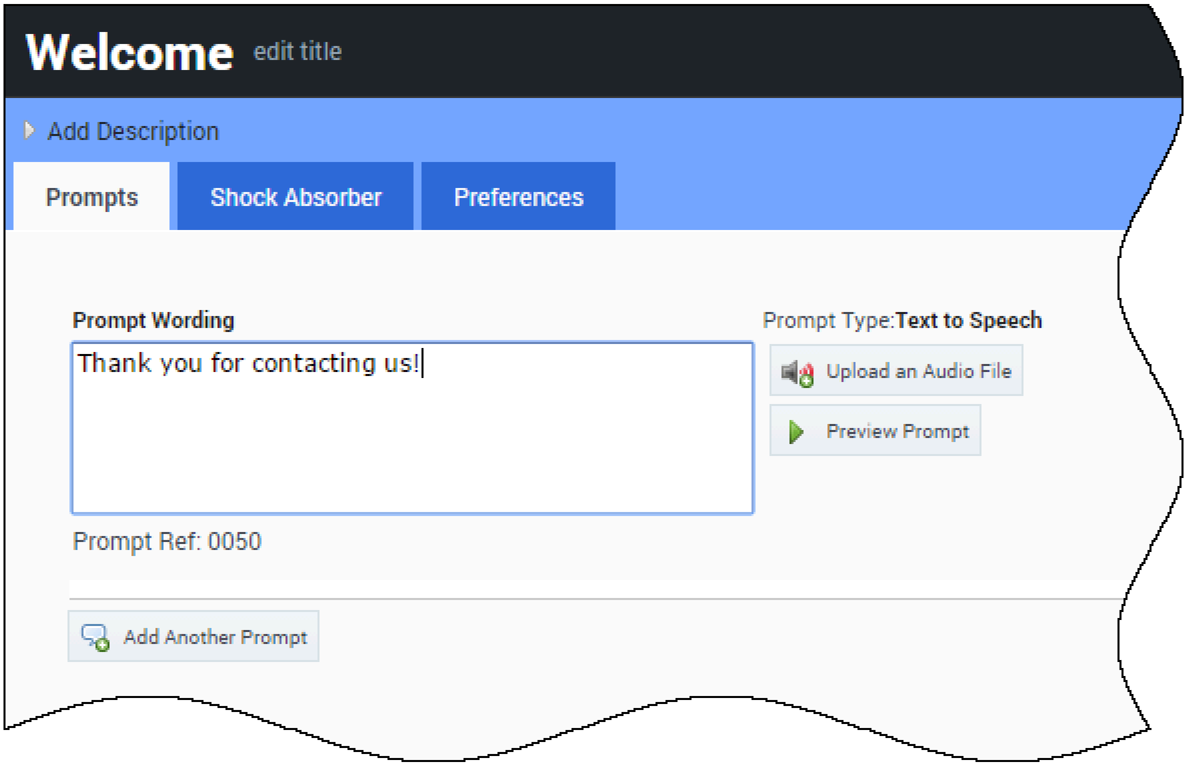
(Default Persona) ▼

(Default Persona)
English
Visual English

Tip

Don't see the visual persona? Remember that you need to enable it in the **Personas** view.

Now you can update the message to something more appropriate for a WebIVR application.



Prompt Wording

Thank you for contacting us!

Prompt Type:**Text to Speech**

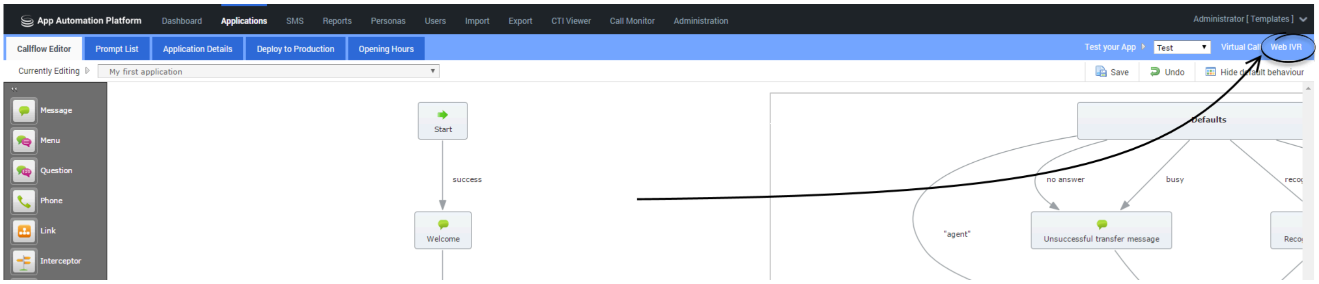
Prompt Ref: 0050

Viewing your WebIVR application

After you have prepared your application, you can click **WebIVR** in the top-right corner to try it out. The WebIVR application opens in a separate window, and you can click through the prompts and options as a customer would see them.

To open the WebIVR location:

- If your environment uses a Load Balancer, the Load Balancer and Messaging Server must be reporting as **Online** in the Administration > Servers tab.
- If your environment does not use a Load Balancer, Messaging Server must be reporting as **Online** in the Administration > Servers tab.



Currently Editing: My first application

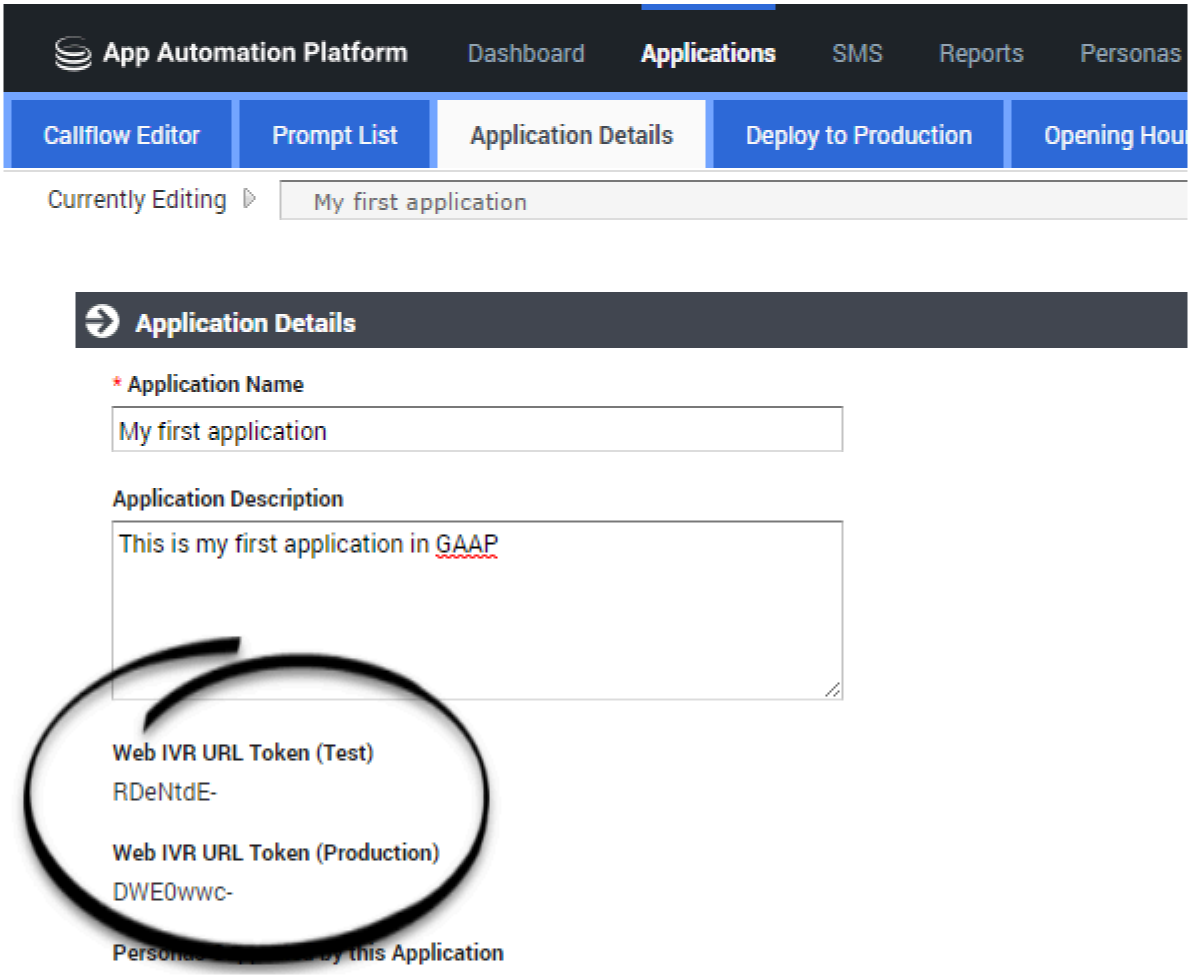
Deploying your WebIVR application

The WebIVR application is deployed to production in the same manner as your voice application.

To use the WebIVR application, you must provide a link that customers can use to access it. This link is based on one of the following formats, depending on whether you use a load balancer for your Messaging server:

- Load balancer before Messaging Server (LB): *http://<LB_ADDRESS>:<LB_PORT>/fish-messaging/go/<application_token>*
- Messaging Server Only (MS): *http://<MS_ADDRESS>:<MS_PORT>/fish-messaging/go/<application_token>*

You can find the *<application_token>* value by clicking the **Application Details** tab in the Callflow Editor and noting the **Web IVR URL Token** value for test or production.



Currently Editing: My first application

* Application Name: My first application

Application Description: This is my first application in GAAP

Web IVR URL Token (Test): RDeNtdE-

Web IVR URL Token (Production): DWE0wwc-

Personas Supported by this Application

Important

You must configure the Default Server Setting **VisualIVR.Security.AllowedDomains** to specify the domains in which the WebIVR application will be embedded before it can be used in your company website.

Passing Parameters in URLs

There are two ways to pass parameters through an URL. Genesys prefers using the secure method.

Passing parameters in a secure way

1. Make a POST request to our Hash Servlet /fish-messaging/hash. The content of the POST should contain all parameters to be passed to the Web IVR as well as the company authentication key (can be found in the **Company Details** section in the GUI). The **Authentication Key** variable name should be *AuthenticationKey*.
2. A hash will be generated and returned as a response to this request.
3. Call the Web IVR URL with the parameters passed in on the query string e.g., */fish-messaging/go/43fa21-?hash=XXXXXXXXX&TestParam=TestValue&MyParam=MyValue* etc.
4. IA will take the parameters and generate a hash on them. It will then compare that hash with the authenticated hash generated previously to make sure they match. If they match, the parameters will be added to the IA context. If not, an error occurs.

Important

The hash is only valid for 30 minutes by default. This can be changed in the **WebIVR.Hash.HashLifetimeInMinutes** default server setting.

Passing parameters in an unsecure way

1. Set the **WebIVR.Hash.PerformHashCheck** default server setting to *false*.
2. Add the parameters to the query string of the visual IVR URL e.g., */fish-messaging/go/43fa21-?TestParam=TestValue&MyParam=MyValue*.
3. They should be picked up by the visual IVR

Important

This is not recommended as it does not allow the variables to be authenticated before adding them to the session

When the variables are added, they are added both as IA and attached variables and also given new names:

- visual_parameter_variable_ParameterName
- visual_parameter_attached_data_ParameterName

You can also add a CLI/ANI or DNIS by adding parameters CLI or DNIS to the request.

Apple Business Chat

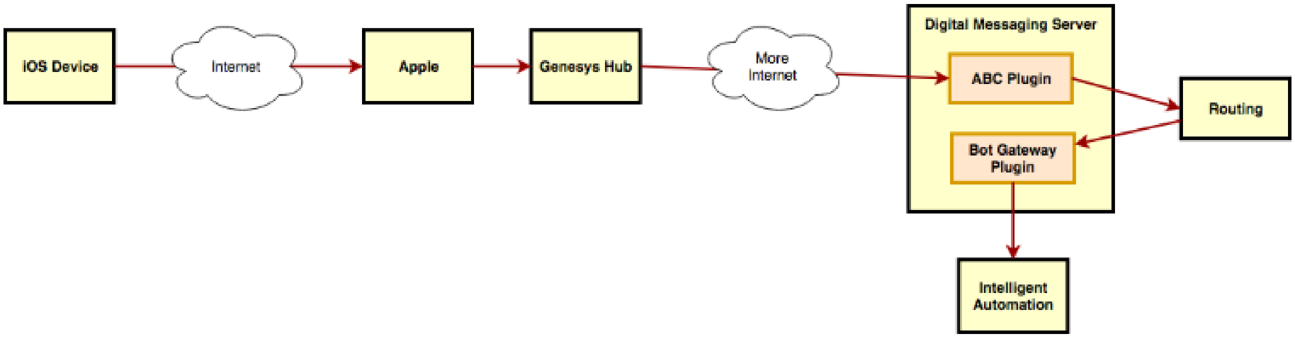
Apple Business Chat enables organizations to connect with customers using Apple's own iMessage framework. Using Business Chat, customers can get answers to questions, resolve issues and complete transactions on their iPhone, iPad, Mac, and Apple Watch.

In Genesys Intelligent Automation, Apple Business Chat is an extension of existing chatbot functionality and provides the look and feel of an Apple application. It works with a callflow to enable your customer to chat with your enterprise. You can set up Apple Business Chat with new or existing callflows.

This page describes the Apple Business Chat feature and how to set it up in Intelligent Automation.

How it works

The following diagram provides a high-level view of how Apple Business Chat works with Genesys Intelligent Automation.



Generally, a customer enters a question, response, or similar information on their iOS device, which sends the message through the Apple Server and Genesys Hub to the Digital Messaging Server. The Routing Server then routes it through the BOT Gateway to Intelligent Automation, which forwards it on to the enterprise. To send a message back to the customer, the enterprise sends a message in a similar route, where Intelligent Automation sends it back to the customer.

Summary

To set up Apple Business Chat in Intelligent Automation, you must do the following:

1. Create or designate a callflow that represents the callflow used by the enterprise to process an Apple Business Chat interaction with the customer. If an existing callflow does not meet your needs, you can create a new one by following the instructions in Using the Callflow Editor

2. Set up the chatbot for Apple Business Chat.
3. Configure the other Genesys servers.
4. Add new prompts and menus, as necessary, to the callflow you are using with Apple Business Chat. For the prompts, add corresponding Standard Responses provided by the Universal Contact Server (UCS).

Set up the chatbot for Apple Business Chat

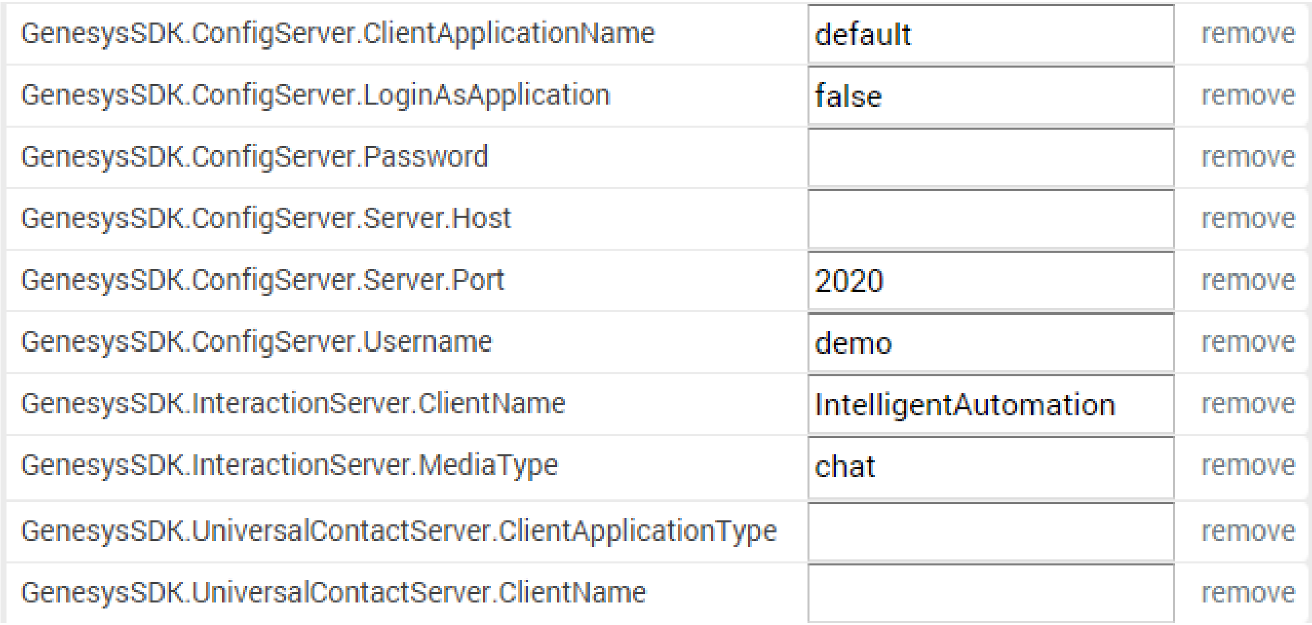
To create and set up a chatbot for Apple Business Chat, do the following:

1. Install and configure Digital Message Server.
2. Configure the chatbot in Interaction Routing Designer.

Configure the other Genesys servers

Chatbots are created with help from other Genesys components (Interaction Server, Universal Contact Server). The configuration information for these other components is stored in the Genesys Configuration Database, and is accessible only through the Genesys Configuration Server. Therefore, settings for Configuration Server and the other servers must be configured in Intelligent Automation.

Enter these settings as **Default Server Settings** under the **Administration** tab, in the fields highlighted in the following diagram. Some values will be entered as defaults, but you can change them as required.

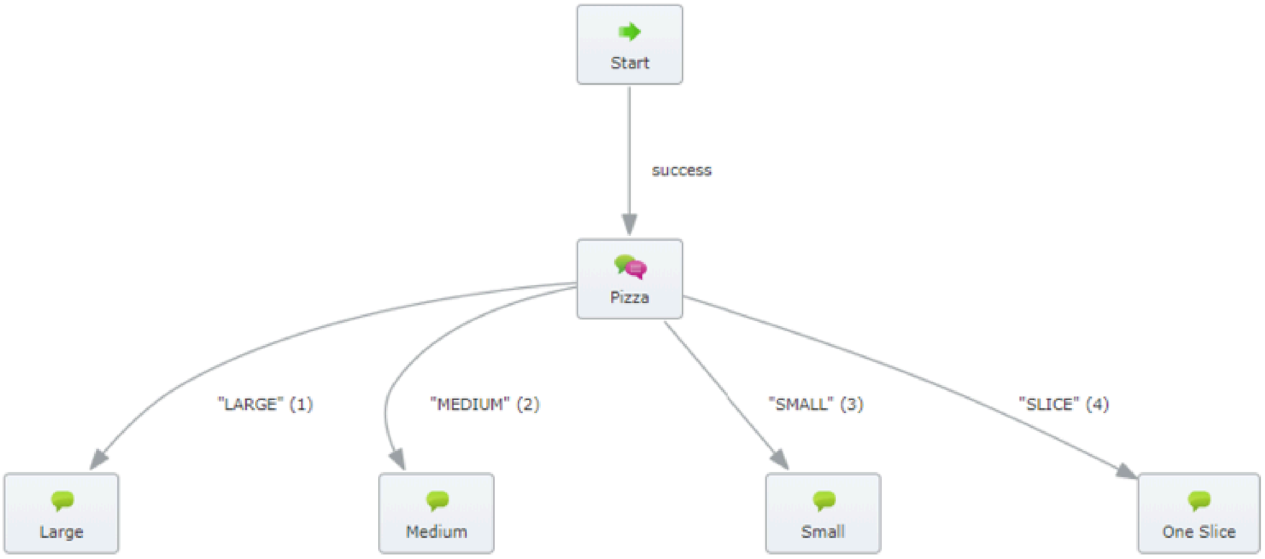


GenesysSDK.ConfigServer.ClientApplicationName	default	remove
GenesysSDK.ConfigServer.LoginAsApplication	false	remove
GenesysSDK.ConfigServer.Password		remove
GenesysSDK.ConfigServer.Server.Host		remove
GenesysSDK.ConfigServer.Server.Port	2020	remove
GenesysSDK.ConfigServer.Username	demo	remove
GenesysSDK.InteractionServer.ClientName	IntelligentAutomation	remove
GenesysSDK.InteractionServer.MediaType	chat	remove
GenesysSDK.UniversalContactServer.ClientApplicationType		remove
GenesysSDK.UniversalContactServer.ClientName		remove

Customize the callflow with menus and prompts

Customize the callflow you are using for Apple Business Chat by adding and modifying menu blocks and prompts, as appropriate.

In Menu blocks, add prompts representing the options in the menu. The chatbot renders these blocks as Apple List Picker items, complete with appropriately illustrated buttons. For example, a Menu block in which the customer selects the size of the pizza they are ordering might look like this:

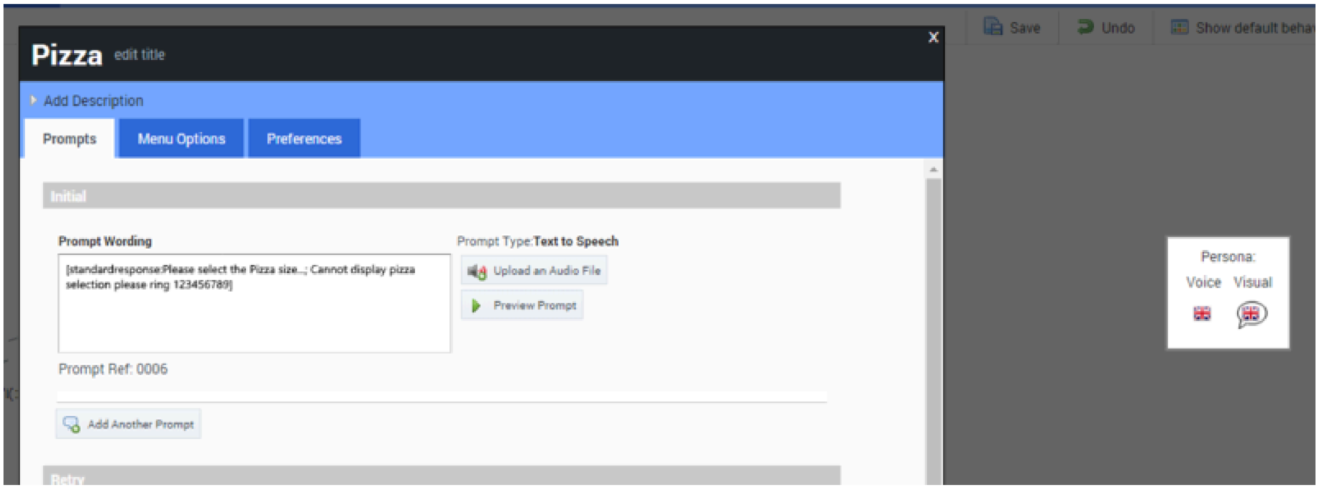


Add prompts as necessary to handle customer responses. When configuring the prompts, enter Standard Responses, defined in UCS, in the following format:

```
StandardResponse:<text of response as defined in UCS>:Any fallback text
```

> ### Important
>
> The **fallback** field for the Standard Response is mandatory. See Standard Responses for configuring Standard Responses.

For the pizza example, the prompt is configured in the Menu block, like this:



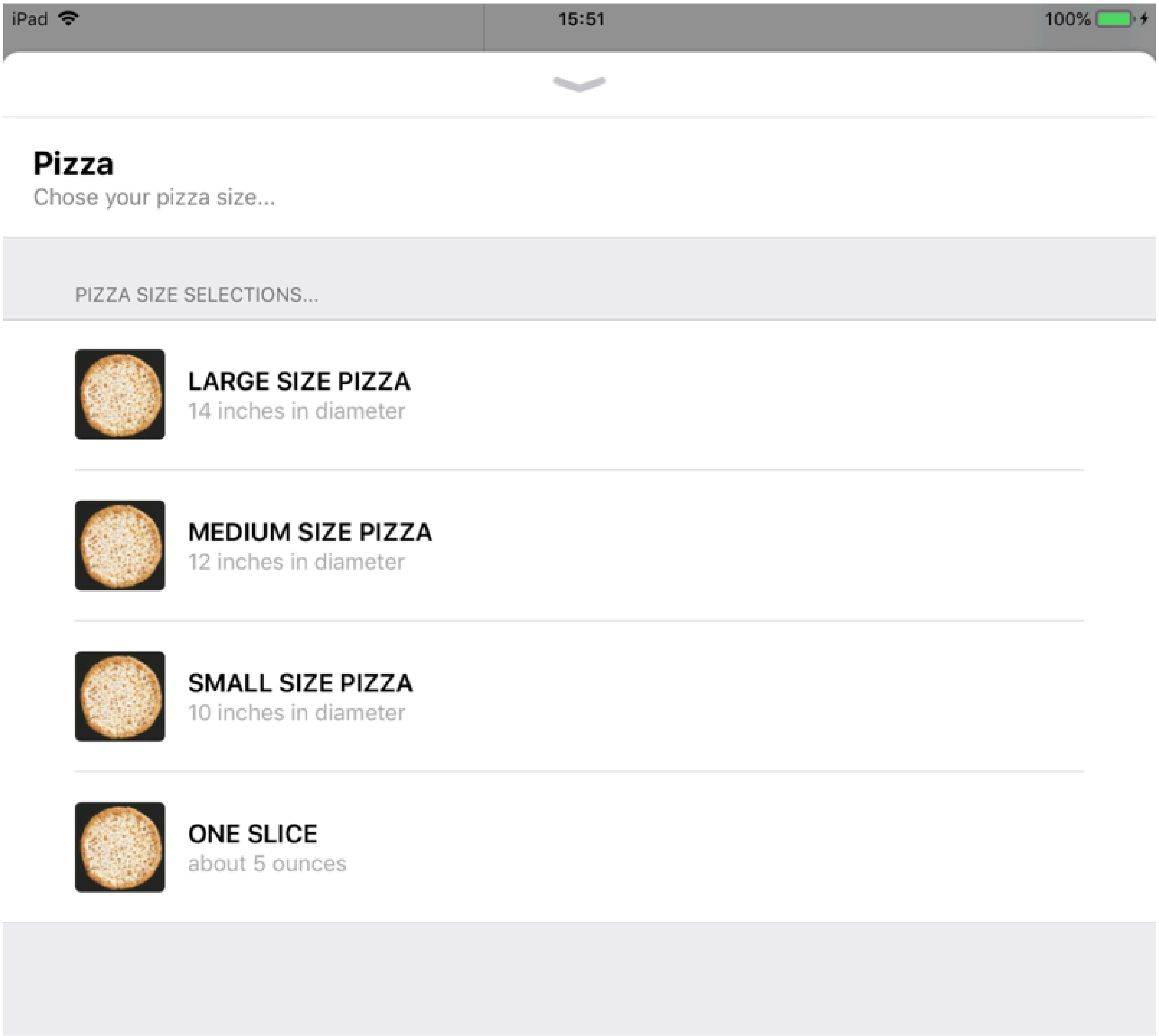
Prompt Wording

[standardresponse:Please select the Pizza size...; Cannot display pizza selection please ring 123456789]

Prompt Type:**Text to Speech**

Prompt Ref: 0006

The chatbox renders this menu, complete with prompts, as an Apple List Picker, like this:



Pizza

Chose your pizza size...

PIZZA SIZE SELECTIONS...

LARGE SIZE PIZZA
14 inches in diameter

MEDIUM SIZE PIZZA
12 inches in diameter

SMALL SIZE PIZZA
10 inches in diameter

ONE SLICE
about 5 ounces

Troubleshooting

This page answers common troubleshooting questions about using Genesys Intelligent Automation.

- **Issue: I've added an option to my Prompt Wording in my Menu, and added a link to this option, but when callers select the number for this option they're not being put through. Why is this?**
    - Suggestion: If you've indicated in the Prompt Wording field that the caller should press a specific number to be put through to a particular department, you must also add this number to the DTMF field in the Edit Path dialog box.
- **Issue: Why are callers receiving an error message rather than being routed to a specific number?**
    - Suggestion: A list of trusted numbers is configured when Intelligent Automation is installed. This prevents callers from being transferred to prohibited numbers, such as premium-rate numbers. Check your reports to identify at which point callers are receiving this error and update the number. This may be in a Phone block, Opening Hours rule, or a Module set up using the Treeview Submodule template. Administrators can add numbers to the trusted numbers list.
- **Issue: Some callers are experiencing technical difficulties at different stages in the callflow. How can I diagnose the problem?**
    - Suggestion: The Call Details report allows you to identify the cause of any problems you are experiencing. To access this report, go to the Reports view and click the Raw Data tab. Select **Call Details** in the **Data Set** menu, and then click **Download** to download a CSV file. Open the file and look at the **Error Messages** column to review where in your callflow a change is required.
- **Issue: During load scenarios, IA Tomcat servers consumes high memory and takes long time to execute Script Blocks, iHub scripts?**
    - Suggestions:
        - Try increasing the Tomcat memory pool size to 6 GB especially for VUI and Integration servers. The default memory pool size is 4 GB.
            - To increase Tomcat available memory, you must change the **tomcat9W.exe** file in the **Tomcat(app)/bin** folder (**../TomcatVUI/bin**) to match the name of the service for that app. e.g., fishVUI.exe.
            - After the change, double-click the file to open the Java config UI and update the **xms** and **xmx** values as desired.
            - Save the settings and restart the application.
        - Add or spin up more Tomcat resources when number of max threads for a Tomcat service exceeds 500 threads by updating the **maxThreads** attribute in the <`Connector ..`> section in the **server.xml** file.
        - Enable caching ("Cache Production Modules") in IA Server's **Administration** page for Production servers.
- **Issue: The Load Balancer sends requests to the wrong messaging server.**
    - Suggestion: You should check the **localhost-access-log** files to see instances of this issue. Find the

incorrect request (look for a 500 response) and then get the session ID. Check if that session ID is in any of the other messaging server **localhost-access-log** files at the same time.

- **Issue: Incorrect `jvmRoute` configuration in server.xml.**
  - Suggestion: The `jvmRoute` option should be configured in the **server.xml** for all messaging and VUI Tomcat servers. This allows them to specify the hostname of the box and append it to the session ID. This value is case-sensitive. Check if you have used the same configuration in all servers.
- **Issue: Setting `cookies=false` in context.xml breaks Multi-modal.**
  - Suggestion : This flag should be set on the `<Context>` element. Set `<Context cookies="false">` value on the Messaging servers only. This also instructs Tomcat not to set session cookies.
  - Setting `cookies="false"` on the VUIs will break Multi-modal.
- **Issue: GIA Tomcat servers generated huge system log files (stderr).**
  - Suggestion: To change the Log level of the stderr.log, change the **logging.properties** file located at the **Platform\<Tomcat>\conf** location.
  - Set the value of **java.util.logging.ConsoleHandler.level** setting to `SEVERE | WARNING | INFO |ALL`. Restart the servers after updating.
- **Issue: Apache Log4j <=2.14.1 JNDI features used in configuration, log messages, and parameters do not protect against attacker controlled LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled. From log4j 2.15.0, this behavior has been disabled by default.**
  - Suggestion: In releases >=2.10, this behavior can be mitigated by setting either the system property `log4j2.formatMsgNoLookups` or the environment variable `LOG4J_FORMAT_MSG_NO_LOOKUPS` to *true*. For releases >=2.7 and <=2.14.1, all `PatternLayout` patterns can be modified to specify the message converter as `%m{nolookups}` instead of just `%m`. For releases >=2.0-beta9 and <=2.10.0, the mitigation is to remove the `JndiLookup` class from the classpath: `zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class`.
- **Issue: In iHub's Deploy to Production process, the What's Included column displays *0 Changed Components*.**
  - Suggestion: This is because after an update, there is no preceding value to be compared. Hence, the columns shows *0 Changed Components*.
- **Issue**: When creating new tags in Block Reports, if an error, *Failed to save the selected blocks* is displayed in the UI and the Catalina logs include the following error message: *More than the maximum number of request parameters (GET plus POST) for a single request ([<maxParameterCount>]) were detected. Any parameters beyond this limit have been ignored.*
  - Suggestion: Update the `maxParameterCount` value in the **server.xml** file of the Tomcat GUI server. The default value is '1000000'.
- **Issue: Users are unable to log in to the GUI**
  - Suggestion:
    1. Check if the Flex manager in configured correctly and connected.

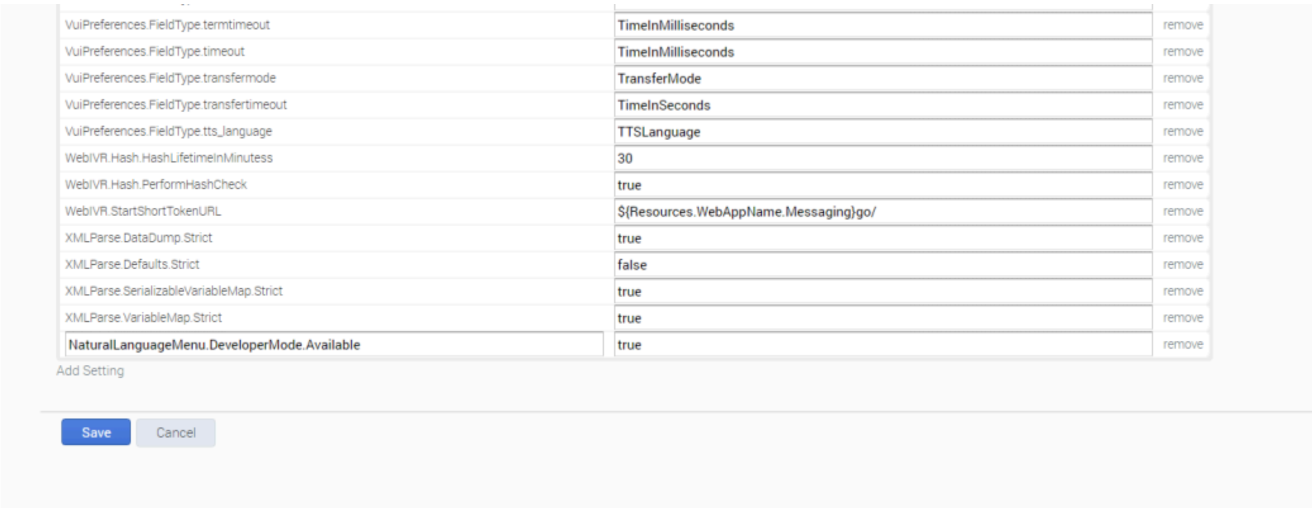
2. If this issue occurs for a specific non-administrator user, check if that user is blocked in CME.

- **Issue: In the product import page of Intelligent Automation, the `This Tomcat Server is Linked to Eclipse` checkbox is set.**
  - Suggestion: This is a developer-only option. **Uncheck this option by default** as the Intelligent Automation is not linked to Eclipse in a customer environment.

Developer Mode

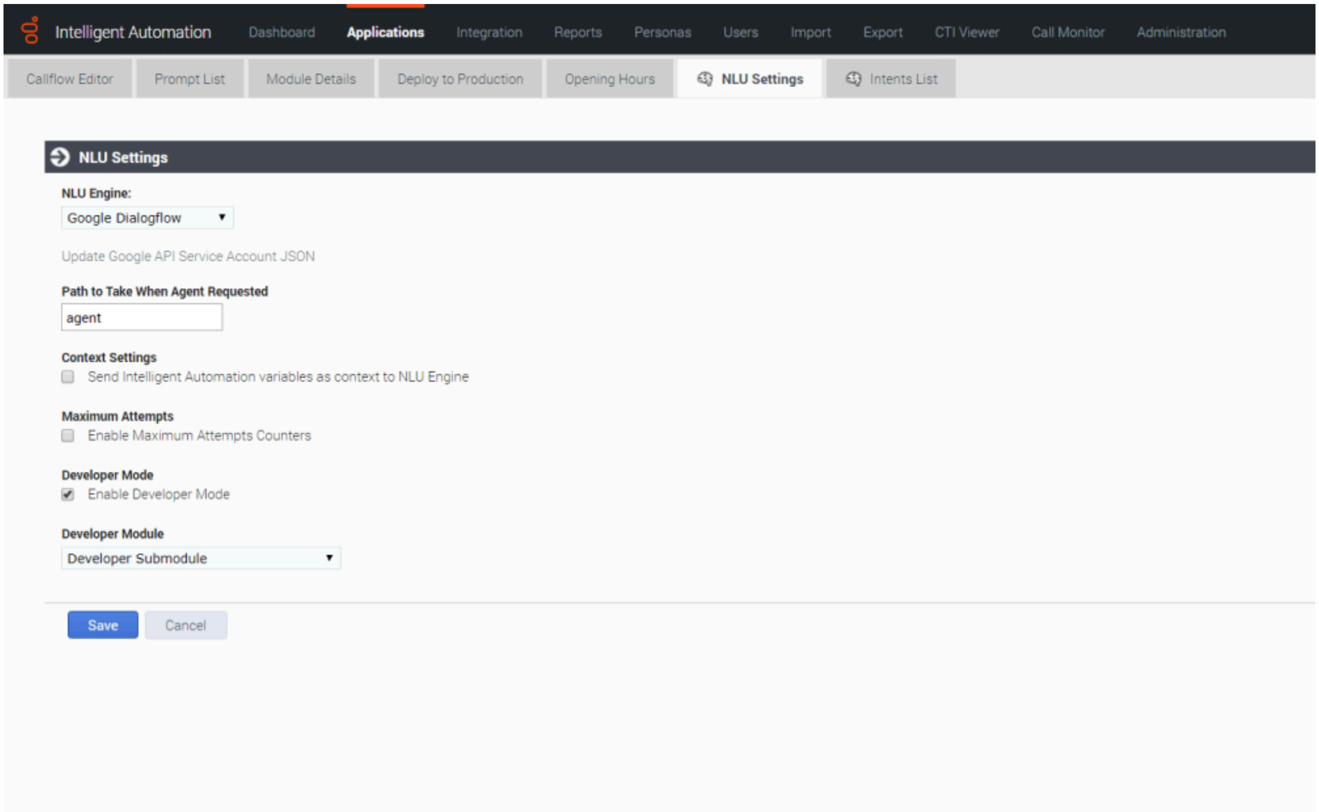
The Developer Module feature allows users to set a module that can be used to intercept responses within an NLU callflow. This module can perform additional operation with the data and can either proceed further in the callflow or break out of the callflow. You can choose which module will be used as the Developer Module.

To enable the Developer Module feature, create a new entry in the **Default Server Settings** page available under **Administration** with the name, **NaturalLanguageMenu.DeveloperMode.Available** and set the value as *True*.



VuiPreferences.FieldType.termtimeout	TimeInMilliseconds	remove
VuiPreferences.FieldType.timeout	TimeInMilliseconds	remove
VuiPreferences.FieldType.transfermode	TransferMode	remove
VuiPreferences.FieldType.transfertimeout	TimeInSeconds	remove
VuiPreferences.FieldType.tts_language	TTSLanguage	remove
WebIVR.Hash.HashLifetimeInMinutess	30	remove
WebIVR.Hash.PerformHashCheck	true	remove
WebIVR.StartShortTokenURL	${Resources.WebAppName.Messaging}go/	remove
XMLParse.DataDump.Strict	true	remove
XMLParse.Defaults.Strict	false	remove
XMLParse.SerializableVariableMap.Strict	true	remove
XMLParse.VariableMap.Strict	true	remove
NaturalLanguageMenu.DeveloperMode.Available	true	remove

Once the setting is enabled, a **Developer Module** option will be available in the **NLU Settings** page where you can choose the module that will be used as the developer module.



NLU Settings

NLU Engine:
Google Dialogflow

Update Google API Service Account JSON

Path to Take When Agent Requested
agent

Context Settings
□ Send Intelligent Automation variables as context to NLU Engine

Maximum Attempts
□ Enable Maximum Attempts Counters

Developer Mode
☑ Enable Developer Mode

Developer Module
Developer Submodule

Document Change History

This page summarizes the changes made in this document for release 9.0.0.

Changes in release 9.0.004.01

- Added page Logging In.
- Expanded description of **Force User to Change Password on Next Login** field when creating users.
- Added new **Configuration Server Tenant DBID** field when creating companies.
- Added description of Working with module parameters in Applications and Modules.

Changes in release 9.0.003.00

- Added page describing Apple Business Chat.

Changes in release 9.0.002.00

There were no changes in this document for release 9.0.002.00

Changes in release 9.0.001.01

There were no changes in this document for release 9.0.001.01

Changes in release 9.0.000.01

- Genesys App Automation Platform has been renamed Genesys Intelligent Automation. Note that some folder and path names still include GAAP and/or SpeechStorm, and these will be rebranded as the product evolves.
- Enhancements to iHub are described on the Integration and the Using iHub pages. Some existing material on these pages has been clarified.
- Customer journey enhancements are described on the Reports page.