This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Genesys Intelligent Automation Reference Guide

Genesys Intelligent Automation Current

6/21/2023

Table of Contents

Genesys Intelligent Automation Reference Guide

9.x **Genesys Intelligent Automation** is part of 9.x, which can include component releases from both 9.1.x, 9.0.x, and 8.5.x code streams. See Genesys Intelligent Automation to check which component releases are part of 9.x. **Genesys Intelligent Automation** is part of 9.x, which can include component releases from both 9.1.x, 9.0.x, and 8.5.x code streams. See Genesys Intelligent Automation to check which component releases are part of 9.x.

This document provides useful reference information pertaining to your Genesys Intelligent Automation deployment.

About Intelligent Automation

Features of Intelligent Automation

Hive-off Processes

User Authentication and Authorization

This section describes how Genesys Intelligent Automation ensures that only authenticated and authorized users access Intelligent Automation environment specifically, and the Genesys software environment generally.

For more information about user authentication and authorization, refer to the User Authentication and User Authorization section of the *Genesys Security Deployment Guide*.

Internal and External Users

There are two types of users in Intelligent Automation:

- Internal users: Created and managed in Intelligent Automation. See the Users page in Intelligent Automation Help. These users are not subject to an external authentication engine; all authentication is done by Intelligent Automation. All authorization is implemented through Roles.
- External users: Created and managed in Genesys Configuration Server. Authentication is done internally by Configuration Server and sometimes by an external authentication engine, such as LDAP (Lightweight Directory Access Protocol), and RADIUS (Remote Authentication Dial In User Service). Authorization is done in Configuration Server. External users are supported starting in release 9.0.004.00.

The two user types mean that when creating a new user, the administrator must first decide whether to create the user in Intelligent Automation or in Configuration Server.

Prerequisites for external users

If you are working with external users, you must do the following:

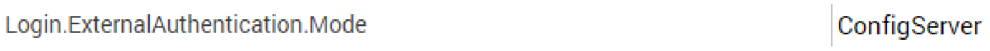
1. Enable Intelligent Automation to work with external users.
2. Configure Intelligent Automation access to Configuration Server.
3. Import a Solution Package Definition (SPD) file containing all role privileges, into the Genesys Configuration Database.
4. Link tenant IDs in the Configuration Database to Intelligent Automation companies. Identify a *default* company for those users that might not belong to a tenant.

> ### Important
>
> When configuring custom roles, ensure that the name for a custom role in the SPD file matches the name created in the **Administration > Roles** option. See Roles in Intelligent Automation and Roles tab in

> Administration for more information.

Configure working with external users

To configure Intelligent Automation to work with external users, set the parameter **Login.ExternalAuthentication.Mode** to `ConfigServer` in the **Default Server Settings** tab under **Administration** in the Intelligent Automation interface:

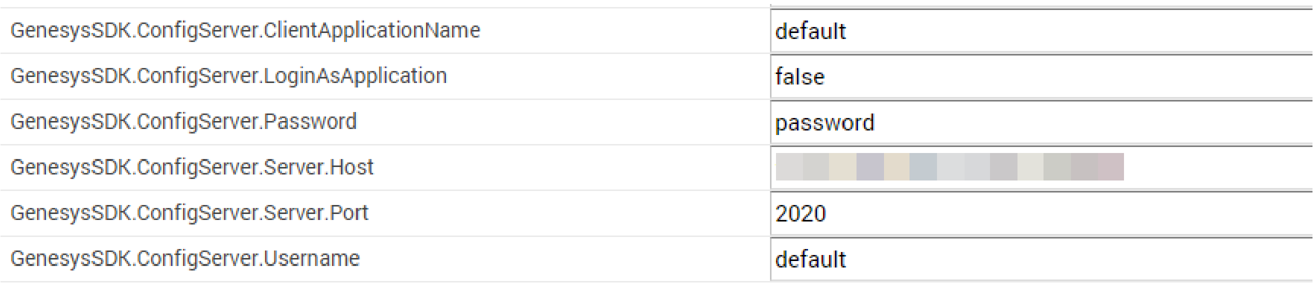


Login.ExternalAuthentication.Mode	ConfigServer

The default value of this parameter is `None`, meaning that external users cannot log in to Intelligent Automation.

Configure access to Configuration Server

To configure Intelligent Automation to work with external users, Intelligent Automation must call upon Configuration Server to, among other things, validate the user. Therefore, you must provide the information required to access Configuration Server in the **Default Server Settings** tab, as follows:



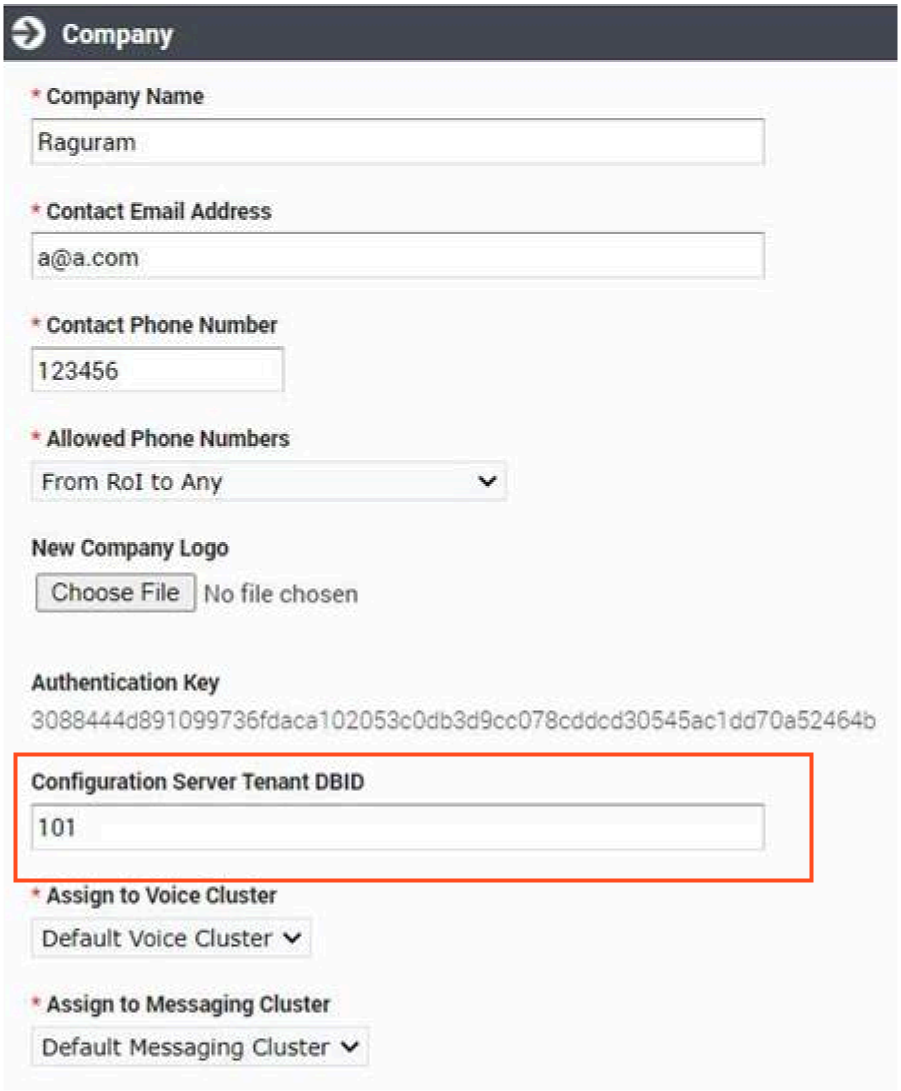
GenesysSDK.ConfigServer.ClientApplicationName	default
GenesysSDK.ConfigServer.LoginAsApplication	false
GenesysSDK.ConfigServer.Password	password
GenesysSDK.ConfigServer.Server.Host	[redacted]
GenesysSDK.ConfigServer.Server.Port	2020
GenesysSDK.ConfigServer.Username	default

Import SPD file into Configuration Database

A Solution Package Definition (SPD) file contains all role privileges for the associated solution. Without it, validated external users would be logged in with no permissions or roles. An SPD file called **IA_SPD_Roles.xml** is provided to you by Intelligent Automation if you choose to manage external users via Configuration Server. It must be imported into the Genesys Configuration Database. For information about installing the SPD file, see Solution Deployment or the Genesys Administrator Extension Developer's Guide.

Linking tenant IDs to Intelligent Automation companies

Each Intelligent Automation company should be linked to a tenant in the Configuration Database by specifying the DBID of that Tenant in the **Configuration Server Tenant (DBID)** field when configuring the company. When an external user logs into Intelligent Automation, the company corresponding to the user's Tenant is loaded for that user.



Company

* Company Name

Raguram

* Contact Email Address

a@a.com

* Contact Phone Number

123456

* Allowed Phone Numbers

From RoI to Any

New Company Logo

Authentication Key

3088444d891099736fdaca102053c0db3d9cc078cddcd30545ac1dd70a52464b

Configuration Server Tenant DBID

101

* Assign to Voice Cluster

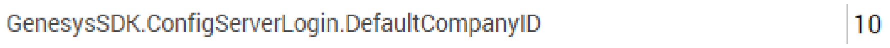
Default Voice Cluster

* Assign to Messaging Cluster

Default Messaging Cluster

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

If a company is not linked to a particular tenant, or the tenant does not exist, you can specify a *default* company in the **GenesysSDK.ConfigServerLogin.DefaultCompanyID** parameter under **Administration** > **Default Server Settings**.



GenesysSDK.ConfigServerLogin.DefaultCompanyID 10

If a default company is not specified, any login by an external user without a linked tenant ID is rejected.

> ## Important
>
> If the external user belongs to the environment tenant, Intelligent Automation will use the **GenesysSDK.ConfigServerLogin.DefaultCompanyID** option to determine the default company ID to be loaded.

Standard Responses

When a user runs a callflow app to fetch the standard response from UCS, Intelligent Automation will look for the company TenantID configured in the **Configuration Server Tenant (DBID)** field first. If this field is empty, the value of the default company ID configured in the DefaultCompanyID (**Administration** > **Default Server Settings**).

If both TenantId and defaultID are not configured, then IA would assume the default tenantID as Environment (TenantID : 1)

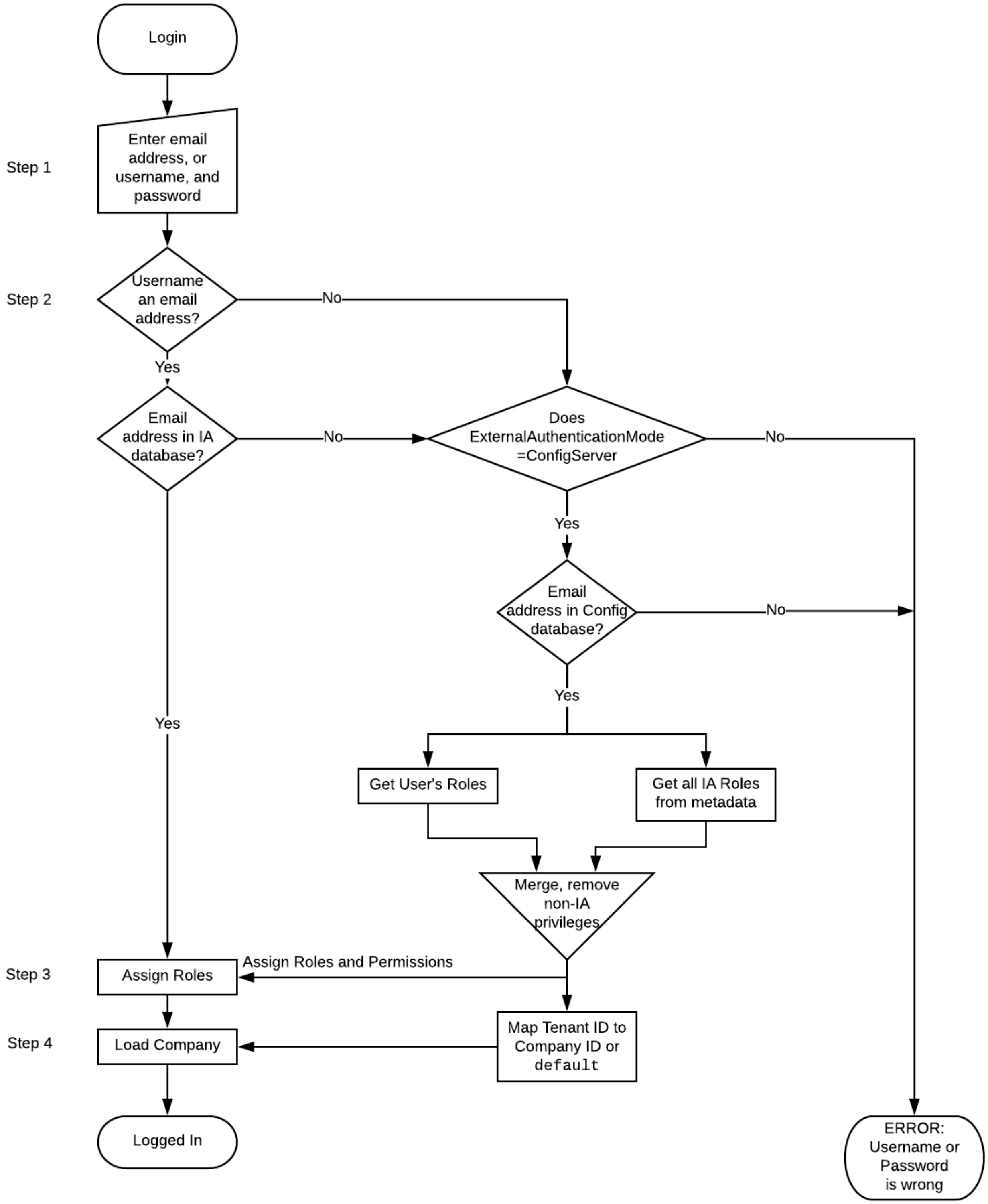
At login

At login, the following high-level actions occur:

1. The user enters their username and password. For internal users, the username is their email address. For external users, the username is their Configuration Server username.
2. The user is authenticated, or validated, by having their user's credentials compared to internally-stored credentials in the application's database.
3. Permissions and roles are assigned to the user, based on the information in their respective database. Permissions define what the user can see, while roles define what the user can do to those items it can see.
4. The company associated with the user is loaded. For external users, the company is based on the Tenant ID with which the user is associated or the **GenesysSDK.ConfigServerLogin.DefaultCompanyID** parameter if no association of the default company is found.

With these four steps completed, the user is now in the Dashboard of the Intelligent Automation interface, ready to start working.

The following flow chart illustrates the login process:


Login
Step 1
Enter email address, or username, and password
Step 2
Username an email address?
No
Yes
Email address in IA database?
No
Does ExternalAuthenticationMode =ConfigServer
No
Yes
Email address in Config database?
No
Yes
Yes
Get User's Roles
Get all IA Roles from metadata
Merge, remove non-IA privileges
Step 3
Assign Roles
Assign Roles and Permissions
Step 4
Load Company
Map Tenant ID to Company ID or default
Logged In
ERROR: Username or Password is wrong

Changing passwords for external users

Passwords for external users can be changed in Intelligent Automation. The process of managing them is completely transparent to the Intelligent Automation user.

> ### Important
> Passwords can be changed only for external users who are configured in Configuration Server.

If **Change password at next login** is checked in the external user's profile in Configuration Server when the external user logs in to Intelligent Automation, the user is prompted to change their password in the same way as for an internal user. See How do I change my password. The password is changed in Configuration Server automatically and **Change password at next login** is cleared in the external user's profile.

Using WebIVR MicroApps

This page describes how you can use WebIVR-based MicroApps in chat interactions to accomplish various business tasks. For example, if an agent is helping a customer with a purchase and you want to securely collect the customer's credit-card information, you can use a MicroApp to securely capture the information without requiring it to go through the agent.

See the Detailed process summary section, below, for more information on the Intelligent Automation widget architecture and how the process executes among various Genesys components.

Prerequisites

The following prerequisites are required:

- Intelligent Automation 3.6.x or higher with Messaging Server
- Workspace Desktop Edition or Workspace Web Edition (8.5.117.07 and higher)
- Genesys Widgets
- The following eServices components:
    - Chat Server (8.5.104.10 and higher)
    - Interaction Server (8.5.109.01 and higher)
    - Knowledge Manager (8.5.x and higher)
    - Genesys Mobile Services (8.5.107.15 and higher)
- Configuration Server (part of Management Framework) (8.5.101.08 and higher)

Configuring your environment

Update Genesys Widgets framework for Intelligent Automation

> **Important**
>
> The Genesys Widgets framework must be deployed in your environment before completing this section.

1. Go to the Intelligent Automation installation folder (for example, **C:\GAAP\Platform\TomcatMessaging\webapps\fish-messaging\widgets**).

2. Copy the latest widget JavaScript prefilters and CSS files from the Intelligent Automation folder noted in Step 1 to your website.
    a. Add **cx-speechstorm.css** to the folder containing other style sheets used within your website.
    b. Add **cx-speechstorm.js** to the JavaScript folder containing **widgets.min.js**.
3. Integrate the Intelligent Automation widgets into your existing website. Refer to **index.html**, which is found in the Intelligent Automation folder noted in Step 1, as a guide. This file contains various imports for the style sheets and JavaScript files that you copied in the previous step, along with the Standard CX Widget Instrumentation Script (which you should already have running as part of the Widgets framework).
4. In the file **index.html**, copy the script block that contains **sChatServerUrl** and **sSpeechStormServer** variables. You must update these variables to point to valid servers within your business. Verify all paths for imports are correct, according to the locations into which you copied files in the previous step.
    a. Update **sChatServerUrl** to point to your chat server. For example:

    ```
    var sChatServerURL = "http://<your_server>/gms_port_8010/genesys/2/chat/customer-
     support";
    ```

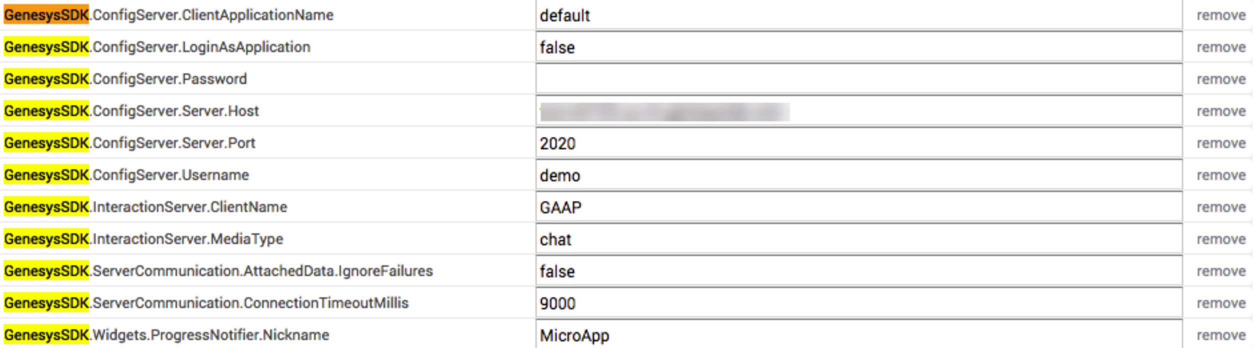
    b. Update **sSpeechStormServer** to point to your Intelligent Automation Messaging Server. For example:

    ```
    var sSpeechStormServer = "http://<Messaging_Server:Port";
    ```

Add default server settings in Intelligent Automation

Next, you must log in to Intelligent Automation and add default server settings for use with the widget framework:

1. Log in to Intelligent Automation.
2. Go to **Administration > Default Server Settings**.
3. Configure the following server settings as necessary for your environment. In particular, ensure you set the correct host and port for Configuration Server, as defined in **GenesysSDK.ConfigServer.server.host** and **GenesysSDK.ConfigServer.server.port**.



GenesysSDK.ConfigServer.ClientApplicationName	default	remove
GenesysSDK.ConfigServer.LoginAsApplication	false	remove
GenesysSDK.ConfigServer.Password		remove
GenesysSDK.ConfigServer.Server.Host	[redacted]	remove
GenesysSDK.ConfigServer.Server.Port	2020	remove
GenesysSDK.ConfigServer.Username	demo	remove
GenesysSDK.InteractionServer.ClientName	GAAP	remove
GenesysSDK.InteractionServer.MediaType	chat	remove
GenesysSDK.ServerCommunication.AttachedData.IgnoreFailures	false	remove
GenesysSDK.ServerCommunication.ConnectionTimeoutMillis	9000	remove
GenesysSDK.Widgets.ProgressNotifier.Nickname	MicroApp	remove

4. Open Windows Services and restart Messaging Server.

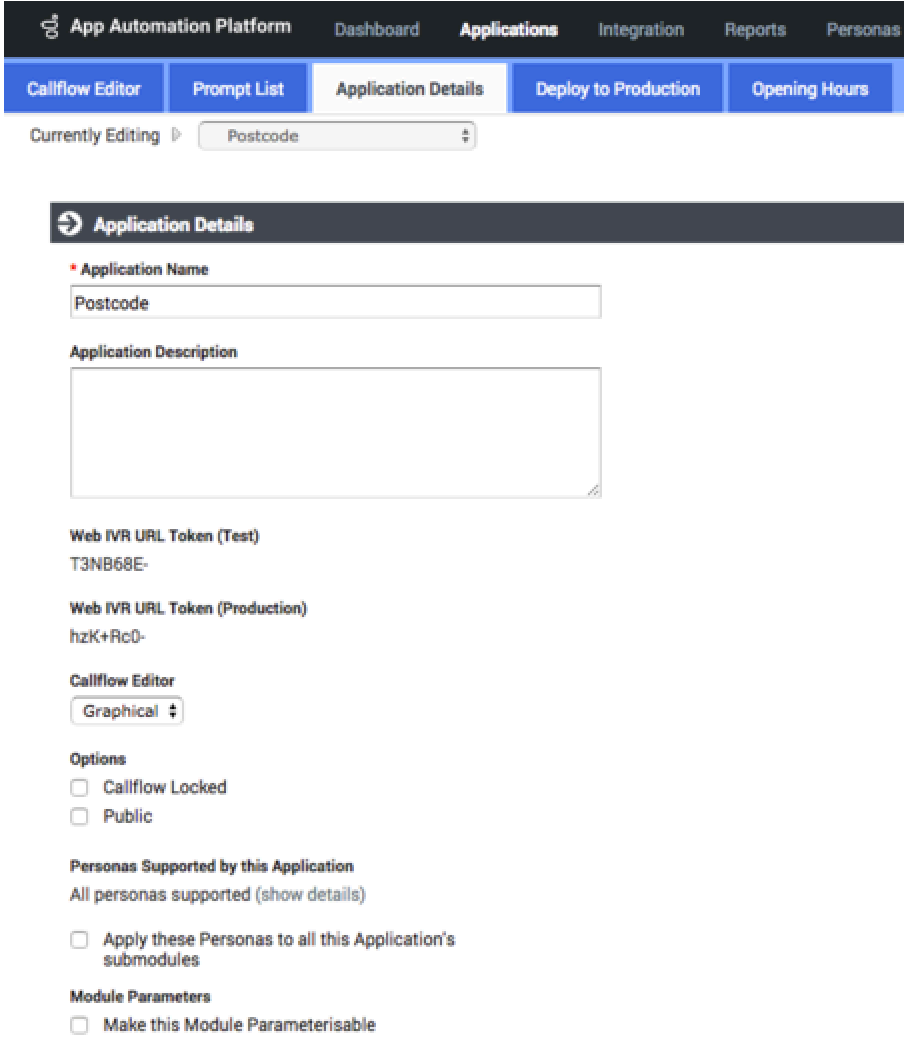
Using MicroApps

To use a MicroApp, an agent in a chat interaction with a customer enters a MicroApp URL into the chat. This chat message sets which MicroApp is used, and defines variables that Chat Server passes to the WebIVR application that powers the MicroApp. For example:

```
microapp://app/T3NB68E-/Payment
AccountNumber=1234567
Amount=USD77.80
```

In the example above, the following parameters are set:

- **T3NB68E-** - This is the WebIVR URL token that is unique to this particular WebIVR application. To find the token for your WebIVR application, open your WebIVR application in the Callflow Editor and click the **Application Details** tab. This tab displays two tokens - one for the test version of your application and one for the production version.



Currently Editing: Postcode

* Application Name: Postcode

Application Description:

Web IVR URL Token (Test): T3NB68E-

Web IVR URL Token (Production): hzK+Rc0-

Callflow Editor: Graphical

Options
□ Callflow Locked
□ Public

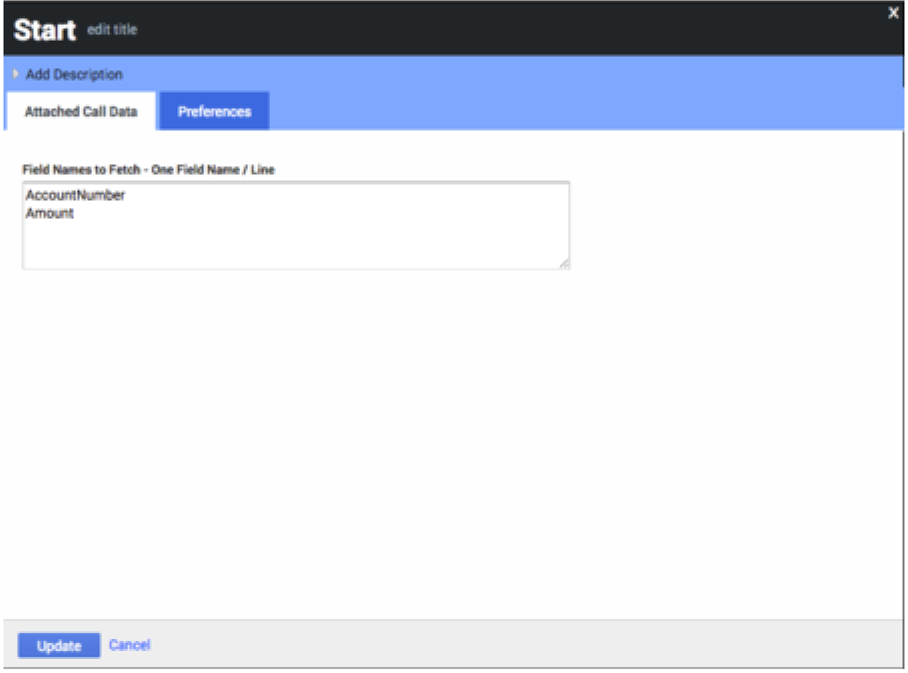
Personas Supported by this Application: All personas supported (show details)

□ Apply these Personas to all this Application's submodules

Module Parameters
□ Make this Module Parameterisable

- **Payment** - This parameter is optional. It allows you to provide a name that describes the purpose of the widget to be launched.

- **AccountNumber=1234567** and **Amount=USD77.80** - These are parameters that Workspace passes into the Intelligent Automation widget. In this case, it specifies the **AccountNumber** variable has a value of **1234567** and the payment **Amount** is **USD77.80**. You define these variables in the **Start** block of your WebIVR application, as seen below:



Field Names to Fetch - One Field Name / Line

```
AccountNumber
Amount
```

Important

Contact your Genesys representative for more information on parameters you can use with your widget implementation.

Use Knowledge Manager to set up standard responses

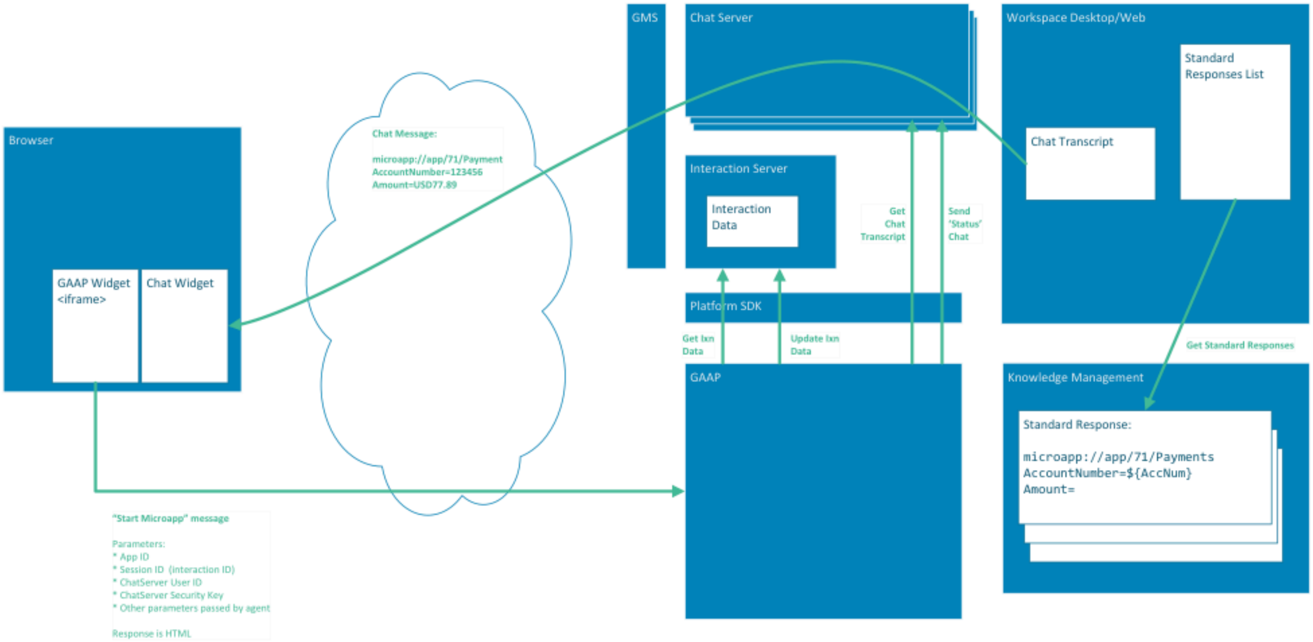
For a more efficient workflow, you can use Knowledge Manager (part of eServices) to set up standard responses that agents can insert into chats to invoke MicroApps.

Refer to the Genesys Knowledge Manager documentation for more information.

Detailed process summary

This section describes in more detail how MicroApps interact with Intelligent Automation and the Genesys environment.

The following graphic explains the high-level architecture of how MicroApps interact with other Genesys components:



The list below provides a step-by-step account of how Intelligent Automation interacts with the Genesys environment to provide the MicroApp service.

1. The agent invokes the MicroApp URL and parameters in a chat window with a customer, preferably using a standard response defined in Knowledge Manager.
2. After the agent sends the URL to the customer, the Intelligent Automation widget uses a pre-filter to prevent MicroApp messages from displaying in the customer's chat window.
3. The Intelligent Automation widget intercepts the *microapp request* message and forwards the request in the correct format to the Intelligent Automation Messaging Server, along with the session ID (also known as Interaction ID), the Chat Server user ID and security key for that session, the MicroApp App ID, and any additional parameters provided by the agent.
4. When the MicroApp session begins, the Intelligent Automation Messaging Server contacts Interaction Server to retrieve any attached data already present in the interaction. This data is made available to the MicroApp in the same way as SIP Server attached data is made available in a voice call.
5. Intelligent Automation Messaging Server contacts Chat Server with the user ID and security key, and retrieves the chat transcription for this session.
6. After receiving the chat transcription, the Intelligent Automation Messaging Server finds the matching *microapp request* message in the transcription to verify that parameters have not been changed by the customer or a third party.
7. As the MicroApp progresses, the Intelligent Automation Messaging Server send *status* chat messages to the agent. These messages state the location of the customer in the MicroApp. The Intelligent Automation widget filters these status messages so that they do not appear in the customer's chat window.
8. The MicroApp might use logic to attach data to the interaction. If so, the Intelligent Automation Messaging Server sends messages directly to Interaction Server. This results in an *interaction data update* notification in Workspace and updates to values in any Case Information fields. For example, you can use this functionality to notify agents when the caller has identified himself or herself.
9. After the MicroApp ends, the Intelligent Automation widget sends a pre-filtered *status* chat message to

the agent on behalf of the user to signify the MicroApp portion of the interaction is complete.

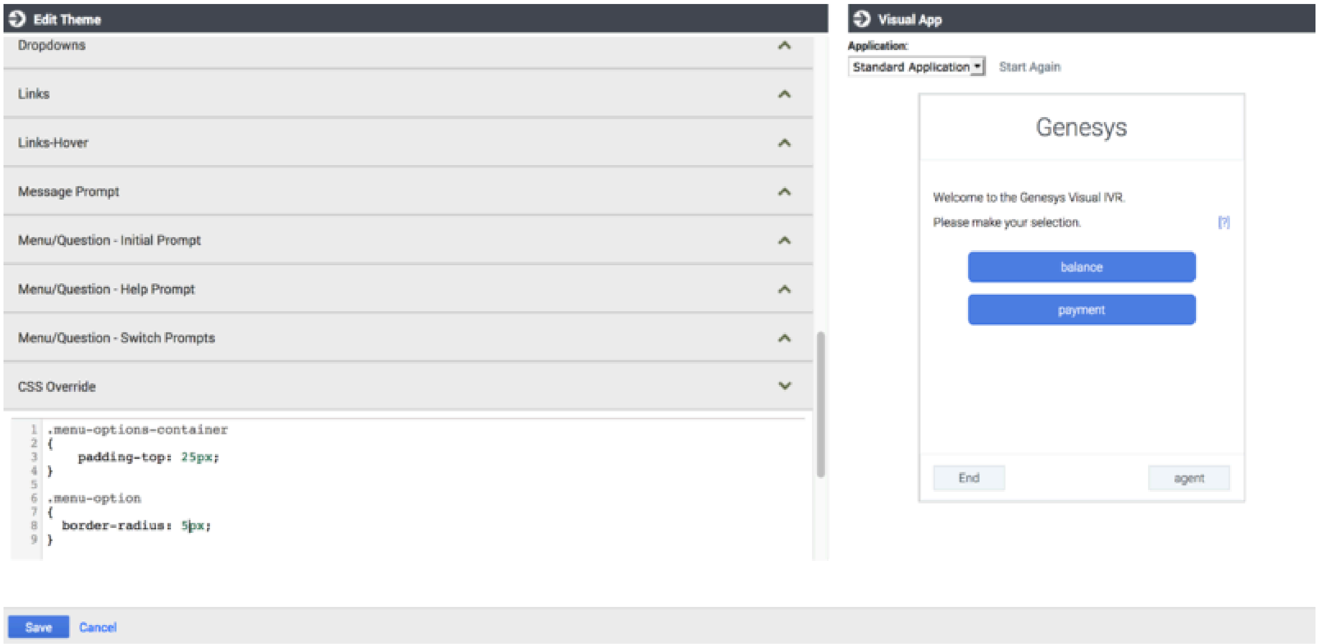
Important

During the whole interaction described above, the agent and customer can send normal chat messages to one another without affecting the MicroApp's execution.

Applying CSS to WebIVR MicroApps

This page describes the CSS classes that are rendered into HTML at runtime for WebIVR MicroApps.

Genesys Intelligent Automation provides a user-friendly interface for designing visual themes for MicroApps (refer to the Help for more information). However, the information on this page allows web designers to directly configure CSS for MicroApps using the **CSS Override** section in the **Personas > Themes** view.



```
.menu-options-container
{
    padding-top: 25px;
}

.menu-option
{
  border-radius: 5px;
}
```

Welcome to the Genesys Visual IVR.

Please make your selection.

> **Important**
>
> The examples on this page might contain various HTML and scripting code. You can ignore these elements and focus only on the elements related to CSS styling.

Overview

Body

The MicroApp body uses the class **body**.

```
<body class="body">
```

Title bar

The MicroApp title bar uses a div element and an h1 element. The div element uses the class **title-bar** and the h1 element uses the class **title-bar-text**.

```
<div class="title-bar">
<h1 class="title-bar-text" id="title-bar-text">Title</h1>
</div>
```

Central area

The central area of the MicroApp renders into the body and uses three div elements, with the classes being **outer**, **middle**, and **blocks-container** (from outermost to innermost).

```
<div class="outer">
<div class="middle">
 <div class="blocks-container" id="blocks-container">
...
</div>
</div>
</div>
```

Footer

The MicroApp footer uses the class **footer**. GAAP uses footers to hold buttons related to dialog defaults, as well as the **End** button (the **End** button is always present to allow the caller to end the session, as closing the window does not end the session on iOS Safari). Defaults such as **Agent** also render in the footer.

MicroApps render buttons as anchor elements with the **default-menu-option** class. The global menu option has the **menu_option_agent** class (the name of the global option is appended). Consequently, you can individually style added options.

The footer is divided into two other div elements (with classes **close-footer-button-container** and **footer-buttons**) to separate the **End** button and the other menu options.

```
<div class="footer">
<div class="close-footer-button-container">
<a class="default-menu-option" id="close-window-button">End</a>
</div>
<div class="footer-buttons" id="footer-buttons">
<a class="default-menu-option menu_option_agent">Agent</a>
</div>
</div>
```

Blocks

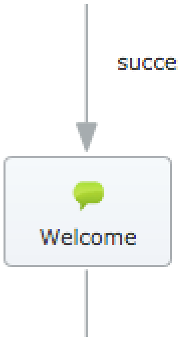
Blocks render inside the div class **blocks-container**.

```
<class="blocks-container" id="blocks-container">
```

As the rendered class is also the name of the block, and thus separate from other elements, you can assign individual styling to each block.

If the block name has spaces, GAAP replaces these spaces with underscores and prepends the word **block**. If the block uses a persona, a **persona** class is also rendered onto the block.

The following example shows a **Message** block with no set persona:



The above example renders the following:

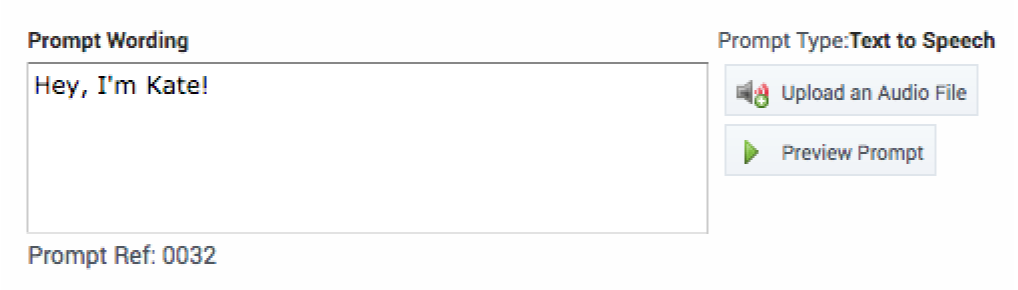
```
<div class="block block_Welcome persona_">
```

Where:

- **block**—specifies this class as a block.
- **block_Welcome**—specifies the block name with **block_** prefix.
- **persona_**—specifies the persona used by the block. In this example, the block uses the default persona and a persona name is not specified.

Prompts

You can use prompts in Message, Menu, Question, and **Recording** blocks. Prompts render inside div elements that identify the block to which they belong. You can insert prompts by using their reference number. For example, the prompt below uses reference **0032**.



Prompt Wording

Hey, I'm Kate!

Prompt Type:**Text to Speech**

Prompt Ref: 0032

This renders as:

```
<div class="prompt-0032">
Hey, I'm Kate!
</div>
```

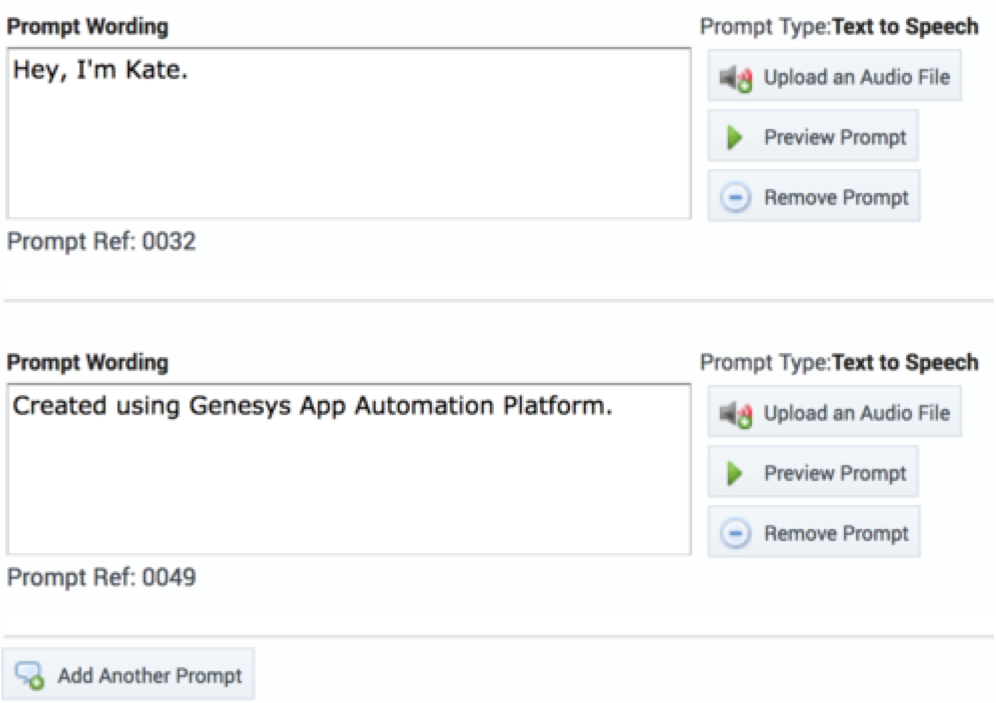
The following section describes prompt use by block type.

Message blocks

Message blocks output a prompt to the IVR or, in this case, text to a MicroApp. GAAP renders them in a div with classes **indent-right** and **message-prompt**. Inside another div, **Message** blocks render with a class that is individual to the prompt by using the prompt reference, as noted above in the Prompts section.

```
<div class="indent-right message-prompt">
<div class="prompt-0032">
Hey, I'm Kate!
</div>
</div>
```

Blocks can contain multiple prompts, with each prompt referring to the prompt reference number, as shown in the following example:



Prompt Wording: Hey, I'm Kate.

Prompt Type:**Text to Speech**

Prompt Ref: 0032

Prompt Wording: Created using Genesys App Automation Platform.

Prompt Type:**Text to Speech**

Prompt Ref: 0049

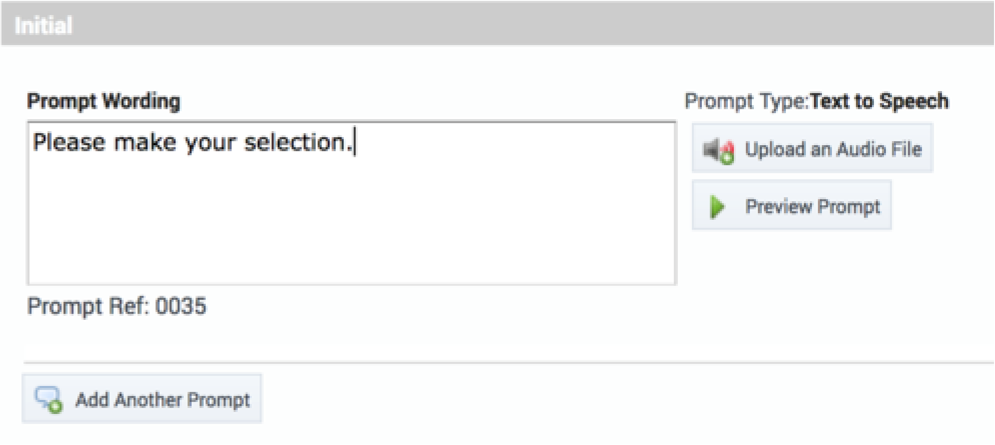
The example above renders as:

```
<div class="indent-right message-prompt">
<div class="prompt-0032">
Hey, I'm Kate!
</div>
<div class="prompt-0049">
Created using Genesys App Automation Platform.
</div>
</div>
```

Menu and Question blocks

You can group prompts in these blocks into a div element with class **menu-prompts-container**.

Initial prompt



Initial

Prompt Wording

Please make your selection.

Prompt Type:**Text to Speech**

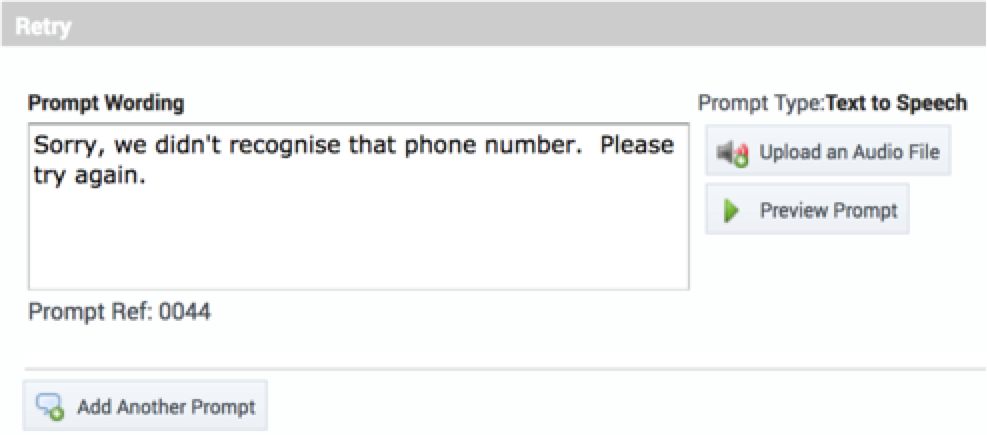
Prompt Ref: 0035

The above graphic shows text in the Initial prompt section. Consequently, this renders an outer div with class **menu-initial-prompt** and an inner div.

```
<div class="menu-initial-prompt" id="initialPromptDiv">
<div class="prompt-0035">
Please make your selection.
</div>
</div>
```

Retry prompt

If the user fails validation on a **Question** block (such as by entering an invalid phone number), the Retry prompt displays below the input area, with the class **validation-message**. The input field also has the class **validationErrorFrame** to highlight the area.



Retry

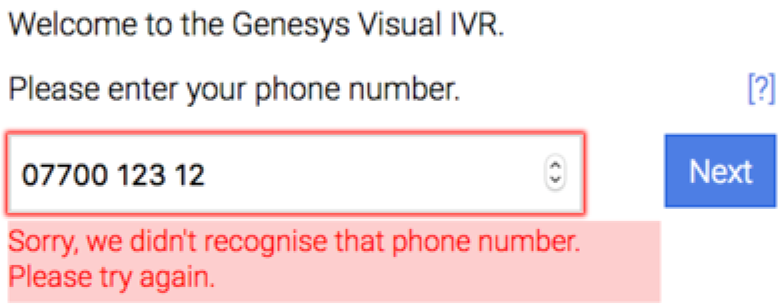
Prompt Wording

Sorry, we didn't recognise that phone number. Please try again.

Prompt Type:**Text to Speech**

Prompt Ref: 0044

The graphic below shows how this prompt appears in the MicroApp:


Welcome to the Genesys Visual IVR.

Please enter your phone number.

07700 123 12

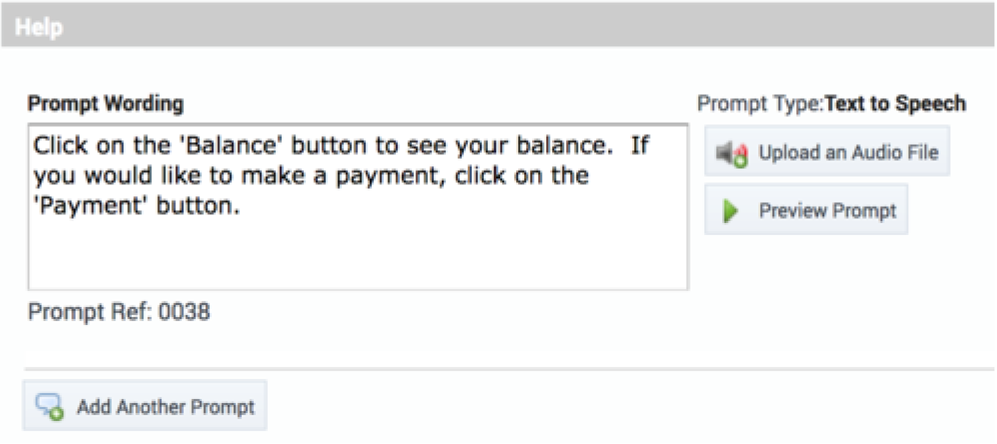
Sorry, we didn't recognise that phone number. Please try again.

Intelligent Automation renders this as a span element inside the **question-input-container** class, below the input field.

```
<span class="validation-message">Sorry, we didn't recognize that phone number.  Please try
again.</span>
```

Help prompt

The Help prompt is initially hidden in the MicroApp—it displays only if the user clicks the **?** link rendered beside the Initial prompt.



Click on the 'Balance' button to see your balance. If you would like to make a payment, click on the 'Payment' button.

Prompt Ref: 0038

When the user clicks **?**, the following outer div with class **menu-help-prompt** displays, and the original prompt is hidden until **?** is clicked again.

```
<div class="menu-help-prompt" id="helpPromptDiv">
<div class="prompt-0038">
Click 'Balance' to see your balance.  If you would like to make a payment, click 'Payment'.
</div>
</div>
```

The question mark rendered with the class **switch-prompts**:

```
<a class="switch-prompts">[?]</a>
```

The render for the prompts thus far is:

```
<div class="menu-prompts-container">
```

```
<div class="menu-initial-prompt" id="initialPromptDiv">
<div class="prompt-0035">
Please make your selection.
</div>
</div>
<div class="menu-help-prompt” id="helpPromptDiv">
<div class="prompt-0038">
Click 'Balance' to see your balance.  If you would like to make a payment, click 'Payment'.
</div>
</div>
<a class="switch-prompts">[?]</a>
</div>
```

Buttons

If the menu uses buttons, each menu choice renders as an input element. Intelligent Automation assigns the **menu-option** class to each choice, and then appends an individual class with the name of the option, such as **menu_option_<choice here>**.

```
<input class="menu-option menu_option_balance" value="balance" type="submit">
<input class="menu-option menu_option_payment" value="payment" type="submit">
```

Choices in Question blocks

The following sections describe the different grammar types and how they dictate the HTML and associated CSS classes rendered. Each choice selection has an associated **Confirm** button rendered with the **submit-button** class.

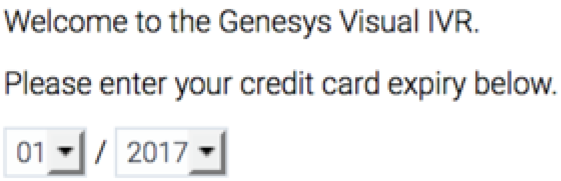
```
<input class="submit-button" type="submit" value="Next">
```

Credit Card Expiry

Use the ID selector as shown below.

```
<select id="CreditCardExpiryMonths">
<option value="01">01</option>
<option value="02">02</option>
<option value="03">03</option>
…
<option value="10">10</option>
<option value="11">11</option>
<option value="12">12</option>
</select>
/
<select id="CreditCardExpiryYears">
<option value="2020">2020</option>
<option value="2019">2019</option>
<option value="2018">2018</option>
<option value="2017" selected="">2017</option>
<option value="2016">2016</option>
<option value="2015">2015</option>
<option value="2014">2014</option>
</select>
```

The preceding code renders as:



Welcome to the Genesys Visual IVR.

Please enter your credit card expiry below.

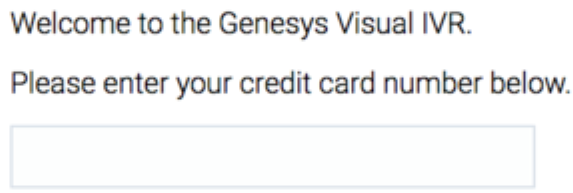
01 / 2017

Credit card number

Credit card numbers use an input box with the class **question-input-field**.

```
<input class="question-input-field"="text">
```

The preceding code renders as:



Welcome to the Genesys Visual IVR.

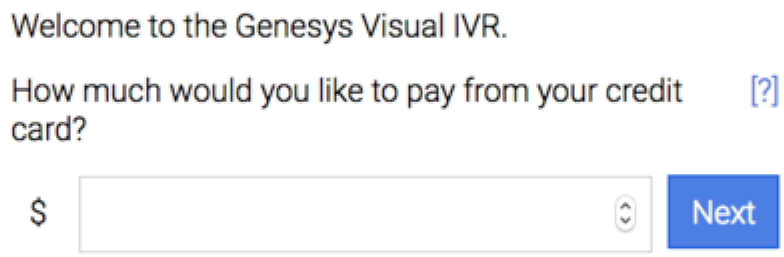
Please enter your credit card number below.

Currency

Currency grammars use a label and an input field with the classes **currency-symbol-indent** and **question-input-field**, respectively.

```
<label class="currency-symbol-indent"> € </label>
<input class="question-input-field" type="number">
```

The preceding code renders as:



Welcome to the Genesys Visual IVR.

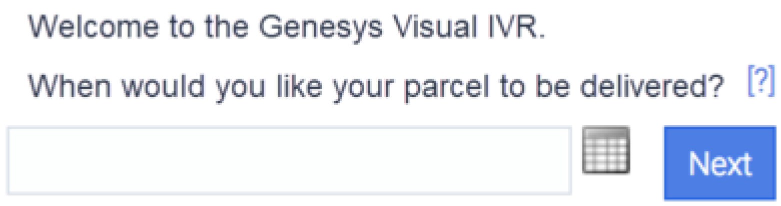
How much would you like to pay from your credit card?

Date

Date grammars render two components. First, it renders an input field that holds the date, which is styled with the **question-input-field** class. An icon is placed alongside the input field, which opens the date-picker when clicked. This link uses the **date-picker-control** class.

```
<input class="question-input-field" id="Date" name="Date" type="text">
<a class="date-picker-control" title="Show Calendar" id="fd-but-Date">
<span class="fd-screen-reader">Show Calendar</span>
</a>
```

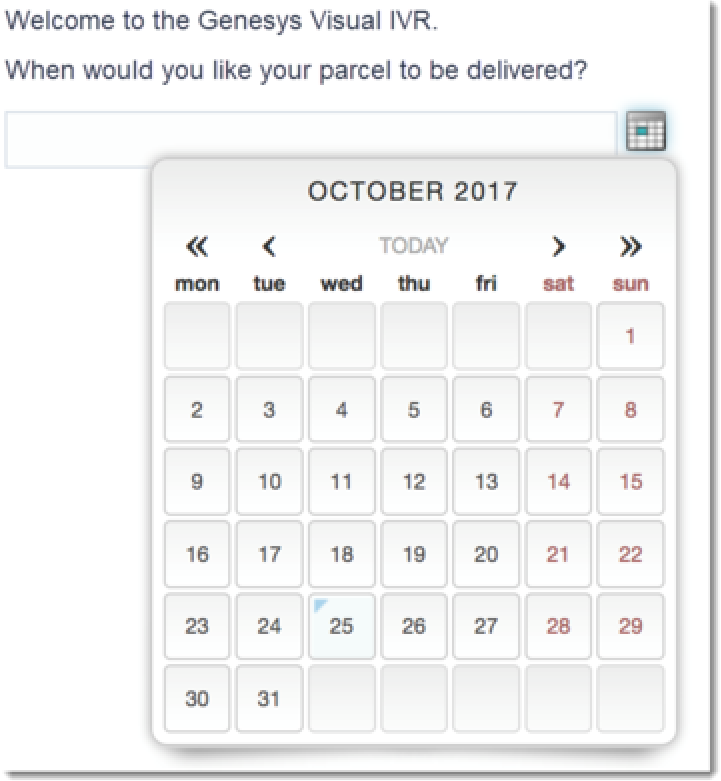
The preceding code renders as:



Welcome to the Genesys Visual IVR.

When would you like your parcel to be delivered? [?]

When the user clicks the icon, the date-picker opens.



Welcome to the Genesys Visual IVR.

When would you like your parcel to be delivered?

OCTOBER 2017

« ‹ TODAY › »

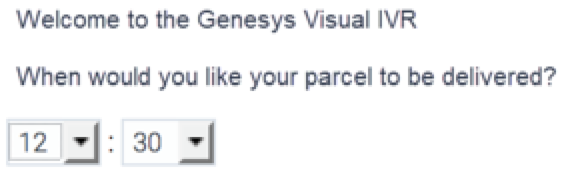
mon	tue	wed	thu	fri	sat	sun
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Time

The time input renders as drop-down menus—one for hours and one for minutes. Each uses an ID selector.

```
<select id="TimeHours">
<option value="00">00</option>
<option value="01">01</option>
<option value="02">02</option>
<option value="03">03</option>
…
<option value="21">21</option>
<option value="22">22</option>
<option value="23">23</option>
</select>
:
<select id="TimeMinutes">
<option value="00">00</option>
<option value="01">01</option>
<option value="02">02</option>
…
<option value="57">57</option>
<option value="58">58</option>
<option value="59">59</option>
</select>
```

The preceding code renders as:



Welcome to the Genesys Visual IVR

When would you like your parcel to be delivered?

Natural numbers

Natural numbers use a standard field with the **question-input-field** class.

```
<input class="question-input-field" type="number">
```

Telephone

Telephone numbers use a standard field with the **question-input-field** class.

```
<input class="question-input-field" type="tel">
```

Grammar Builder

Grammar Builder uses a standard field with the **question-input-field** class.

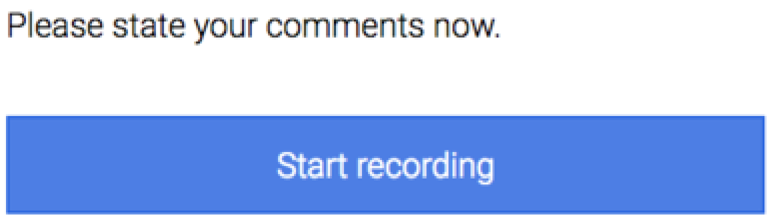
```
<input class="question-input-field" type="text">
```

Recording blocks

The **Recording** block requests permission from users to use their microphone to capture an audio recording. It displays buttons to start and stop the recording, which use the **recording-button-wrapper** class. After recording, the user clicks the **Next** button, which uses the **recording-done-button-wrapper** class. These elements are wrapped inside a form that can be styled using the ID selector **recordingform**. Above these controls is a prompt for indicating to users that they can make a recording, using the standard prompt CSS format mentioned earlier on this page.

```
<div class="prompt-0047">
Please state your comments now.
</div>

<form id="recordingform">
<div class="recording-button-wrapper">
<input class="recording-button" type="submit" value="Start recording">
</div>
<div class="recording-done-button-wrapper">
<input class="submit-button" type="submit" value="Next">
</div>
</form>
</div>
```

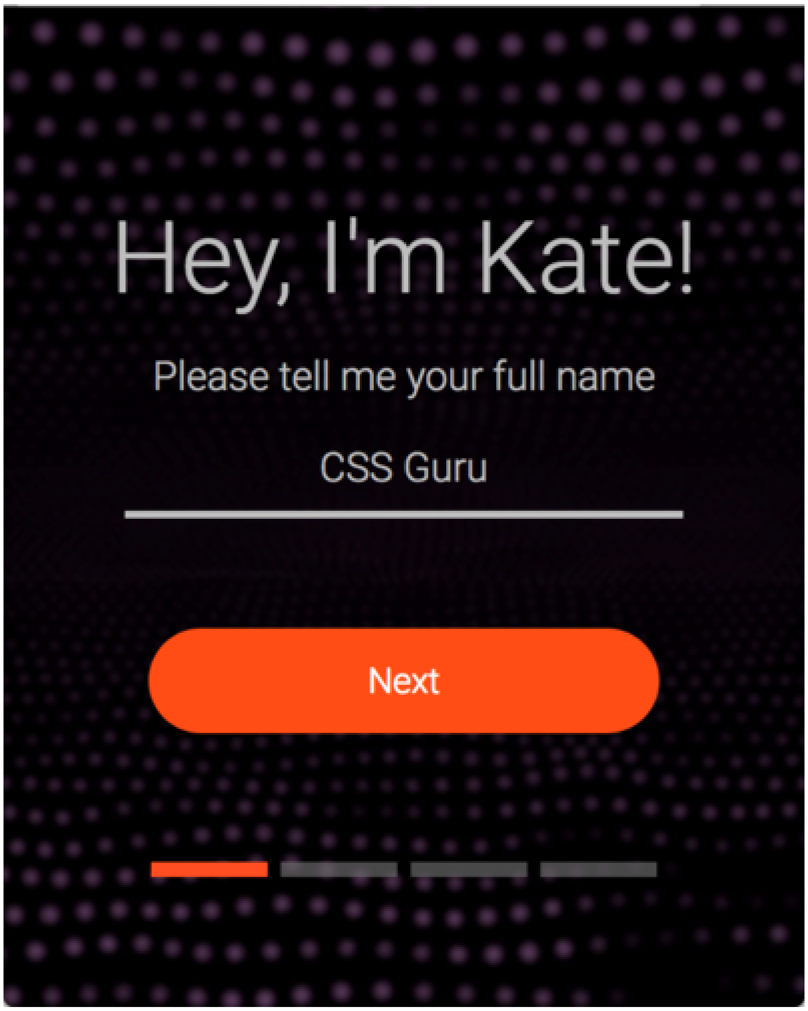
The preceding code renders as:


Please state your comments now.

Styling Tips

This section provides basic tips for styling MicroApps.

- Make full use of the space available, while taking into consideration the various screen formats that you need to support. For example, are you targeting phones only, or also desktops? Use media queries for responsive design to ensure your MicroApp looks good regardless of device.
- The example below uses a large button and simple messaging to make the goal obvious to the user. It also uses a background image for depth. It maintains a consistent style and font, and elements are centered.

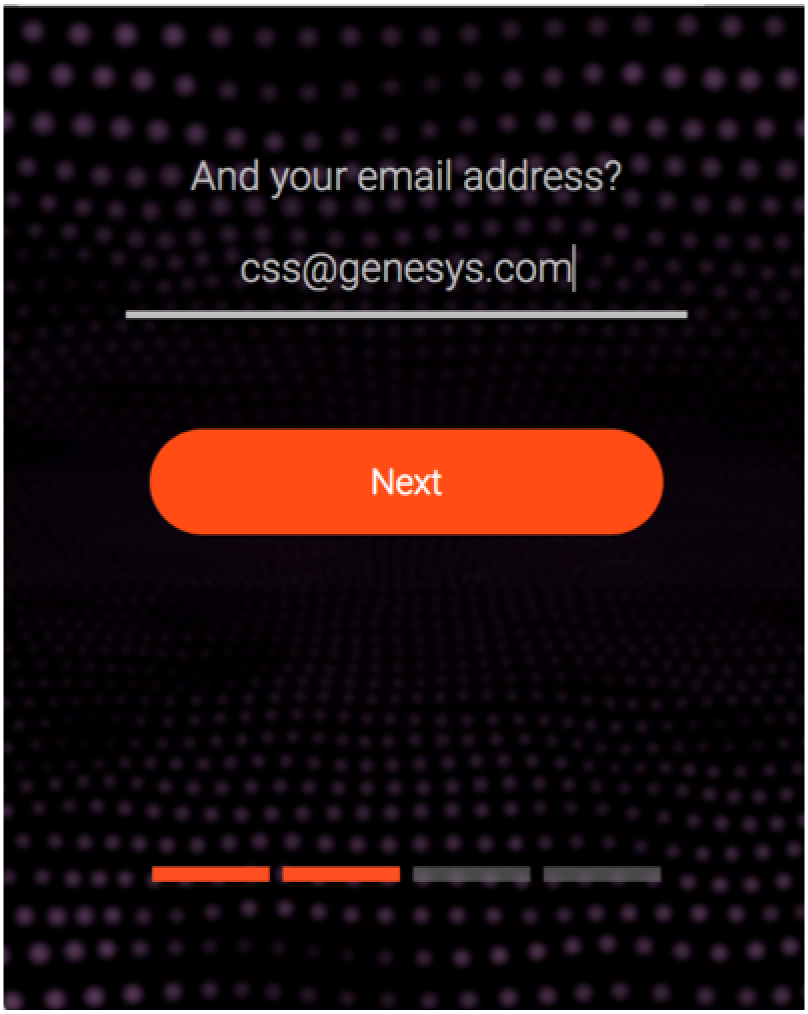


Hey, I'm Kate!

Please tell me your full name

CSS Guru

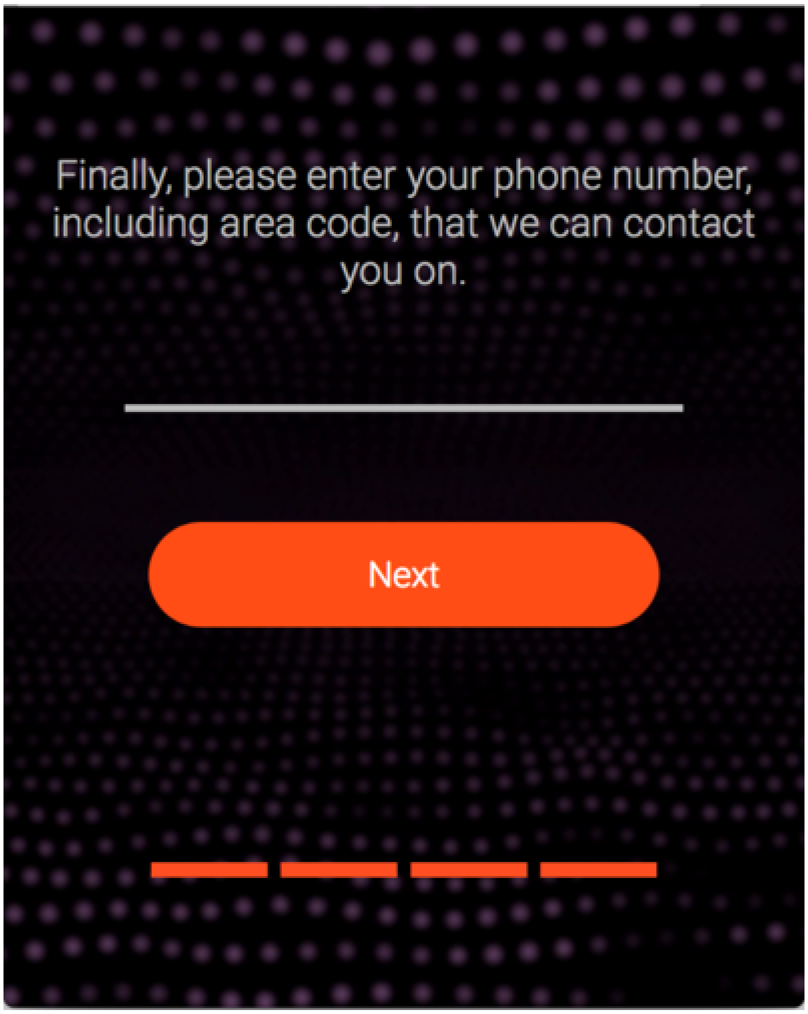
- You can use blocks and prompts in inventive ways. In the preceding and following graphics, you can see a progress bar appears beneath the button to guide the user through the MicroApp. You can use this approach by applying an image to a block that is only present at a certain step along the user journey.



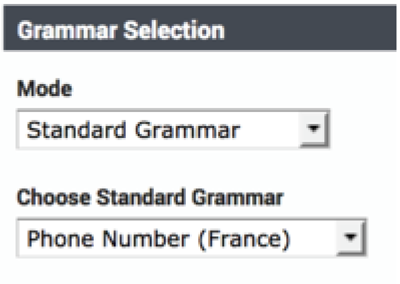
And your email address?

css@genesys.com

- The example above requests an email address. This uses Grammar Builder, which renders an input box for user entry. It is styled through CSS to give a simple underline, but it also uses Grammar Builder to supply regular expressions to ensure only valid email addresses are supplied. Regular expressions can validate a wide variety of user inputs.
- Drop-down options can use CSS to provide clean styling that matches the rest of the design. You can provide additional icons, or apply base64 encoding in the CSS.
- You can use regular expressions to validate user input, or you can use one of the validators that come with GAAP.



Finally, please enter your phone number, including area code, that we can contact you on.



Grammar Selection

Mode

Standard Grammar

Choose Standard Grammar

Phone Number (France)

Selector reference

This section lists the various elements you can call in your MicroApp.

Selector	Use
block	Assigned to the div that holds the content of the

Selector	Use
	block.
block_<block-name>	Assigned to the same element as **block**, but allows tweaking of the style on a block-by-block basis.
blocks-container	Assigned to the div that holds all blocks. More than one block can be rendered inside a **blocks-container**, such as two **Message** blocks followed by a **Question** block.
body	Assigned to the body element for the entire window.
close-footer-button-container	Assigned to the div inside the footer that contains the **End** button. This button ends the session.
currency-symbol-indent	Assigned to the label that holds the currency symbol if a **Question** block requests monetary input.
date-picker-control	Assigned to the anchor tag that holds the date icon, which the user clicks to open the date-picker window.
default-menu-option	Assigned to all buttons that appear in the footer.
footer	Assigned to the div that contains all footer elements. This element is the parent of **close-footer-button-container** and **footer-buttons**.
footer-buttons	Assigned to the div that holds all buttons for default options. For example, if a general **Agent** global menu is set, it appears as a div inside this element (though it is assigned **menu_option_<option-name>**).
indent-right	Frequently used on container div elements to keep the content separated from the left border of the window.
message-prompt	Assigned to div elements that house prompts used for **Message** blocks.
menu-help-prompt	Assigned to the div that contains the **Menu** or **Question** block Help prompt.
menu-initial-prompt	Assigned to the div that contains the **Menu** or **Question** block Initial prompt.
menu-retry-prompt	Assigned to the div that contains the **Menu** or **Question** block Retry prompt.
menu-option	Assigned to all menu option div elements.
menu_option_<menu-choice>	Assigned to all menu options, but with the option appended to the end, thus allowing control over the styling of specific options.
menu-options-container	Assigned to the div that surrounds all menu options.
menu-prompts-container	Assigned to the div that surrounds all **Menu** or **Question** block prompts.
middle	Assigned to the div that allows some control over

Selector	Use
	the central elements.
outer	Assigned to the div that allows some control over the central elements. Encompasses the **middle** element.
persona_<persona-name>	Additional class assigned to indicate the persona. If the default persona is active, no content appears after the underscore character. This allows styling control at a persona level. For example, **persona_** is assigned on blocks that have no explicit persona set (the default). However, you might assign **persona_italy** if an Italian persona has been created and made active.
prompt-<prompt-ref>	Assigned to all div elements that contain prompts. This allows individual styling on a prompt-by-prompt basis.
question-input-container	Assigned to the div that contains all **Question** block elements, regardless of the input type (example: date, credit card, currency, and so on).
question-input-field	Assigned to the div that is used for general input (example: natural numbers, phone number, and so on). Note: Server-side validation still occurs based on the grammar type. Failed validation results in hiding the Initial prompt and showing the Retry prompt.
recording-button	Assigned to the div that controls recording.
recording-done-button-wrapper	Assigned to the div that wraps both the recording functionality, as well as the **Next** button.
submit-button	Assigned to any div that is acting as a form of **submit** type. Can be used to proceed on **Menu** block options whereby the number of items is beyond the value configured for button rendering, or for **Question** and **Recording** blocks.
switch-prompts	Assigned to the anchor element that allows switching between Initial and Help prompts.
title-bar	Assigned to the div that holds the title bar area.
title-bar-text	Assigned to the div that holds the text for the title bar.
validationErrorFrame	Applied to the div on which the input failed.
validation-message	Applied to a span element that is added if validation failed.

Using Multimodal Communication

> **Important**
>
> This feature requires Orchestration Server 8.1.4.

Multimodal communication allows you to take advantage of various communication channels in a single interaction to enhance the customer experience.

For example, a customer might call your company to enquire about a product you sell. Before getting to an agent, you want to ask the caller for his or her email address so you can send more information about the product. However, it's not practical to have the customer enter an email address in a traditional IVR environment - it is much easier to have the customer type this information using a mobile app on a smartphone. Genesys Intelligent Automation can take advantage of both methods in a single interaction to best serve the customer.

Here's how it works:

- The customer calls your company.
- An Intelligent Automation application answers the call, identifies the caller, and begins routing the caller to the right department.
- When prompted for an email address, the application sends the customer a SMS message to open a mobile app on a smartphone.
- The customer opens the link and provides the information. The call continues in the background, with the Intelligent Automation application guiding the customer throughout the entire interaction.
- Once the information is captured, the customer returns to the phone call without any interruption in service.

> **Important**
>
> When a call enters multimodal mode, **Message** block prompts appear in the WebIVR application but these prompts are not read aloud by the IVR application. Only **Menu** or **Question** block prompts are read aloud by the IVR application while in multimodal mode.

Preparing your environment

Integrating with ORS

Use Genesys Administrator to create a Routing Point to point to a new Orchestration script.

1. Log in to Genesys Administrator.
2. Go to **Provisioning > Routing/eServices > Orchestration.**
3. Click **New...**.
4. In the **Configuration** tab, configure the following:
   - **Name** - Enter a name for this object. For example, `IntelligentAutomation_Multimodal`.
   - **Script Type** - Select **Enhanced Routing**.
   - **URI** - Enter the URL of the Intelligent Automation VUI server or VUI Load Balancer to use for multimodal communication. For example:

     ```
     http://<host:port>/fish-vui/ors/
     MultiModalContainer.jsp?gap_testsiteid=<#>⪆_istestcall=<boolean>⪆_url=<url_including_protocol_and_port>
     ```

     Where:

     - `<host:port>` specifies the URL for the Intelligent Automation VUI server.
     - `gap_testsiteid=` specifies the ID number of the Intelligent Automation application.
     - `gap_istestcall=` specifies whether this is a test call (`true` or `false`).
     - `gap_url=` specifies the URL for the Intelligent Automation VUI server.
5. Click **Save**.
6. In **Switching** > **Switches** > **SIP_Switch** > **DNS** > **Routing Point**
   - Click **New...**
   - In the **Configuration** tab,
     - Enter the DID number and select type as **Routing Point**.
     - Under **Routing & Orchestration**, add the orchestration application created during ORS integration.
   - Click **Save**.
7. In Orchestration, within **DN** tab, add the created DN and save it.

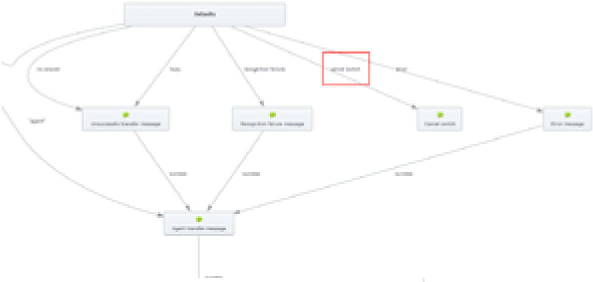
Configuring Intelligent Automation

The following VUI preferences are available in Intelligent Automation installations after the 3.5.x release. For releases before 3.5.x that were upgraded to more recent versions, you must add these preferences manually or import a new standard application template.

Setting	Description	Valid values
Standard ‘Visual Switch’ exit read-only DTMF	Specifies the DTMF key that callers can press to exit multimodal operation and resume IVR-only or voice mode. This is particularly useful for iPhone callers, as these users cannot click **Close** to end the WebIVR application.	Any DTMF key. For example: *****.

Setting	Description	Valid values
Standard ‘Visual Switch’ menu caller phone no. grammar	Specifies the grammar used to capture the caller's phone number for SMS.	Select a specific country, or **Digits** for general use.
Standard ‘Visual Switch’ menu caller phone no. variable	Specifies the variable against which to store the caller's phone number. If this variable is pre-populated before the user enters multimodal mode, Intelligent Automation does not invoke the SMS phone number question and grammar.	String
Standard ‘Visual Switch’ menu mode	Specifies the multimodal mode to use for the current block.	• **IVR Only** - Use normal IVR operation with no multimodal functionality. Note: This value must not be used to switch back from multimodal mode. Instead, select **Visual Mandatory Off**. • **Visual Optional** - Allows the caller to choose whether to use IVR or multimodal. See **Standard ‘Visual Switch’ menu option DTMF** or **Standard ‘Visual Switch’ menu option synonyms** to set values for how the caller can change modes. • **Visual Mandatory On** - Forces use of multimodal. As a result, Intelligent Automation prompts the caller for a SMS phone number. • **Visual Mandatory Off** - Forces the interaction back to IVR only. Use this option in conjunction with **Visual Mandatory On** to specify when multimodal must begin and end. Important • Intelligent Automation checks this setting for Menu blocks only. You cannot switch modes with any other block. • When Visual mode switches back to IVR, it is

Setting	Description	Valid values
		not required to set **Visual Mandatory Off** under Default behavior as the switch happens automatically when the maximum number of retries limit is reached on the Visual IVR menu.
Standard ‘Visual Switch’ menu option DTMF	If **Standard ‘Visual Switch’ menu mode** is set to **Visual Optional**, this value specifies the DTMF key the caller must press to switch modes.	Any DTMF key
Standard ‘Visual Switch’ menu option synonyms	If **Standard ‘Visual Switch’ menu mode** is set to **Visual Optional**, this value specifies the ASR grammar used to switch modes when spoken by the caller.	Comma-separated list
Standard ‘Visual Switch’ menu option weight	If **Standard ‘Visual Switch’ menu mode** is set to **Visual Optional**, this value specifies the ASR weighting applied to the **Standard ‘Visual Switch’ menu option synonyms** spoken by the caller. This can be set accordingly to reduce the likelihood of accidental recognition of a mode switch.	From **-100** to **+100**
Standard ‘Visual Switch’ menu sms cancel dtmf	After Intelligent Automation sends an SMS to the user, the IVR plays in a loop while it waits for the caller to click the WebIVR link. The IVR only moves forward after the caller opens the WebIVR link. Important The **Prompt Ref: Standard visual switch sms wait prompt** can be modified to the specific dtmf in the application's **Prompt List**. An asterisk is present as the prompt by default.	Any DTMF key

Setting	Description	Valid values
	 This value specifies the DTMF key that the caller can press to cancel the loop and resume IVR-only mode. The *cancel switch* event is triggered when the caller presses the DTMF key. This is useful if the caller does not receive the SMS message.	
Standard ‘Visual Switch’ menu sms cancel synonyms	After Intelligent Automation sends an SMS to the user, the IVR plays in a loop while it waits for the caller to click the WebIVR link. The IVR only moves forward after the caller opens the WebIVR link. Therefore, this value specifies the ASR grammar item the caller can use to cancel the loop and resume IVR-only mode. This is useful if the caller does not receive the SMS message.	Comma-separated list
Standard ‘Visual Switch’ menu sms cancel weight	The ASR weighting applied to **Standard ‘Visual Switch’ menu sms cancel synonyms**. This is useful to reduce the likelihood of accidental recognition of a mode cancellation.	From **-100** to **+100**
Standard ‘Visual Switch’ menu sms wait prompt iterations	After Intelligent Automation sends an SMS to the user, the IVR plays in a loop while it waits for the caller to click the WebIVR link. The IVR only moves forward after the caller opens the WebIVR link. This value specifies the number of times the loop plays before resuming IVR-only mode. You can use this option in conjunction with **Standard ‘Visual Switch’ read-only timeout** to play the prompt only once and wait in silence until the link is clicked.	Integer
Standard ‘Visual Switch’ read-only max retries	After Intelligent Automation sends an SMS to the user, this	Integer

Setting	Description	Valid values
	value specifies the maximum number of times to reject invalid input before resuming IVR-only mode.	
Standard ‘Visual Switch’ read-only max timeouts	After Intelligent Automation sends an SMS to the user, this value specifies the maximum number of times to time out before resuming IVR-only operation.	Integer
Standard ‘Visual Switch’ read-only prompt mode	Specifies whether to use the voice persona or the visual persona for prompts.	**Voice** or **Visual**
Standard ‘Visual Switch’ read-only timeout	After Intelligent Automation sends an SMS to the user, this value specifies the length of time, in milliseconds, to wait before timing out.	Integer

Frequently asked questions

How do I force the user to switch to multimodal mode at a certain point in the callflow?

To force a switch to WebIVR mode:

- Select a Menu block to force a switch to WebIVR.
- Open the **Menu** block.
- Add a VUI preference named **Standard ‘Visual Switch’ menu mode** and set it to **Visual Mandatory On**.

To force a switch back to the phone call:

- Select a Menu block to force a switch to WebIVR.
- Open the **Menu** block.
- Add a VUI preference named **Standard ‘Visual Switch’ menu mode** and set it to **Visual Mandatory Off**.

Can I let the caller choose when and where to use WebIVR?

Yes. Add the following default VUI preferences:

- **Standard ‘Visual Switch’ menu mode** - Set to **Visual Optional**.

- **Standard ‘Visual Switch’ menu option DTMF** - Specify a DTMF key the caller must press to initiate the switch to WebIVR.
- **Standard ‘Visual Switch’ menu option synonyms** - Specify recognition phrases the caller must say to initiate the switch to WebIVR.

Do I have to send the SMS to the phone I'm calling from?

No. The number can be changed by the following steps:

- Add the VUI preference **Standard ‘Visual Switch’ menu caller phone no. variable** to a variable name.
- Use an additional **Question** block to confirm the destination number for SMS to send this IVR URL and update the variable to the required destination number.
- Add the VUI preference **Standard ‘Visual Switch’ menu caller phone no. grammar** if the destination number needs further amendments.
- Add the VUI preference '**Standard ‘Visual Switch’ menu** mode. This will now send SMS to the required destination.

Important

The phone number must be confirmed in a separate menu block before the block that invokes the multi-modal menu block.

The IVR reads back the text from the WebIVR while in multimodal mode. Can I change this?

Yes. The VUI preference **Standard ‘Visual Switch’ read-only prompt mode** indicates which persona set is used by the IVR when in multimodal mode.

The default value, **Visual**, allows a single callflow to support an optional visual switch. If a caller is on the phone, he or she hears prompts that are relevant to a phone call. If the caller switches to WebIVR, the Visual persona prompts are read back by the IVR.

However, should a particular callflow only need to support multimodal mode (in other words, you are forcing the caller to switch to WebIVR), the value can be set to **Voice**.

I didn't receive a text. Must I wait for the IVR to time out before getting out of the loop?

No. If you didn't receive the text, you can use the values of VUI preferences **Standard ‘Visual Switch’ menu sms cancel dtmf** and **Standard ‘Visual Switch’ menu sms cancel synonyms** to cancel the wait and move back to normal IVR mode.

If the IVR is in read-only mode and I lose connectivity, how can I return to IVR-only mode?

You can configure the VUI preference '**Standard ‘Visual Switch’ exit read-only DTMF** to specify a DTMF key that cancels multimodal mode. Any other DTMF key triggers a retry and, assuming the prompt is configured appropriately, reminds the caller of the only applicable DTMF key at that time.

I don't want the IVR to repeat the prompts in read-only mode. How can I achieve this?

Set the VUI preference **Standard ‘Visual Switch’ read-only timeout** to a large value to offer a larger *silence* time between prompts.

When using multimodal mode and the IVR is in read-only mode, why are only Menu and Question prompts spoken to the caller?

This is a limitation that will be addressed in a later release.

Where are the prompts that relate to multimodal functionality?

They are in the standard prompts of the inbound application, named:

- Standard visual goodbye
- Standard visual switch back to ivr
- Standard visual switch callback help prompt 1
- Standard visual switch callback help prompt 2
- Standard visual switch callback initial prompt 1
- Standard visual switch callback initial prompt 2
- Standard visual switch callback retry prompt 1
- Standard visual switch callback retry prompt 2
- Standard visual switch callback timeout prompt 1
- Standard visual switch callback timeout prompt 2
- Standard visual switch enter phoneno help prompt
- Standard visual switch enter phoneno initial prompt
- Standard visual switch enter phoneno retry prompt
- Standard visual switch enter phoneno timeout prompt
- Standard visual switch sms cancel confirmation prompt
- Standard visual switch sms sent prompt
- Standard visual switch sms system cancelled prompt
- Standard visual switch sms user cancelled prompt
- Standard visual switch sms wait prompt
- Standard visual switch transfer confirm help prompt
- Standard visual switch transfer confirm initial prompt
- Standard visual switch transfer confirm retry prompt
- Standard visual switch transfer confirm timeout prompt

How can I configure the WebIVR title bar?

Specify a new value for **Standard visual title bar message prompt**.

How can I configure the WebIVR **End** and **Back to Voice** buttons?

Specify new values for **Standard visual back to voice button prompt** and **Standard visual end button prompt**.

Can I transfer to an agent from a multimodal call?

Currently, Genesys Intelligent Automation only supports bridged transfers, which means the Genesys Intelligent Automation session is kept open for the duration of the transfer. However, your Genesys representative can implement custom transfer logic in order to support other kinds of transfers.

Using Bots

Genesys Intelligent Automation can provide your customers with chat bot-based access to your WebIVR applications. Customers can converse with a bot using natural language to accomplish various business needs, such as making a payment or checking an account balance.

> ### 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.

The following components are required:

- Digital Messaging Server 9.0.x - Provides the bot platform.
- Genesys Administrator - Used for configuring the chat bot client.
- Interaction Routing Designer - Provides the routing strategy.

> ### Important
> For SMS (Short Message Service) functionality, you must also install Social Messaging Server 8.5.x with SMS plugin.

> ### Important
> To test your bot functionality using Virtual Call, please set the **Channel** field to `WEBCHAT` when starting your Virtual Call session..

Once installed, the bot acts similarly to a WebIVR application using prompts from the associated visual persona. However, if using SMS, remember that customers cannot use special characters in response to queries from the bot.

Preparing the chatbot environment

Install Digital Messaging Server

1. Refer to the Digital Messaging Server Guide for information on how to install Digital Messaging Server.
2. Copy the chatbot jar file to the Digital Messaging Server subfolder for chatbots. For example: **C:\Program Files\GCTI\eServices 9.0\Digital Messaging Server\DMS\media-channel-drivers\channel-chatbot\bots-repo\**.

Use Genesys Administrator to configure Digital Messaging Server

1. Open Genesys Administrator and log in to your configuration environment.
2. Go to **Provisioning > Environment > Applications** and open the Digital Messaging Server object.
3. Create a section called `channel-chatbot-monitor-bots`.
4. Create a chatbot option. You must ensure the option name exactly matches the name of the jar file that you previously uploaded to Digital Messaging Server. For example: `fish-cbp-9.0.0.jar`.
5. Set the value of the chatbot option to point to the load balancer for VUI Server. You can also add the parameters **downstream_request_timeout_millis** and **session_timeout_minutes** to set timeout values. In the following example, the request timeout is **120000** milliseconds (120 seconds) and the session timeout is **30** minutes:

```
{ "base_url" : "http://www.yourcompany.com:8081/fish-vui", "downstream_request_timeout_millis" : 120000, "session_timeout_minutes" :
30}
```

6. Click **Save & Close**.

Important

Starting from 9.0.109.00 release, IA provides support for proxy servers. The following optional parameters can be passed to the **fish-cbp** jar file:

- **x_vui_proxy_host** - The proxy server URL or IP address
- **x_vui_proxy_port** - The port number
- **x_vui_proxy_username** and **x_vui_proxy_password** - The credentials to access the proxy server. Required only when the proxy server requires authentication.

Example:

```
{ "base_url" : "http://www.yourcompany.com:8081/fish-vui",
"downstream_request_timeout_millis" : 120000, "session_timeout_minutes" : 30,
"x_vui_proxy_host" : "proxy_server_url", "x_vui_proxy_port" : "port_number",
"x_vui_proxy_username": "username", "x_vui_proxy_password": "password"}
```

If the proxy server information is not configured or are empty, the VUI server will be accessed directly.

Starting from 9.0.109.02 release, IA supports configuring the default error message that is displayed when the connection between the Chat Bot Platform (CBP) and VUI fails. The new optional JSON entity, **x_vui_default_error_msg**, can be used to configure the message.

```
{"base_url" : "http://www.yourcompany.com:8081/fish-vui",
"downstream_request_timeout_millis" : 120000, "session_timeout_minutes" : 30,
"x_vui_proxy_host" : "proxy_server_url", "x_vui_proxy_port" : "port_number",
"x_vui_proxy_username": "username", "x_vui_proxy_password": "password",
"x_vui_default_error_msg" : "Sorry, an error has occurred"}
```

(Optional) Use Genesys Administrator to configure SMS Server

Important

- Use this section only if you want to provide SMS (Short Message Service) access to your chatbot. Otherwise, go to the next section.
- Refer to SMS Server (part of eServices) documentation for more information on SMS Server.
- This feature uses the *Session Mode* of SMS Server, which refers to creating and keeping an interactive conversation between a mobile client and an agent in the form of a conventional chat session. All messages received and sent during this session are associated with one interaction, which corresponds to this SMS session.

1. Open Genesys Administrator and log in to your configuration environment.

2. Go to **Provisioning > Environment > Applications** and open the SMS Server object.
3. Open the channel that you want to use with the bot.
4. For the option **inbound-route**, specify the access point in your routing strategy that is used to place submitted interactions for incoming messages.
5. Click **Save & Close**.

Use Interaction Routing Designer to configure the chatbot

Once you have installed the bot interface, you can use Interaction Routing Designer to configure settings.

1. Open Interaction Routing Designer and log in to your environment.
2. Open the routing strategy that is triggered by the *session* endpoint, and add an **External service** block to the start of the flow.
3. Open the **External service** block and go to the **General** tab. Configure the settings as described below:
   - **Application type** - Set to the Digital Messaging Server or Social Messaging Server in your environment.
   - **Application name** - Set to the Digital Messaging Server or Social Messaging Server in your environment.
   - **Service** - Set to `ChatBotPlatform`.
   - **Method** - Set to `StartBot`.

   In the **Parameters** section, add the following:
   - **Nickname** - The bot name displayed to customers.
   - **SiteID** - The ID number of your Genesys Intelligent Automation application to use with the bot.
   - **IsTestCall** - If **true**, the test version of the application is used. Otherwise, set to **false** to use the production version.
   - **AuthToken** - The value of the **Authentication Key** field in your company page.
   - **StopBotOnAgentArrival** - Set to **true** if you want the bot to end once an agent joins the session. Otherwise, set to **false**.
   - **StopBotOnCustomerLeft** - Set to **true** if you want the bot session to end once a customer leaves the session. Otherwise, set to **false**.
   - **Visibility** - Set to **ALL**.
   - **ChatBotID** - Set to `speechstorm-chatbot`.
   - **ChatBotName** - Do not enter information in this field.
   - **_umsChannel** - Set to `channel-chatbot`.
4. Click **OK**.

Genesys recommends that you add new blocks to your routing strategy, just after the **External service** block, to pause the strategy until the chatbot has completed its task. Otherwise, the routing strategy might queue up the session for an agent and, as soon as an agent joins, the **StopBotOnAgentArrival** parameter will cause the bot to terminate early.

To determine if a chatbot interaction is still alive, Genesys Intelligent Automation checks the following interaction data values:

- **IsOnline** - If the value is **0**, the bot is offline and you can terminate the interaction.
- **FishTransferRequested** - If the value is **true**, you can proceed to route the interaction to an agent.

Refer to the Interaction Routing Designer documentation or consult your Genesys representative for information on the best routing strategy for your environment.

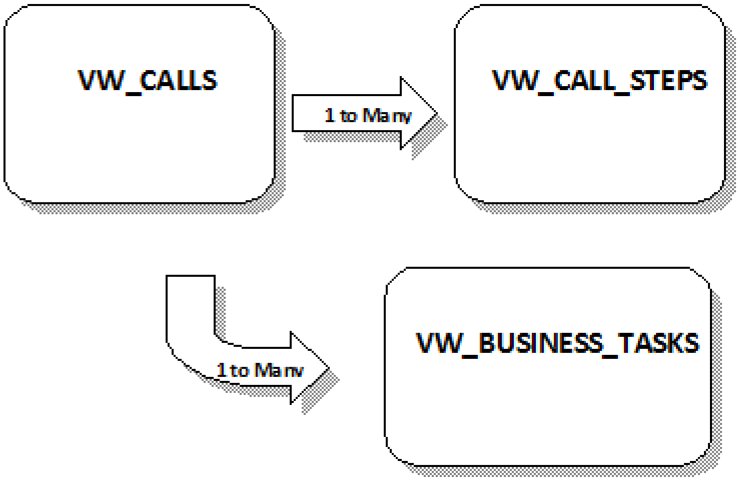
Database Views Schema

Call Reporting Database Views

The three main call reporting views within the reporting database are:

- VW_CALLS - Records each call made into the Genesy Intelligent Automation application.
- VW_CALL_STEPS - Records each step made by the caller.
- VW_BUSINESS_TASKS - Records which business tasks started and ended, along with details of each instance.

The diagram below describes the relationship between the three views:



These views are configured so they will not lock the database when they run. Genesys recommends that you always filter based on:

- **company_id**.
- **is_test_call**.
- **call_start_date**.

Refer to the Useful SQL Queries section to find queries that might be useful when working with the database

VW_CALLS

This view records each *call* made into the Intelligent Automation application.

Calls might be physical or not. For example, if a caller started a phone call to Intelligent Automation but was transferred out of Intelligent Automation to a routing strategy and then returned. In this case, two rows are recorded into this view.

Column	DataType	Length	Description	Example
call_id	bigint		Unique identifier for each call. This is an incrementing number.	10001
company_id	int		Link to Intelligent Automation company identifier. The company ID can be found in the GUI (look in **Administration > Companies**).	3
voice_platform_session_id	nvarchar	100	Media Platform Identifier for the call within Intelligent Automation (for example, Genesys Voice Platform).	26A92695-9557-444A-A9B2-CCB4D71C1B69-1791
voice_platform_full_call_id	nvarchar	100	Unique identifier to link multiple Intelligent Automation interactions together within a single phone call.	10KMMSG8LL37TETGHTNQOF4KK80001NV
start_site_id	int		The module ID of the Inbound application from which the call started.	1
start_site_name	nvarchar	100	The name of the Inbound application from which the call started.	SIM Activation
call_start_time	datetime		The timestamp for when the call started.	2014-02-13 15:35:31.737
call_start_date	datetime		The date timestamp for when the call started (time is always **00.00.00.000**).	2014-02-13 00:00:00.000
call_start_hour	int		The hour of the day for when the call started, in 24-hour format.	15

Column	DataType	Length	Description	Example
call_end_time	datetime		The timestamp for when the call ended.	2014-02-13 15:36:02.640
call_end_date	datetime		The date timestamp for when the call ended (time is always **00.00.00.000**).	2014-02-13 00:00:00.000
call_end_hour	int		The hour of the day when the call ended, in 24-hour format.	15
call_end_site_id	int		The ID of the last module accessed on the call.	3
call_end_site_name	nvarchar	100	The name of the last module accessed on the call.	Transfer to RS – With Announcement
call_end_block_type	tinyint	1	Type of the last block accessed on the call. Possible values are: • **1** – Start. • **2** – End. • **3** – Script. • **4** – Message. • **5** – Menu. • **6** – Custom Question. • **7** – Phone Transfer. • **8** – URL Transfer. • **9** – Recording. • **10** – Interceptor.	2

Column	DataType	Length	Description	Example
call_end_block_name	nvarchar	100	The name of the last block or step accessed on the call.	End Call
call_end_result	nvarchar	100	The resulting outcome description. Possible values are: • **error**. • **success**. • **hangup**. • **system hangup**.	**hangup**
has_recent_failure	bit	1	Internal flag to determine if an error or recognition failure occurred in the callflow step/ block. This flag can be reset, as controlled by VUI preferences, if the succeeding Menu or Custom Question step/block is successful.	0
is_test_call	bit	1	Flag to determine if call is for the test or production version of the module. Values can be **1** for Test or **0** for production.	1
call_duration	int		Call duration in seconds.	30
cli	nvarchar	45	Calling Line Identifier (the number that the caller is calling from).	3100
dnis	nvarchar	45	Dialed number that is associated to the Intelligent Automation application.	1234

Column	DataType	Length	Description	Example
cluster_id	int		The ID of the Intelligent Automation server cluster that handled the call. Cluster information can be found in the Intelligent Automation GUI (look in **Administration > Clusters**).	1
cluster_name	nvarchar	100	The name of the Intelligent Automation server cluster that handled the call. Cluster information can be found in the Intelligent Automation GUI (look in **Administration > Clusters**).	Default Voice Cluster
cti_fields	nvarchar	1000	Computer Telephony Integration (CTI) attached data. Fields are pipe separated, with each field represented as a key-value pair separated by colon. Literal colons, pipes, or percent symbols within keys or values are represented as **%c**, **%p**, or **%%**, respectively. You can define the list of allowed CTI variables that are included in this field in the Intelligent Automation GUI (look in **Administration > Default Server Settings**).	Segment:Gold
last_menu_block_type	tinyint	1	The type of the last Menu or Custom Question block accessed on the call. Possible values are:	5

Column	DataType	Length	Description	Example
			• **5** – Menu • **6** – Custom Question	
last_menu_block_name	nvarchar	100	The name of the last Menu or Custom Question block accessed on the call.	If the callflow visited the following blocks: Welcome Message > Proceed With Activation Menu >Process Request Script > Successful Message > End Call This value is **Proceed With Activation Menu**.
cli_type	smallint	1	Representation of whether the caller is using a landline or a mobile handset. Possible values are: • **0** - Unknown • **1** - Landline • **2** - Mobile The list of CLI mobile number prefixes that are defined in server settings is used to determine if the CLI is a mobile number.	1
server_id	int		The ID of the Intelligent Automation server that handled the call. Server information can be found in the Intelligent Automation GUI (look in **Administration > Servers**).	1

Column	DataType	Length	Description	Example
start_channel	int		The channel in which the call started. Possible values are: • **0** - Unknown. • **1** – Voice. • **2** – Web. • **3** – Facebook. • **4** – Web with Voice. • **14** – Voicebot.	2

VW_CALL_STEPS

Each row in this view details a single block within the callflow that the caller progressed through.

Column	DataType	Length	Notes	Example
id	int		Unique call step identifier within call. This is an incrementing number.	1
call_id	int		See VW_CALLS.call_id.	10001
call_start_site	int		See VW_CALLS.start_site_id.	1
call_start_date	date		See VW_CALLS.call_start_date.	2014-02-13
cli_type	smallint	1	See VW_CALLS.cli_type.	1
company_id	int		See VW_CALLS.company_id.	3
is_test_call	bit	1	See VW_CALLS.is_test_call.	1
site_id	int		The ID of the Intelligent Automation module where this block/step (in the callflow) belongs.	265
site_name	nvarchar	100	The name of the Intelligent Automation module where this block/step (in the callflow) belongs.	Call Initialization
is_submodule	bit	1	Flag to determine if the module is flagged as an Inbound application or just a module.	1
block_type	tinyint	1	Block type indicator. Possible values are: • **1** – Start • **2** – End • **3** – Script	1

Column	DataType	Length	Notes	Example
			- **4** – Message - **5** – Menu - **6** – Custom Question - **7** – Phone Transfer - **8** – URL Transfer - **9** – Recording - **10** – Interceptor	
block_name	nvarchar	100	Name of the block or step in the callflow.	Start
block_detail	nvarchar	500	Internal field used to store additional information (if any) about the block.	For example, if this is a URL Transfer block, this field will show the module it will transfer to. (**Link to module : 5**)
start_time	datetime		Timestamp for when the callflow step/block was first visited.	2014-02-13 15:35:34.770
start_date	datetime		Date timestamp for when the callflow step/block was first visited (time is always **00.00.00.000**).	2014-02-13 00:00:00.000
start_hour	int		Hour of the day when the callflow step/block was first visited, in 24-hour format.	15
end_time	datetime		Timestamp for when the callflow step/block ended.	2014-02-13 15:35:34.780
end_date	datetime		Date timestamp for when the callflow step/block ended	2014-02-13 00:00:00.000

Column	DataType	Length	Notes	Example
			(time is always **00.00.00.000**).	
end_hour	int		Hour of the day for when the callflow step/block ended, in 24-hour format	15
duration	int		Duration, in seconds, spent within the step/block.	0
result	nvarchar	100	Resulting outcome description. Possible values are: • **error**. • **success**. • **hangup**. • **system hangup**.	Success
result_detail	nvarchar	100	Additional information relating to the result (for example, **transferred telephone number**).	tel://123456789
error_messages	nvarchar	500	Detailed error messaging (if any)	
wav_filename	nvarchar	200	Only applicable for a Recording block. This is the filename of the saved recording.	temprecording_123456.wav
is_recording_saved	bit	1	Only applies for a Recording block. This is the flag to determine if there is a wav file recording saved. Possible values are **0** if no recording is saved or **1** if a recording	1

Column	DataType	Length	Notes	Example
			exists.	
recognition_type	int		Internal recognition type. Possible values are: • **0** – None • **1** – Menu • **2** – Custom • **3** – Defaults • **4** - Global	1
is_dtmf	bit	1	Flag that indicates if block/step is DTMF enabled. Value will be set to **1** if its DTMF enabled; otherwise, this value is **0**.	1
num_retries	tinyint		Count of no-match entries by caller in this callflow step/block. This field populates only if the caller leaves this block (i.e. doesn’t hang up).	2
num_timeouts	tinyint		Count of no-input entries by caller in this callflow step/block. This field populates only if the caller leaves this block (i.e. doesn’t hang up).	1
num_helps	tinyint		Number of times the *help* command was used in this callflow block/step. This field populates only if the caller leaves this block (i.e. doesn’t hang up).	1

Column	DataType	Length	Notes	Example
num_repeats	tinyint		Number of times the *repeat* command was used in this callflow block/step. This field populates only if the caller leaves this block (i.e. doesn’t hang up).	1
num_recovery_attempts	int		Number of times the callflow step/block was visited when the caller failed to be recognized and took the Intelligent Automation recovery route.	1
num_nbest	tinyint		Number of best possible matches (nbest) recognized in Automatic Speech Recognition (ASR). For an answer provided via DTMF, this will always have a value of **1**.	10
nbest_meaning_1	nvarchar	45	First highest match from the ASR against the SRGS grammar in context.	07712344401
nbest_rawanswer_1	nvarchar	100	First highest synonym match from the ASR against the SRGS grammar in context.	Oh seven seven one two three four four four oh one
nbest_confidence_1	int		Confidence scoring out of 1000 of first highest match against the SRGS grammar in context. For DTMF, this value is always **1000**.	700
nbest_slots_1	nvarchar	100	First highest slot content from the ASR against the SRGS grammar in context.	Type:Mobile|Number:12344401
nbest_recognition_type_1	int		Internal recognition type	2

Column	DataType	Length	Notes	Example
			reference for the first highest match. Possible values are: • **0** – None • **1** – Menu • **2** – Custom • **3** – Defaults • **4** - Global	
nbest_meaning_2	nvarchar	45	Second highest match from the ASR against the SRGS grammar in context.	07712344501
nbest_rawanswer_2	nvarchar	100	Second highest synonym match from the ASR against the SRGS grammar in context.	Zero seven seven one two three four four five oh one
nbest_confidence_2	int		Confidence scoring out of 1000 of second highest match against the SRGS grammar in context. For DTMF, this value is always **1000**.	10
nbest_slots_2	nvarchar	100	Second highest slot content from the ASR against the SRGS grammar in context.	Type:Mobile|Number:12344501
nbest_recognition_type_2	int		Internal recognition type reference for the second highest match. Possible values are:	2

Column	DataType	Length	Notes	Example
			- **0** – None - **1** – Menu - **2** – Custom - **3** – Defaults - **4** - Global	
nbest_meaning_3	nvarchar	45	Third highest match from the ASR against the SRGS grammar in context.	07712345401
nbest_rawanswer_3	nvarchar	100	Third highest synonym match from the ASR against the SRGS grammar in context.	Zero seven seven one two three four five four zero one
nbest_confidence_3	int		Confidence scoring out of 1000 of third highest match against the SRGS grammar in context. For DTMF, this value is always **1000**.	10
nbest_slots_3	nvarchar	100	Third highest slot content from the ASR against the SRGS grammar in context.	Type:Mobile|Number:12345401
nbest_recognition_type_3	int		Internal recognition type reference for the third highest match. Possible values are: - **0** – None - **1** – Menu	2

Column	DataType	Length	Notes	Example
			• **2** – Custom • **3** – Defaults • **4** - Global	
output_node_name	nvarchar	100	Name of the path that leads to this callflow step/block.	success
sequence_in_call	int		Given the list of callflow steps/blocks that were visited within the call, this is the position this step/block was visited within the sequence.	3
sequence_in_site	int		Given the list of callflow steps/blocks that were visited within the module, this is the position this step/block was visited within the sequence.	1
persona_name	nvarchar	100	The name of the persona active during the current call step. An empty string is saved if using the default persona.	French
channel	int		The channel that is being used by the caller during this call step. Possible values are: • **0** – Unknown • **1** – Voice • **2** – Web	2

Column	DataType	Length	Notes	Example
			• **3** – Facebook • **4** – Web with Voice • **14** – Voicebot.	

VW_BUSINESS_TASKS

Each row in this view details the business task that was processed (started, ended, and so on) within the callflow that the caller progressed through.

Column	DataType	Length	Notes	Example
id	bigint		Unique business task identifier within the call. This is an incrementing number.	10008
call_id	int		See VW_CALLS.call_id.	10001
voice_platform_session_id	nvarchar	100	See VW_CALLS.voice_platform_session_id.	26A92695-9557-444A-A9B2-CCB4D71C1B69-1791
voice_platform_full_call_id	nvarchar	100	See VW_CALLS.voice_platform_full_call_id.	10KMMSG8LL37TETGHTNQOF4KK80001NV
company_id	int		See VW_CALLS.company_id.	3
is_test_call	bit	1	See VW_CALLS.is_test_call.	1
start_site_id	int		The ID of the module where this business task belongs.	10
start_site_name	nvarchar	100	The name of the module where this business task belongs.	Payment By Full Balance
start_time	datetime		Timestamp for when this business task started.	2014-02-13 15:36:31.367
start_date	datetime		Date timestamp for when this business task started (time is always **00.00.00.000**).	2014-02-13 00:00:00.000
start_hour	int		Hour of the day for when this business task started, in 24-hour format.	15
end_time	datetime		Timestamp for when this business task ended.	2014-02-13 15:36:50.367
end_date	datetime		Date timestamp for when this business task ended (time is always **00.00.00.000**).	2014-02-13 00:00:00.000

Column	DataType	Length	Notes	Example
end_hour	int		Hour of the day for when this business task ended, in 24-hour format	15
duration	int		Duration, in seconds, between when the time business task started and ended.	19
name	nvarchar	100	Name of the business task (for example, **postal address lookup**).	Payment
outcome_category	tinyint	1	Outcome category identifier. Possible values are: • **1** – success • **2** – failure • **3** – CPFL (customer perceived failure) • **4** – Unknown • **5** - Hangup	3
outcome_description	nvarchar	100	Outcome category description.	Invalid balance
details	nvarchar	100	Additional information regarding the business task outcome.	Balance = null
call_start_site	int		See VW_CALLS.start_site_id.	1
call_start_date	date		See VW_CALLS.call_start_date.	2014-02-13

Useful SQL Queries

This page describes SQL queries that might be useful when working with the database.

Get complete call details for a given call ID

```
SELECT
calls.*, call_steps.*
FROM VW_CALLS calls
INNER JOIN VW_CALL_STEPS call_steps
ON calls.call_id = call_steps.call_id
AND calls.call_id = xxx
```

Get complete business task details for a given call ID

```
SELECT
calls.*, business_tasks.*
FROM VW_CALLS calls
INNER JOIN VW_BUSINESS_TASKS business_tasks
ON calls.call_id = business_tasks.call_id
AND calls.call_id = xxx
```

Get complete call details for a company, filtered by call date and module version

```
SELECT
calls.*, call_steps.*
FROM
VW_CALLS calls,
VW_CALL_STEPS call_steps
WHERE
        calls.company_id = call_steps.company_id
        AND calls.company_id = 1
        AND calls.is_test_call = call_steps.is_test_call
        AND calls.is_test_call = 0
        AND calls.call_start_date = '2017-01-01'
```

Call Processing

This page describes the logic applied at the end of calls. It also describes associated backlog processing, if necessary.

End-of-call logic

When a call ends, Genesys Intelligent Automation writes the call records to the configured database. A call ends under the following conditions:

- Intelligent Automation receives a hang-up event from MCP.
- There are no more blocks to process in the callflow.
- The call reaches the **End Call** block (can be configured either to disconnect or return to strategy).
- The call reaches a **Transfer** block.
- Session timeout due to no MCP requests during the configured timeout period.

  The **<session-timeout>** parameter value, found in the application **web.xml** file, defines this timeout period in seconds. In the example below, the session timeout period is set at 30 seconds:

```
<session-config>
 <session-timeout>30</session-timeout>
 </session-config>
```

Important

The **<session-timeout>** parameter resets to the default value each time you upgrade Intelligent Automation. Therefore, you must update this value after each upgrade if you do not want to use the default value.

Backlog processing

Important

Backlog processing of calls only occurs if the **Backlog.Processor.Enabled** setting is set to **true** and the hive-off process is not running. Backlog processing is paused during the hive-off process.

When a call ends, Intelligent Automation attempts to write the call to the database. If this process fails, Intelligent Automation sends the call to backlog processing.

The table below, **Handled Database Write Errors**, describes reasons for why the write might fail. If the write fails for another reason that is not described in this table, the call is unrecoverable and its data is lost.

Handled Database Write Errors

Database timeout
Connection pool exhaustion
The SQL Exception from the JDBC driver matches one of the following: • 08 – Connection Error • 66 – Driver Error • HY – Operation Cancelled • S0001 – SQL Login Failed

Configuring backlog processing

You can configure the settings below to periodically check the the backlog folder size:

- **Backlog.DiskSpaceMonitor.CheckIntervalMillis** - Specify, in milliseconds, how often to check the size of the backlog folder.
- **Backlog.DiskSpaceMonitor.WarningUsedMB** - Specify, in megabytes, the size that the backlog folder must exceed before a WARNING-level SMTP alert is sent.
- **Backlog.DiskSpaceMonitor.MaxUsedMB** - Specify, in megabytes, the size that the backlog folder must exceed before a MAJOR-level SMTP alert is sent. This is the size that, once reached, data is lost.

As of the 3.5.100.04 release, you must configure the following settings to specify which call-records errors are sent to the backlog:

- **Backlog.TreatAllFailuresAsBackloggable** - If `true`, all possible errors are sent to the backlog. If `false`, you can specify which errors are backlogged by configuring the following settings:
  - **Backlog.BackLoggableErrorsList.ExactMatch** - Specify an error code that must be matched. For example, `S0001`.
  - **Backlog.BackLoggableErrorsList.StartsWith** - Specify a comma-separated list of prefixes for accepted error codes. For example, `08,66,HY`.

> Important
>
> If you set **Backlog.TreatAllFailuresAsBackloggable** to `true`, Intelligent Automation sends all records to the backlog, even if these records cannot be backlogged. This movement consumes bandwidth until the records are manually removed from the backlog directory on the disk.

Sending backlog items to the database

If processing continues, the call data is serialized to XML, saved in an XML file on disk, and added to the internal file backlog list. You can use the setting **Backlog.Processor.MinItemAgeBeforeProcessingMillis** to specify, in milliseconds, how long Intelligent Automation must wait before trying to re-insert an item from the backlog back into the database.

Multiple backlog processor threads process any items that are added to the backlog. The **Backlog.Processor.ThreadCount.CallHistory** server setting defines the number of threads to use. Consequently, this value is also the maximum number of concurrent database writes from the backlog, regardless of database pool availability.

> ### Important
> The backlog shares the connection pool with the main application. Therefore, pool exhaustion might result if the **Backlog.Processor.ThreadCount.CallHistory** server setting is set too high.

Scenario

The following provides an example of how backlog processing functions.

Consider the following scenario:

- Maximum Pool Size = **100**
    - Main Application Connection Usage = **95**
    - Backlog Thread Count = **10**

The total number of items is **105**, which is five more than the maximum pool size. The excess items are sent to the backlog.

> ### Important
> Intelligent Automation does not guarantee whether excess items come from the backlog or the main application. Therefore, you must provide some headroom on your maximum pool size configuration. You must also account for the number of VUI servers and gauge whether the database server can cope with the number of total connections configured per server.

Once items are in the backlog, each backlog processing thread tries to grab an item from the backlog and re-insert it into the database. If no backlog items exist, the processing threads sleep for 20 seconds. This sleep value is not configurable.

Once a thread finds an item to process:

1. It attempts to move the file into the processing folder.
2. It de-serializes the file contents back into call data and deletes the XML file.

> Important
>
> If an error occurs in one of the first two steps, the file moves permanently to the *failed* folder and is no longer accessible to the backlog processing threads. This is the only scenario by which a file is moved to the *failed* folder and abandoned. In other words, the *failed* folder is not used for calls that cannot be re-saved. In that case, a backlog thread continues to put the call into the work queue without limit on failed attempts (unless the error does not match one of the handled errors referenced above).

3. It attempts to re-save the call data. One of the following occurs:
   - If the re-save is successful, the thread moves onto the next item in backlog.
   - If the save fails but it is a handled error, the thread adds a new backlog item for the call data. The thread that tried to process the item sleeps for 60 seconds (this value is non-configurable). The new item is not processed by another thread until the value of **Backlog.Processor.MinItemAgeBeforeProcessingMillis** has passed.
   - If the save fails and it is an unhandled error (not part of the list of handled errors, call data is lost.

Notes

- There is no limit on the the number of times a backlog thread can fail to process an item. Each time it fails, the item is added to the backlog again.
- The same code executes each time a backlog thread attempts to re-save an item. Therefore, a thread in the reporting connection pool is used each time an attempt is made.
- The only server setting that you can change at runtime is **Backlog.Processor.Enabled**. All other settings require a restart of the server.

SNMP Traps

This page gives an introduction to SNMP traps and lists the traps that are currently available within Genesys Intelligent Automation.

Important

Intelligent Automation supports Simple Network Management Protocol (SNMP) v2c.

What are SNMP Traps?

SNMP Traps are a way of allowing systems (SNMP agents) to send asynchronous notifications to SNMP managers. These notifications are conditions that the SNMP managers should be aware of and allow operational departments within an organisation to react accordingly. Genesys Intelligent Automation uses the SNMP Traps protocol to send important notifications to SNMP managers.

The notifications that we send can be grouped under three categories:

- Server stop/start/heartbeat alarms
- Licensing alarms
- General/call-flow error alarms

A list of all traps being sent by Genesys Intelligent Automation is detailed later in this page. The traps that Genesys Intelligent Automation can generate are defined by an MIB file.

What are MIBs?

SNMP itself does not define which information a managed system should offer. Rather, SNMP uses an extensible design, where the available information is defined by management information bases (MIBs). MIBs describe the structure of the traps and use a hierarchical namespace containing object identifiers (OID). Each OID identifies a variable that can be read or set via SNMP.

Genesys Intelligent Automation traps are defined in the Genesys Intelligent Automation Management Information Bases (MIBs), which are available as part of the Genesys Intelligent Automation release.

Reacting to Genesys Intelligent Automation Traps

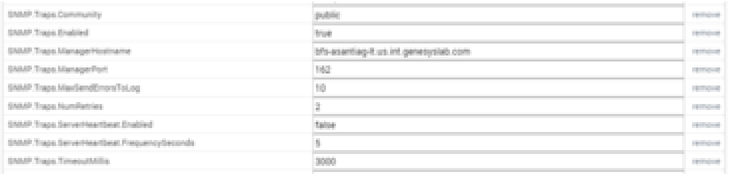
There is no specific way to respond to the notifications. The actions to be taken depend on each customer’s own processes, available resources, and capabilities. Therefore, we cannot predict in advance all possible combinations of alarm details, especially for the third category, as these depend heavily on the environment, call-flows, and third-party integration such as web services.

We recommend that if there are any errors that you are aware of that are likely to happen (for example, a web service call might time out), then it is worth simulating that in advance so that the Ops team can recognize that particular pattern.

Configuring Genesys Intelligent Automation to Send Traps

Traps can be configured through the Administration tab in the Genesys Intelligent Automation Control Centre. The following variables can be configured in the Default Server Settings tab:

- Enable Traps by setting **SNMP.Traps.Enabled** to `true`.
- Enable Server Heartbeat by setting **SNMP.Traps.ServerHeartbeat.Enabled** to `true`.
- Set the Server Heartbeat frequency using the **SNMP.Traps.ServerHeartbeat.FrequencySeconds** field.
- Provide the Manager host name in **SNMP.Traps.ManagerHostName**.

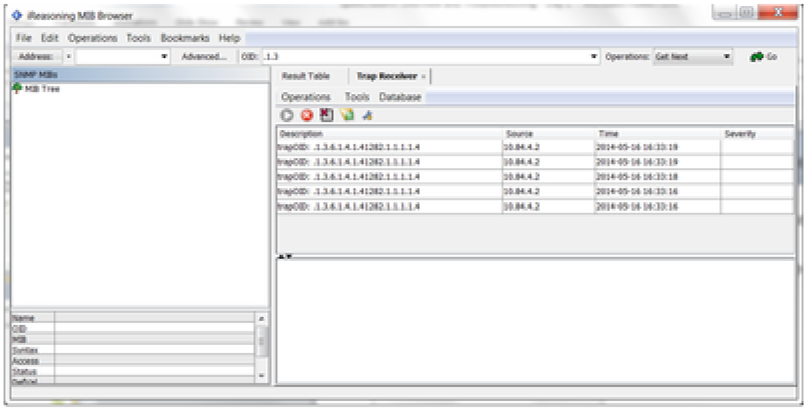


SNMP.Traps.Community	public	remove
SNMP.Traps.Enabled	true	remove
SNMP.Traps.ManagerHostname	bfs-asantiag-lt.us.int.genesyslab.com	remove
SNMP.Traps.ManagerPort	162	remove
SNMP.Traps.MaxSendErrorsToLog	10	remove
SNMP.Traps.NumRetries	2	remove
SNMP.Traps.ServerHeartbeat.Enabled	false	remove
SNMP.Traps.ServerHeartbeat.FrequencySeconds	5	remove
SNMP.Traps.TimeoutMillis	3000	remove

SNMP Traps and the configuration settings within Genesys Intelligent Automation are covered in detail during the Genesys Intelligent Automation Systems Administration training course.

View Traps

The Traps will start to regularly appear in your SNMP manager.



Description	Source	Time	Severity
trapOID: .1.3.6.1.4.1.41282.1.1.1.1.4	10.84.4.2	2014-05-16 16:33:19	
trapOID: .1.3.6.1.4.1.41282.1.1.1.1.4	10.84.4.2	2014-05-16 16:33:19	
trapOID: .1.3.6.1.4.1.41282.1.1.1.1.4	10.84.4.2	2014-05-16 16:33:18	
trapOID: .1.3.6.1.4.1.41282.1.1.1.1.4	10.84.4.2	2014-05-16 16:33:16	
trapOID: .1.3.6.1.4.1.41282.1.1.1.1.4	10.84.4.2	2014-05-16 16:33:16	

Limitations of SNMP Traps

As traps are sent in an asynchronous mode through UDP from Genesys Intelligent Automation to the configured SNMP manager, there is some uncertainty over delivery of the traps. Therefore, Genesys Intelligent Automation cannot assume that the traps are delivered and the destination cannot assume that all the traps are received.

Types of Traps

Reactive Traps

These traps are sent when there is some condition in Genesys Intelligent Automation that requires administrative attention. For example, if your Genesys Intelligent Automation license expires or if there is an error in a script block.

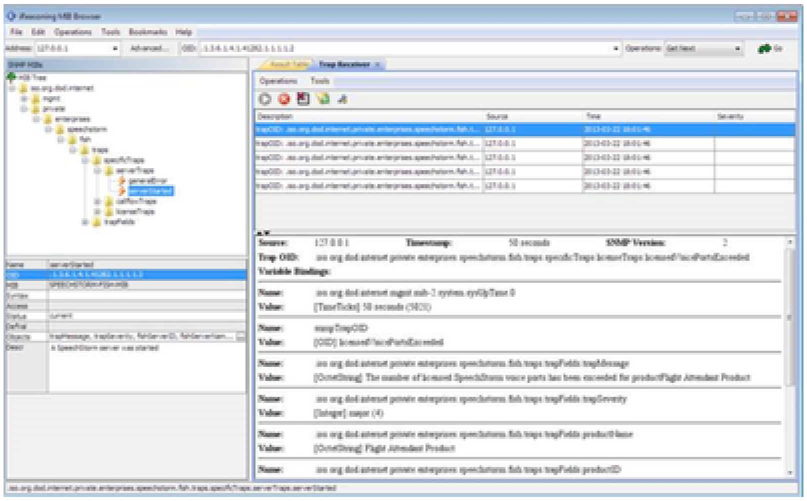
Server Traps

These traps are sent when the Tomcat Server is started or shutdown.

Genesys Intelligent Automation Server Heartbeat

Genesys Intelligent Automation Server Heartbeat can be used to monitor the status of the Server. The traps are sent at regular intervals that can be defined in the Genesys Intelligent Automation Default Server Settings. For example, if the Server is alive message is not received, it could indicate that the Server is down.

The following image shows an example of licenses that exceeded alerts:



Source: 127.0.0.1 Timestamp: 58 seconds SNMP Version: 2
Trap OID: .iso.org.dod.internet.private.enterprises.speechstorm.fish.traps.specificTraps.licenseTraps.licensedVoicePortsExceeded
Variable Bindings:
Name: .iso.org.dod.internet.mgmt.mib-2.system.sysUpTime.0
Value: [TimeTicks] 58 seconds (5833)
Name: snmpTrapOID
Value: [OID] licensedVoicePortsExceeded
Name: .iso.org.dod.internet.private.enterprises.speechstorm.fish.traps.trapFields.trapMessage
Value: [OctetString] The number of licensed SpeechStorm voice ports has been exceeded for productFlight Attendant Product
Name: .iso.org.dod.internet.private.enterprises.speechstorm.fish.traps.trapFields.trapSeverity
Value: [Integer] major (4)
Name: .iso.org.dod.internet.private.enterprises.speechstorm.fish.traps.trapFields.productName
Value: [OctetString] Flight Attendant Product
Name: .iso.org.dod.internet.private.enterprises.speechstorm.fish.traps.trapFields.productID

Genesys Intelligent Automation Traps and Descriptions

Trap OID	Trap Name	Severity	Description	User Action
.1.3.6.1.4.1.41282.1.1.1.1.2	server-status-UP	Minor	A Genesys Intelligent Automation server was started. This cancels a **server-status-DOWN** trap that was fired from the same fish-server-ID.	No action needed.
.1.3.6.1.4.1.41282.1.1.1.1.3	server-status-DOWN	Critical	A Genesys Intelligent Automation server was stopped. This generally signals a graceful or planned stop but there may be other cases when the process ends unexpectedly and does not have the opportunity to send an SNMP notification. A **server-status-UP** trap with the same fish-server-ID should be sent when this server component is started again.	The Genesys Intelligent Automation service must be restarted on that specific server after checking the logs if it is not a planned activity.
.1.3.6.1.4.1.41282.1.1.1.1.4	server-status-heartbeat	Minor	A trap that is sent periodically from each Genesys Intelligent Automation server to indicate that the server process is still running. The scheduling for these can be adjusted from within the Genesys Intelligent Automation GUI. They can also be disabled if desired	No action needed as this will be switched off in the platform.
.1.3.6.1.4.1.41282.1.1.1.2.1	callflow-block-exception	Major	An error occurred while processing a particular block as part of a callflow.	Check the specific error in fish VUI logs and take action accordingly

Trap OID	Trap Name	Severity	Description	User Action
			This could have been caused by a malformed script, an uninitialized variable, an instruction in a script to log an error, a failed web service call from within the block, or any other kind of general error that might occur within a callflow block.	in the call flow level.
.1.3.6.1.4.1.41282.1.1.1.3.1	licensed-voice-ports-exceeded	Major	The system attempted to check out a license for a new voice call when there were no available licenses remaining.	Check the SS fish/LMAdmin logs and identify host from which the voice ports license issue is reported and then contact SS with details for support/guidance.
.1.3.6.1.4.1.41282.1.1.1.3.2	licensed-product-ports-exceeded	Major	The system attempted to check out a license for a new product-based module during a call when there were no available licenses remaining.	Check the SS fish/LMAdmin logs and identify host and product from which the licenses issue is reported and then contact SS with details for support/guidance.
.1.3.6.1.4.1.41282.1.1.1.1	general-error	Major	An unexpected error was reported. This will generally be sent as a major error.	Check the specific error in fish logs and take action accordingly.

Hive off recovery process for Oracle

Process Summary

The purpose of this memo is to give background information on the Genesys Intelligent Automation ‘hive off’ process and to provide instruction on the most efficient way to recover from a failure in the hive off process in Oracle.

Due to the large amount of reporting data captured and processed by the Genesys Intelligent Automation application it is imperative that, should a failure in the hive off process occur, the system is quickly brought back into a state whereby the next night’s hive off can run successfully.

> **Important**
>
> If corrective action is not taken within several hours of the original failure, then data loss is likely to result.

Definitions and Background

Term	Definition
Hive off	The nightly batch job that runs on the primary Genesys Intelligent Automation GUI server in order to manage the large amounts of reporting data that are captured during the day. The hive off job typically runs in the early hours of each morning and can take several hours to complete.
Daily tables	The DB tables that receive details about incoming calls and Control Centre auditing for the current day. These tables are constantly receiving new reporting data from the Genesys Intelligent Automation servers: • calls • call_sites • call_steps • business_tasks • gui_actions
Historical tables	Read-only data that for previous days, that has

Term	Definition
	been moved from the ‘daily’ tables to their ‘historical’ equivalents as part of the hive off: • historical_calls • historical_call_sites • historical_call_steps • historical_business_tasks • historical_gui_actions
Aggregate tables	Read-only data for previous days, that has been moved from the ‘daily’ tables as part of the hive off process and report generation: • HIST_STATS_RECOGNITION1 • HIST_STATS_BLOCKRESULTS • HIST_STATS_BUSINESSTASKS • LAST31DAYS_SLOTS • DAILY_HEALTH_AGGREGATES_START • DAILY_HEALTH_AGGREGATES • DAILY_DASHBOARD_AGGREGATES • HIST_STATS_BUSINESSTASKS1 • HIST_STATS_BUSINESSTASKS2 • HIST_STATS_CALLJOURNEYS • HIST_STATS_CALLSPERHOUR • HIST_STATS_CALLSPERDAY • HIST_STATS_LASTMENU • HIST_STATS_RECOGNITION • HIST_STATS_CALLSUMMARY

Steps involved in hive off process

The Genesys Intelligent Automation hive off process consists of four distinct steps:

1. Calculating aggregates and copying records from the ‘daily’ tables into the ‘historical’ tables;
2. Deleting those records from the ‘daily’ tables that were copied to the ‘historical’ tables in step 1;
3. Deleting any expired data from the ‘historical’ tables as per the **NumDaysHistoricalDataToKeep** setting;

4. Re-enabling indexes on the ‘daily’ tables. **Note:** Indexes do not get disabled or re-enabled in the Oracle version of Genesys Intelligent Automation. However, the step is still present in the hive off and so you may still see references to it when trying to correct hive off issues.

Handling hive off failure

Should the hive off process fail for any reason, the corrective actions to ensure that the next night’s hive off runs successfully will be different depending on which step the hive off failed.

Determining at which step the hive off failed

This can be best determined by looking at the **fish.log** file (and, perhaps, fish.log.1, fish.log.2, etc.) in the primary GUI server’s `SpeechStorm\Platform\TomcatGUI\logs` directory.

The names of the steps in the log file appear as:

1. populateAndHiveOff
2. deleteBatchAfterHiveOff
3. deleteOldHistoricalData
4. rebuildIndexes

The following entries should be written to the logs during each hive off run, and the presence or absence of certain of the entries below will indicate which step of the hive off failed, thus allowing you to follow the correct path in the Recovery Process in the next section.

Start of step #1:

```
[INFO ] 2016-06-15 00:15:00,833 (Timer-1::)
com.speechstorm.fish.reporting.PopulateAggregatesAndHiveOffHistoricalDataTask.populateAggregates
starting at ...
```

The following entry indicates that step #1 has finished and step #2 is about to begin:

```
[INFO ] 2016-06-15 00:37:48,557 (Timer-1::)
com.speechstorm.fish.reporting.PopulateAggregatesAndHiveOffHistoricalDataTask.populateAggregates
finished populateAndHiveOff at ...
```

This entry indicates that step #2 has finished and step #3 is about to begin:

```
[INFO ] 2016-06-15 01:26:43,590 (Timer-1::)
com.speechstorm.fish.reporting.PopulateAggregatesAndHiveOffHistoricalDataTask.populateAggregates
finished all deleteBatchAfterHiveOff steps at ...
```

This entry indicates that step #3 has finished and step #4 is about to begin:

```
[INFO ] 2016-06-15 02:06:24,271 (Timer-1::)
com.speechstorm.fish.reporting.PopulateAggregatesAndHiveOffHistoricalDataTask.deleteOldHistorica
finished at ...
```

This entry indicates that all steps have completed:

```
[INFO ] 2016-06-15 02:06:45,571 (Timer-1::)
com.speechstorm.fish.reporting.PopulateAggregatesAndHiveOffHistoricalDataTask.rebuildIndexes()
```
finished rebuildIndexes at ...

Recovery process

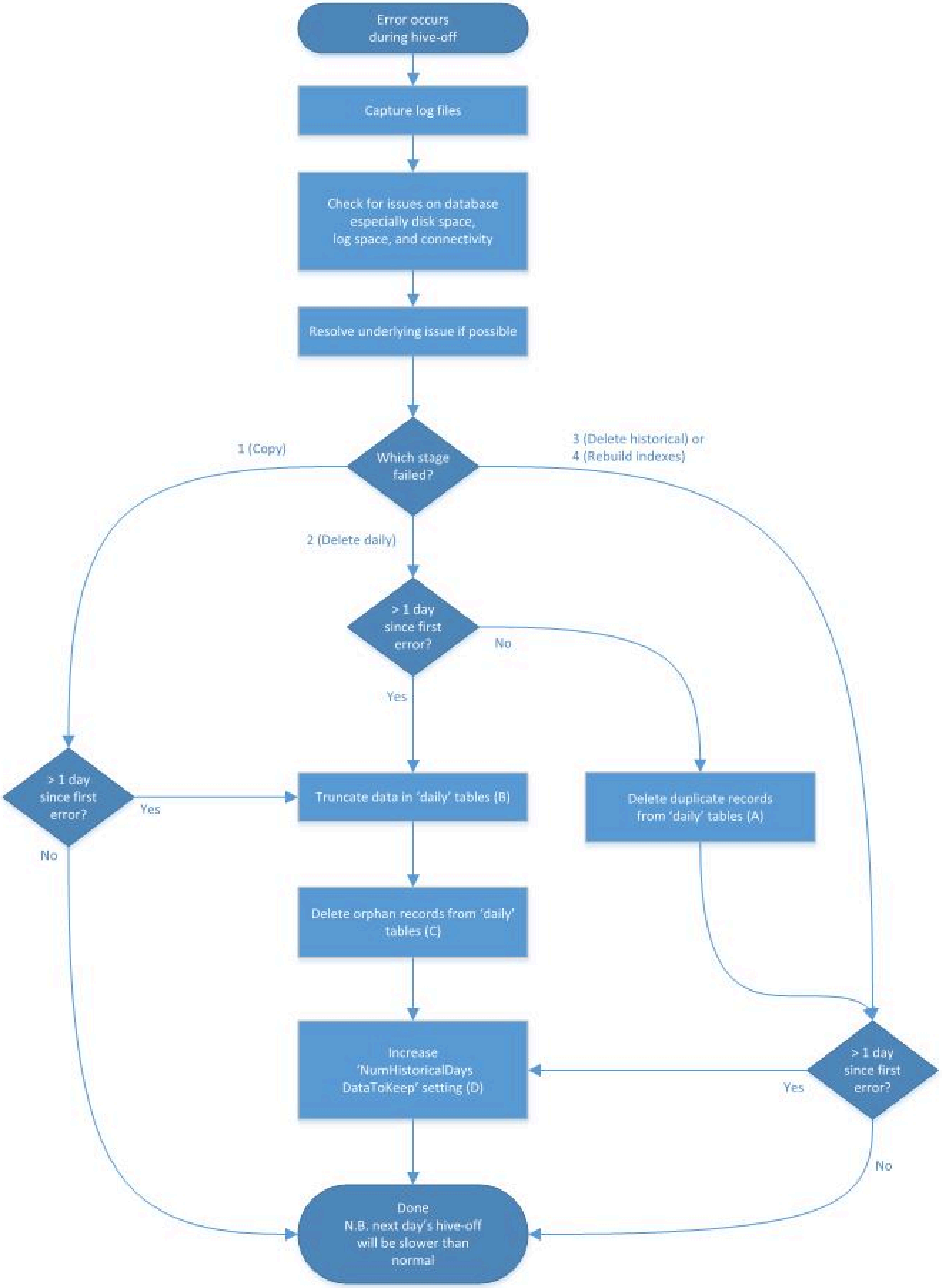
While it is important to understand and fix the problem that caused the hive off to fail, if this cannot be done immediately (for example, it requires more investigation) then it is better to take action immediately to ensure that the next night's hive off still runs.

There is a short window of time each night in which to run the hive off (for example, between midnight and 7am) and if the hive off does not run successfully for several days in a row then the amount of data in the 'daily' tables will become prohibitive and will have an impact on both call handling and reporting; the only practical solution should this be allowed to happen is to do a bulk delete of the 'daily' tables, thus causing reporting data to be lost.

The main factors to take into account are:

- There must be no duplicate records in the 'daily' and 'historical' tables when the hive off runs;
- If the hive off has not run for several days and the value of the **DBOvernightJobs.NumDaysHistoricalDataToKeep** setting is not increased accordingly then the hive off will attempt to delete a much larger chunk of records from the 'historical' tables, which will take proportionally longer to execute and could lead to running out of transaction log space on the DB server.

The overall hive off recovery process is described in the following diagram:


Error occurs during hive-off
Capture log files
Check for issues on database especially disk space, log space, and connectivity
Resolve underlying issue if possible
Which stage failed?
1 (Copy)
2 (Delete daily)
3 (Delete historical) or 4 (Rebuild indexes)
> 1 day since first error?
No
Yes
> 1 day since first error?
Yes
No
Truncate data in 'daily' tables (B)
Delete duplicate records from 'daily' tables (A)
Delete orphan records from 'daily' tables (C)
Increase 'NumHistoricalDays DataToKeep' setting (D)
> 1 day since first error?
Yes
No
Done N.B. next day's hive-off will be slower than normal

Corrective actions in detail

One or more of the following actions may be required according to the diagram above.

(A) Delete duplicate records from ‘daily’ tables

This action is required if the hive off stopped during step 2. There will be no loss of reporting data when you execute this step, as the data has already been copied to the ‘historical’ tables.

The SQL to be run against the ‘fishreports’ database is as follows:

```
DELETE FROM call_steps WHERE id IN (SELECT id FROM historical_call_steps) AND rownum
<= 20000

DELETE FROM call_sites WHERE id IN (SELECT id FROM historical_call_sites) AND rownum
<= 20000

DELETE FROM business_tasks WHERE id IN (SELECT id FROM historical_business_tasks) AND
rownum <= 20000

DELETE FROM calls WHERE call_id IN (SELECT call_id FROM historical_calls) AND rownum
<= 20000

DELETE FROM gui_actions WHERE id IN (SELECT id FROM historical_gui_actions) AND
rownum <= 20000
```

> **Important**
>
> This SQL should be run multiple times until there are no more duplicate records reported.

The max rownum limit can be increased with caution, but care needs to be taken that this does not have an adverse effect on table locking; monitoring is key here.

(B) Truncate data in ‘daily’ tables

This action is required if the amount of data in the ‘daily’ tables is such that the hive off is likely to run into peak business hours.

> **Important**
>
> Be aware that this step will result in a loss of all reporting data in the ‘daily’ tables, although the reporting data in the ‘historical’ tables will not be affected. We recommend taking a backup of the database just before running it in order to preserve the data in the ‘daily’ tables for further analysis of the root cause.

The SQL to run against the ‘fishreports’ database is as follows:

```
truncate table calls;

truncate table call_sites;

truncate table call_steps;

truncate table business_tasks;

truncate table gui_actions;
```

(C) Delete orphan records from ‘daily’ tables

This action is required once the ‘daily’ tables have been truncated, to ensure that there are no child records (for example, call_steps) whose parent records (i.e. calls) have been truncated. This situation can occur due to the timing of the truncate statements. It will result in the loss of a minute amount of reporting data, and can be viewed as the completion of the ‘truncate’ operation above.

There should be a one-minute pause between executing action (B) and this action. The SQL to run against the ‘fishreports’ database is as follows:

```
delete from call_sites where call_id not in (select call_id from calls);

delete from call_steps where call_id not in (select call_id from calls);

delete from business_tasks where call_id not in (select call_id from calls);
```

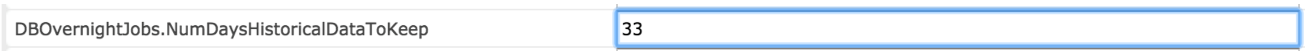
(D) Increase ‘numdayshistoricaldatatokeep’ setting

Run the following SQL against the ‘fishreports’ database to determine what value should be used for the **NumDaysHistoricalDataToKeep** setting in order to only hive off a maximum of 2 days’ old historical data at once:

```
DECLARE mindate DATE;

begin

select (SELECT min(call_start_date) FROM historical_calls) into mindate from DUAL;

dbms_output.put_line((SYSDATE-mindate) - 1);

end;
```

You need only change the value of the setting if the result of the above SQL is greater than the original/current value of the setting.

The setting can be changed via the Genesys Intelligent Automation Control Centre, in the **Administration -> Default Server Settings** page:



DBOvernightJobs.NumDaysHistoricalDataToKeep	33

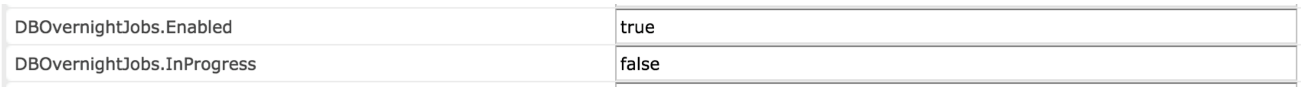
The value should be decreased each day from this point forward until it reaches its original value. In this way each night's hive off will involve deleting a maximum of 2 days old historical data.

Next Steps

1. Ensuring that the next hive off will run

Ensure that the hive off job is enabled and that the **in progress** flag is set to `false`.

These settings can be changed via the Genesys Intelligent Automation Control Centre, in the **Administration -> Default Server Settings** page:

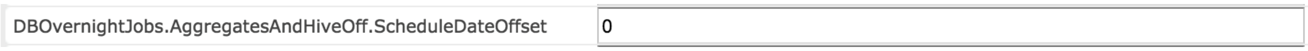


DBOvernightJobs.Enabled	true
DBOvernightJobs.InProgress	false

You might also need to change the **DBOvernightJobs.Enabled** setting to `true` on the Primary GUI via the Genesys Intelligent Automation Control Centre, in the **Administration** -> **Servers** -> **Edit** (for the Primary GUI).

2. (Optional) Manually re-running the hive-off

In high-call traffic scenarios, it is best not to run the hive-off during the day but instead to wait for it to run automatically the following night. If the recovery steps have been undertaken shortly after a failure, or in a low-call traffic scenario, however, it is possible to kick off the hive-off process manually. To do this, change the following setting (in the same location as above) from `1` to `0`:



DBOvernightJobs.AggregatesAndHiveOff.ScheduleDateOffset	0

Now restart the primary GUI server and the hive-off will begin immediately, even though the current time is after the schedule hour and minute that are defined in the server settings.

When the hive-off has finished, be sure to change this setting back to 1 so that any future restarts of the GUI server will not trigger an immediate hive-off. **Note:** It is not required to restart the GUI again at this point.

3. Identifying the root cause

If the underlying cause for the failure has not yet been identified and resolved, then you should take steps to record more detailed log events in order to better troubleshoot if the issue recurs. On the database server, your DBAs will be able to advise on the best additional logging setup. On the Genesys Intelligent Automation primary GUI server, you should enable full SQL debug logs by editing the `SpeechStorm\Platform\TomcatGUI\webapps\fish-gui\WEB-INF\classes\log4j.properties` file and ensure that the following line is not commented out (i.e. ensure that it does not begin with a '#' character):

```
log4j.logger.java.sql=DEBUG
```

The setting should be picked up automatically without requiring a restart.

Enabling full SQL debug logs will create a lot of log files on the primary GUI server, so care must be taken to perform a daily archive of these logs stretching back to at least 3 months.

4. Managing the ongoing deletion of old historical records

Remember to decrease **NumDaysHistoricalDataToKeep** by 1 each day if it was increased as per action (E) above. This should be done daily until the setting is back to its original value.

Hive Off Recovery Process for SQL Server

Process Summary

The purpose of this memo is to give background information on the Genesys Intelligent Automation ‘hive off’ process and to provide instruction as to the most efficient way to recover from a failure in the hive off process in an SQL Server environment.

Due to the large amount of reporting data captured and processed by the Genesys Intelligent Automation application it is imperative that, should a failure in the hive off process occur, the system is quickly brought back into a state whereby the next night’s hive off can run successfully.

> ### Important
> If corrective action is not taken within several hours of the original failure, then data loss is likely to result.

Definitions and Background

Term	Definition
Hive off	The nightly batch job that runs on the primary Genesys Intelligent Automation GUI server in order to manage the large amounts of reporting data that is captured during the day. The hive off job typically runs in the early hours of each morning and can take several hours to complete.
Daily tables	The DB tables that receive details about incoming calls and Control Centre auditing for the current day. These tables are constantly receiving new reporting data from the Genesys Intelligent Automation servers: • calls • call_sites • call_steps • business_tasks • gui_actions
Historical tables	Read-only data for previous days, that has been

Term	Definition
	moved from the ‘daily’ tables to their ‘historical’ equivalents as part of the hive off: • historical_calls • historical_call_sites • historical_call_steps • historical_business_tasks • historical_gui_actions
Aggregate tables	Read-only data for previous days, that has been moved from the ‘daily’ tables as part of the hive off process and report generation: • daily_dashboard_aggregates • daily_health_aggregates • daily_health_aggregates_start • last31days_slots • reports_blockresults_blockResults_aggregate • reports_businessTasksSummary_businessTasksSummary_aggre • reports_callJourneys_aggregate • reports_callsbyday_callsByDate_aggregate • reports_callsbyHour_callsByHour_aggregate • reports_recognitionsummary_inputBlockSummary_aggregate • reports_summary_businesstasks1_aggregate • reports_summary_businesstasks2_aggregate • reports_summary_callsummary_aggregate • reports_summary_lastmenu_aggregate • reports_summary_recognitionsummary1_aggregate

The Genesys Intelligent Automation hive off process consists of four distinct steps:

1. Calculating aggregates and copying records from the ‘daily’ tables into the ‘historical’ tables;
2. Disabling indexes on the ‘daily’ tables and deleting those records from the ‘daily’ tables that were copied to the ‘historical’ tables in step 1;
3. Deleting any expired data from the ‘historical’ tables as per the **NumDaysHistoricalDataToKeep** setting;
4. Re-enabling indexes on the ‘daily’ tables.

Should the hive off process fail for any reason, the corrective actions to ensure that the next night’s hive off runs successfully will be different depending on which step the hive off failed.

Determining at Which Step the Hive Off Failed

This can be best determined by looking at the fish.log file (and, perhaps, fish.log.1, fish.log.2, etc.) in the primary GUI server's *SpeechStorm\Platform\TomcatGUI\logs* directory.

The names of the steps in the log file appear as:

1. populateAndHiveOff
2. deleteBatchAfterHiveOff
3. deleteOldHistoricalData
4. rebuildIndexes

The following entries should be written to the logs during each hive off run, and the presence or absence of certain of the entries below will indicate which step of the hive off failed, thus allowing you to follow the correct path in the Recovery Process in the next section.

Start of step #1:

```
[INFO ] 2016-06-15 00:15:00,833 (Timer-1::)
com.speechstorm.fish.reporting.PopulateAggregatesAndHiveOffHistoricalDataTask.populateAggregates
```
starting at ...

The following entry indicates that step #1 has finished and step #2 is about to begin:

```
[INFO ] 2016-06-15 00:37:48,557 (Timer-1::)
com.speechstorm.fish.reporting.PopulateAggregatesAndHiveOffHistoricalDataTask.populateAggregates
```
finished populateAndHiveOff at ...

This entry indicates that step #2 has finished and step #3 is about to begin:

```
[INFO ] 2016-06-15 01:26:43,590 (Timer-1::)
com.speechstorm.fish.reporting.PopulateAggregatesAndHiveOffHistoricalDataTask.populateAggregates
```
finished all deleteBatchAfterHiveOff steps at ...

This entry indicates that step #3 has finished and step #4 is about to begin:

```
[INFO ] 2016-06-15 02:06:24,271 (Timer-1::)
com.speechstorm.fish.reporting.PopulateAggregatesAndHiveOffHistoricalDataTask.deleteOldHistorica
```
finished at ...

This entry indicates that all steps have completed:

```
[INFO ] 2016-06-15 02:06:45,571 (Timer-1::)
```

com.speechstorm.fish.reporting.PopulateAggregatesAndHiveOffHistoricalDataTask.rebuildIndexes()
finished rebuildIndexes at ......

Recovery Process

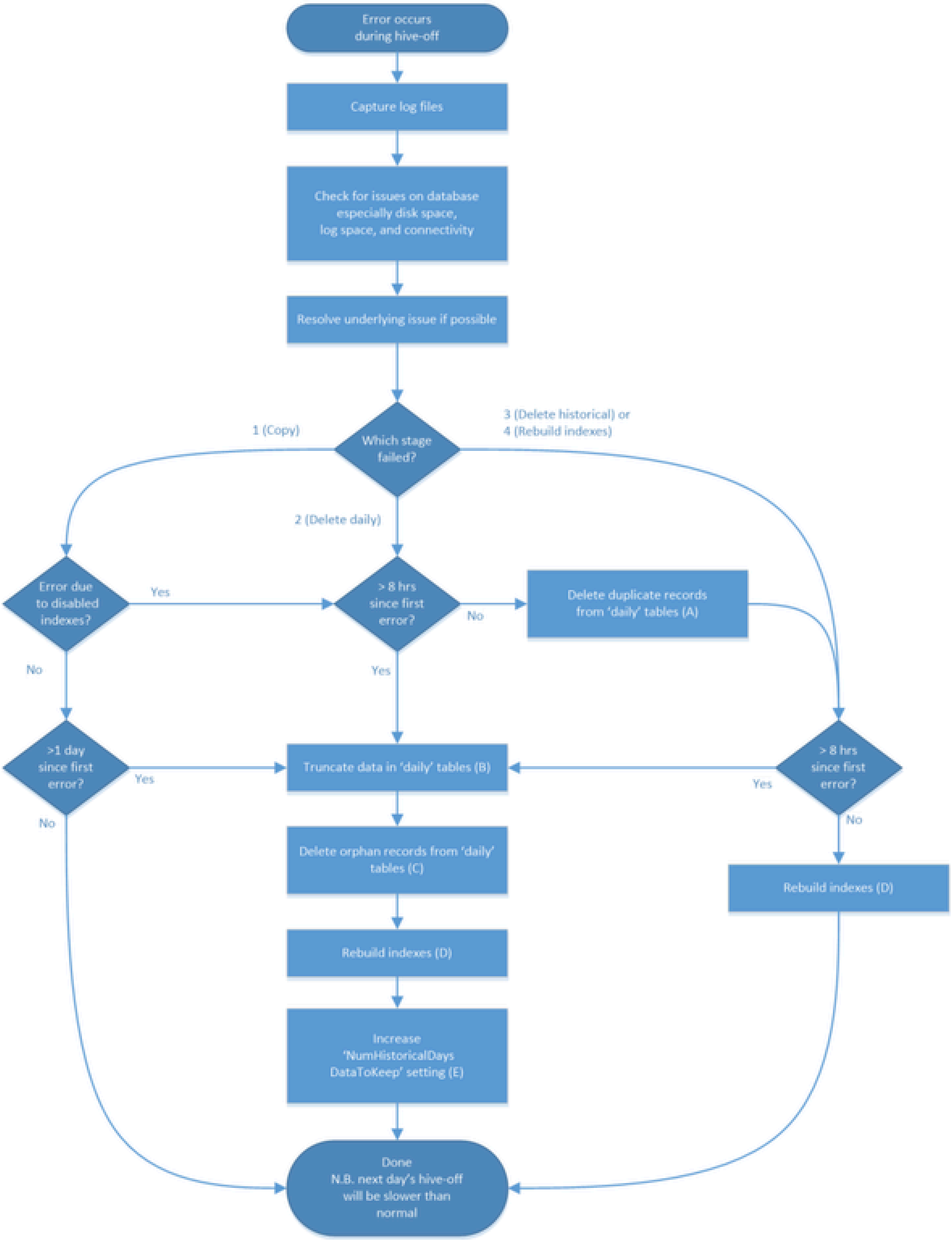
While it is important to try to understand and fix the problem that caused the hive off to fail, if this cannot be done immediately (for example, it requires more investigation) then it is better to take action immediately to ensure that the next night’s hive off still runs.

There is a short window of time each night in which to run the hive off (for example, between midnight and 7am) and if the hive off does not run successfully for several days in a row then the amount of data in the ‘daily’ tables will become prohibitive and will have an impact on both call handling and reporting; the only practical solution should this be allowed to happen is to do a bulk delete of the ‘daily’ tables, **thus causing reporting data to be lost**.

The main factors to take into account are:

- There must be no duplicate records in the ‘daily’ and ‘historical’ tables when the hive off runs;
- The indexes on the ‘daily’ tables need to be enabled before the hive off runs;
- Re-enabling indexes on the ‘daily’ tables when there is a lot of data in those tables will lock the tables for a significant period of time (for example, minutes or even hours) which could adversely affect call handling or cause a large backlog of reporting data waiting to be written;

If the hive off has not run for several days and the value of the **DBOvernightJobs.NumDaysHistoricalDataToKeep** setting is not increased accordingly then the hive off will attempt to delete a much larger chunk of records from the ‘historical’ tables, which will take proportionally longer to execute and could lead to running out of transaction log space on the DB server. The overall hive off recovery process is described in the following diagram:


Error occurs during hive-off
Capture log files
Check for issues on database especially disk space, log space, and connectivity
Resolve underlying issue if possible
Which stage failed?
1 (Copy)
3 (Delete historical) or 4 (Rebuild indexes)
2 (Delete daily)
Error due to disabled indexes?
Yes
No
> 8 hrs since first error?
No
Yes
Delete duplicate records from 'daily' tables (A)
>1 day since first error?
Yes
No
Truncate data in 'daily' tables (B)
> 8 hrs since first error?
Yes
No
Delete orphan records from 'daily' tables (C)
Rebuild indexes (D)
Rebuild indexes (D)
Increase 'NumHistoricalDays DataToKeep' setting (E)
Done N.B. next day's hive-off will be slower than normal

Corrective Actions in Detail

One or more of the following actions may be required according to the diagram above.

(A) Delete Duplicate Records From ‘Daily’ Tables

This action is required if the hive off stopped during step 2. There will be no loss of reporting data when you execute this step, as the data has already been copied to the ‘historical’ tables. The SQL to be run against the ‘fishreports’ database is as follows:

```
set ROWCOUNT 20000;

delete from call_steps where id in (select id from historical_call_steps);

set ROWCOUNT 20000;

delete from call_sites where id in (select id from historical_call_sites);

set ROWCOUNT 20000;

delete from business_tasks where id in (select id from historical_business_tasks);

set ROWCOUNT 20000;

delete from calls where call_id in (select call_id from historical_calls);

set ROWCOUNT 20000;

delete from gui_actions where id in (select id from historical_gui_actions);
```

> **Important**
>
> This SQL should be run multiple times until there are no more duplicate records reported.

The max **ROWCOUNT** limit can be increased with caution, but care needs to be taken that this does not have an adverse effect on table locking; monitoring is key here.

(B) Truncate Data in ‘Daily’ Tables

This action is required if the amount of data in the ‘daily’ tables is such that re-enabling the indexes would lock the tables for a prohibitive amount of time.

Be aware that this step will result in a loss of all reporting data in the ‘daily’ tables, although reporting data in the ‘historical’ tables will not be affected. We recommend taking a backup of the database just before running it in order to preserve the data in the ‘daily’ tables for further analysis of the root cause.

The SQL to run against the ‘fishreports’ database is as follows:

```
truncate table calls;

truncate table call_sites;

truncate table call_steps;

truncate table business_tasks;

truncate table gui_actions;

declare @maxid bigint;

select @maxid = max(call_id) + 2000 from historical_calls;

dbcc checkident (calls, RESEED, @maxid);

select @maxid = max(id) + 2000 from historical_call_sites;

dbcc checkident (call_sites, reseed, @maxid);

select @maxid = max(id) + 2000 from historical_call_steps;

dbcc checkident (call_steps, reseed, @maxid);

select @maxid = max(id) + 2000 from historical_business_tasks;

dbcc checkident (business_tasks, reseed, @maxid);

select @maxid = max(id) + 2000 from historical_gui_actions;

dbcc checkident (gui_actions, reseed, @maxid);

delete from calls where call_id < 2000;

delete from call_sites where id < 2000;

delete from call_steps where id < 2000;

delete from business_tasks where id < 2000

delete from gui_actions where id < 2000;
```

(C) Delete Orphan Records from ‘Daily’ Tables

This action is required once the ‘daily’ tables have been truncated, to ensure that there are no child records (for example, call_steps) whose parent records (i.e. calls) have been truncated. This situation can occur due to the timing of the truncate statements. It will result in the loss of a minute amount of reporting data, and can be viewed as the completion of the ‘truncate’ operation above.

There should be a one-minute pause between executing action (B) and this action. The SQL to run against the ‘fishreports’ database is as follows:

```
delete from call_sites where call_id not in (select call_id from calls);
delete from call_steps where call_id not in (select call_id from calls);

delete from business_tasks where call_id not in (select call_id from calls);
```

(D) Rebuild Indexes

This action is required any time the hive off fails after step 1, in order to ensure that the indexes on the ‘daily’ tables are enabled and working for the next night’s hive off. The SQL to run against the ‘fishreports’ database is as follows:

```
exec USP_RebuildIndexes
```

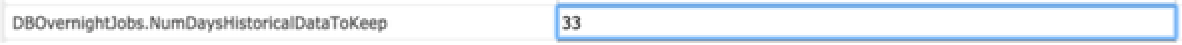
This could be time-intensive, depending on the amount of data that is in the ‘daily’ tables at the time that the command is issued.

(E) Increase NumDaysHistoricalDataToKeep Setting

Run the following SQL against the ‘fishreports’ database to determine what value should be used for the **NumDaysHistoricalDataToKeep** setting in order to only hive off a maximum of 2 days’ old historical data at once:

```
declare @mindate DATE

select @mindate = min(call_start_date) from historical_calls

select @mindate, datediff(d, @mindate, GETDATE()) - 1 as NewDaysToKeepSettingValue
```

You need only change the value of the setting if the result of the above SQL is greater than the original/current value of the setting. The setting can be changed via the Genesys Intelligent Automation Control Centre, in the **Administration** -> **Default Server Settings** page:



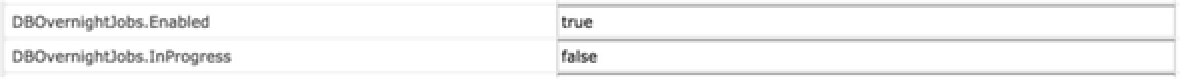
DBOvernightJobs.NumDaysHistoricalDataToKeep	33

The value should be decreased each day from this point forward until it reaches its original value. In this way each night’s hive off will involve deleting a maximum of 2 days old historical data.

Next Procedure

1.Ensuring That the Next Hive Off Will Run

Ensure that the hive off job is enabled and that the **in progress** flag is set to *false*. These settings can be changed via the Genesys Intelligent Automation Control Centre, in the **Administration** -> **Default Server Settings** page:

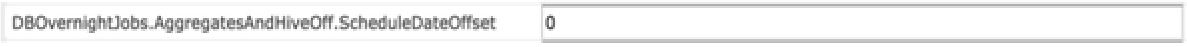


DBOvernightJobs.Enabled	true
DBOvernightJobs.InProgress	false

2.(Optional) Manually Re-Running the Hive-Off

In high-call traffic scenarios, it is best not to run the hive-off during the day but instead to wait for it to run automatically the following night.

If the recovery steps have been undertaken shortly after a failure, or in a low-call traffic scenario, however, it is possible to kick off the hive-off process manually. To do this, change the following setting (in the same location as above) from 1 to 0:



DBOvernightJobs.AggregatesAndHiveOff.ScheduleDateOffset 0

Now restart the primary GUI server and the hive-off will begin immediately, even though the current time is after the schedule hour and minute that are defined in the server settings. When the hive-off has finished, be sure to change this setting back to 1 so that any future restarts of the GUI server will not trigger an immediate hive-off. (It is not required to restart the GUI again at this point.)

3.Identifying the Root Cause

If the underlying cause for the failure has not yet been identified and resolved, then you should take steps to record more detailed log events in order to better troubleshoot if the issue recurs. On the database server, your DBAs will be able to advise on the best additional logging setup. On the Genesys Intelligent Automation primary GUI server, you should enable full SQL debug logs by editing the `SpeechStorm\Platform\TomcatGUI\webapps\fish-gui\WEB-INF\classes\log4j.properties` file and ensure that the following line is not commented out (i.e. ensure that it does not begin with a ‘#’ character):

```
log4j.logger.java.sql=DEBUG
```

The setting should be picked up automatically without requiring a restart.

Enabling full SQL debug logs will create a lot of log files on the primary GUI server, so care must be taken to perform a daily archive of these logs stretching back to at least 3 months.

4.Managing the Ongoing Deletion Of Old Historical Records

Remember to decrease ‘NumDaysHistoricalDataToKeep’ by 1 each day if it was increased as per action (E) above. This should be done daily until the setting is back to its original value.

Auditing

To track the usage of bots used in Intelligent Automation, Genesys offers SQL (MS-SQL and Oracle PL-SQL) scripts to retrieve usage data.

Requirements

- Intelligent Automation 9.0.106 and later
- Permissions to run SQL scripts

Important

- Run when the Hive Off activity is not running. Hive Off is recommended to be run nightly.
- Six weeks historical data is recommended. You can decide on the retention policies based on your organization requirements. The default setting is 30 days.

Running the Audit scripts

Important

For Interaction Billing - Genesys measures from **midnight** to **midnight** and counts the unique channel conversation IDs* within that timeframe.

- Depending on your database format, download the following script in the Intelligent Automation database:
  - **botaudit-mssql.sql** if you are using MS-SQL
  - **botaudit-oracle.sql** if you are using Oracle SQL
- Modify the following script (for both customer types)
  - For MS-TSQL:

```
declare @start_dateX as date = '2021-01-03 00:00:00.0'
declare @end_dateX as date = '2021-12-31 23:59:59.999'
declare @companyIDX int = 2
declare @siteIdX int = null
```

- - For Oracle PL-SQL:

```
start_dateX DATE := TO_DATE('01/01/2021', 'DD/MM/YYYY');
end_dateX date := TO_DATE('30/12/2021', 'DD/MM/YYYY');
companyIDX int := 2;
siteIdX int := 0;
```

The parameters to be modified are:

1. **start_dateX** - The start date range. Use the DD/MM/YYYY format.
2. **end_dateX** - The end date range. Use the DD/MM/YYYY format.
3. **companyIDX** - Use the correct company ID. Wildcard characters are not allowed.
4. **siteIdX** - This is not used for calculation, but rather for display purposes only.

The scripts generate three reports:

1. Count of Interactions count for Chatbot legacy customers
2. Count of Turns for chatbot new customers
3. Count of interactions for Voicebot customers

Chatbots

If NLU is used at any time during the call, Genesys counts and sums the entire call duration for a 24-hour period.

Limitations

- Chatbot interactions will no longer be offered to new customers.
- Chatbot turns are the new meter.
- Chatbot customers will be switched to turns when they renew their contract.

Voicebots

If NLU is used at any time during the call, Genesys counts and sums the entire call duration for a 24-hour period.

Limitations

- Voicebot interactions will no longer be offered.
- Voicebot minutes will remain as the only meter.
- Voicbot are not offered on a turn basis.

Sample Reports

For Chatbot Turns report,

- For MS-TSQL databases,
    - declare `@ChatBotTurns bit = 0`
        - For chat use, set `@IsChatCall bit = 1`
        - For voice use, set `@IsChatCall bit = 0`
- For Oracle PL-SQL databases
    - ChatBotTurns = false
        - For chat, use `IsChatCall number(1) := 1`
        - For voice, use `IsChatCall number(1) := 0`

Server Settings Options

Name	Default Value	Valid Values	Takes Effect	Description
SSML.Pause.IncrementDurationMillis	50	Integer	After GUI restart	Duration in milliseconds for the [pause] when rendering a prompt using SSML
Grammar.EmptyGrammar.RecognitionPhrase	fish grammar	String	After GUI re-start	String to configure the grammar sent to help troubleshooting when finding issues with the DTMF key ahead functionality.
Personas.Google.en-gb.TTSVoiceNames	en-GB-Standard-A	Comma separated Strings	After GUI re-start	Voice name for Google en-gb to display in the Persona>TTS Voices dropdown
Personas.Google.en-us.TTSVoiceNames	en-US-Standard-B	Comma separated Strings	After GUI re-start	Voice name for Google en-US to display in the Persona>TTS Voices dropdown
Personas.Google.es-mx.TTSVoiceNames	es-ES-Standard-A	Comma separated Strings	After GUI re-start	Voice name for Google es-ES to display in the Persona>TTS Voices dropdown
Personas.Google.fr-fr.TTSVoiceNames	fr-FR-Standard-A	Comma separated Strings	After GUI re-start	Voice name for Google fr-FR to display in the Persona>TTS Voices dropdown
Personas.Nuance.en-gb.TTSVoiceNames	Daniel	Comma separated Strings	After GUI re-start	Voice name for Nuance en-gb to display in the Persona>TTS Voices dropdown
Personas.Nuance.en-us.TTSVoiceNames	Allison	Comma separated Strings	After GUI re-start	Voice name for Nuance en-us to display in the Persona>TTS Voices dropdown
Personas.Nuance.es-mx.TTSVoiceNames	Angelica	Comma separated Strings	After GUI re-start	Voice name for Nuance es-mx to display in the

Name	Default Value	Valid Values	Takes Effect	Description
				Persona>TTS Voices dropdown
Personas.Nuance.fr-fr.TTSVoiceNames	Audrey	Comma separated Strings	After GUI re-start	Voice name for Nuance fr-fr to display in the Persona>TTS Voices dropdown
DialogEngine.JOPv3.BaseURL		String	After GUI re-start	Base API URL for DialogEngine NLU runtime
DialogEngine.JOPv3.AuthBaseURL		String	After GUI re-start	Base URL for authorisation and token retrieval for subsequent DialogEngine runtime requests.
DialogEngine.JOPv3.FetchTimeoutMillis	5000	Integer	After GUI re-start	Timeout for DialogEngine NLU runtime request/responses in milliseconds
DialogEngine.JOPv3.ClientID		String	After GUI re-start	DialogEngine NLU model Client ID
DialogEngine.JOPv3.ClientSecret		String	After GUI re-start	DialogEngine NLU model secret password
DialogEngine.JOPv2.APIKey		String	After GUI re-start	Legacy Dialog Engine NLU API key password for request/response authentication
Resources.AllowedUploadAudioMimeTypes	audio/x-wav, audio/vnd.wave, audio/wave, audio/wav	Comma separated Strings	After GUI re-start	White list of allowed audio MIME types for GUI uploads when using Rich Media prompts.
Resources.AllowedUploadImageMimeTypes	image/jpeg, image/x-ms-bmp, image/png, image/gif	Comma separated Strings	After GUI re-start	White list of allowed image MIME types for GUI uploads when using Rich Media prompts.
Resources.AllowedUploadVideoMimeTypes	video/quicktime, video/mp4, video/3gpp, video/x-ms-wmv, video/x-msvideo, video/x-flv	Comma separated Strings	After GUI re-start	White list of allowed video MIME types for GUI uploads when using Rich Media prompts.
Resources.AllowedUploadContentTypes	audio,image,text,application/	Comma separated	After GUI re-start	Generic white list

Name	Default Value	Valid Values	Takes Effect	Description
	octet-stream,application/x-zip-compressed,application/zip,application/vnd.openxmlformats-officedocument.spreadsheetml.sheet,application/x-x509-ca-cert,video	Strings		of all allowed MIME file types to upload through the GUI
Prompts.RetryTimeoutMaxAllowed	2	Integer	After GUI re-start	Maximum number of different retry prompts allowed per individual menu/question.
GenesysSDK.ConfigServer.IsTLSEnabled	false	String: true or false	After GUI re-start	Indicates IA whether the security TLS protocol is enabled for the connection with Genesys Config Server.
GenesysSDK.PlatformServer.IsTLSEnabled	false	String: true or false	After GUI re-start	Indicates IA whether the security TLS protocol is enabled for the connection with Genesys Platform Server.
DialogEngine.JOPv2.APIKey		String	After GUI re-start	Legacy Dialog Engine API key for NLU model
DialogEngine.JOPv2.BaseURL		String	After GUI re-start	Base API URL for Legacy Dialog Engine NLU runtime
DialogEngine.JOPv2.TimeoutMillis	1000	Integer	After GUI re-start	Timeout for Legacy Dialog Engine NLU runtime request/responses in milliseconds
DialogEngine.JOPv2.Username		String	After GUI re-start	Legacy Dialog Engine NLU model username
DialogEngine.JOPv2.Password		String	After GUI re-start	Legacy Dialog Engine NLU model password
Caching.ReloadGenesysServerCaches.HttpTimeout	5000	Integer	After GUI re-start	Number of milliseconds to wait for reloading the cache on all configured

Name	Default Value	Valid Values	Takes Effect	Description
				Genesys servers for IA
Vui.SessionTimeoutInMins	30	Integer	After VUI re-start	Maximum number of minutes that each VUI session can remain open with no activity before it times out.
VoiceXML.SSML.RulesTypeOverride	Default	String		Choose which TTS engine and language is to be used. Depending on the TTS engine and language in use, GIA will attempt to pick the most specific vxml file for each playback type on the folder specified. For example, `GoogleSS`, `NuanceVocalizer5`. If no TTS engine is specified, the `Default` value is used.
VoiceXML.SSML.AlwaysUseSSML	false	String		Support for interpreting SSML.
VoiceXML.SSML.Play.DynamicSSML	false	String		Support for Dynamic SSML playback from NLU engines. When *true*, the SSML tags are interpreted as SSML tags and played. When *false*, the SSML tags are interpreted as string values .
GoogleASR.NBestValueConfidenceScoreBased	false	String		The confidence level to pick up the best match value returned by ASR for bot interactions. When set to *True*, the value that has the highest confidence score in the **nbestEntries** list is picked. When set to

Name	Default Value	Valid Values	Takes Effect	Description
				False, the first value in the **nbestEntries** list is picked without considering the confidence score.
GenesysSDK.ConfigServerRoles.UniversalFormat	false	String		Support for 2-level role permission hierarchy for GAX 9.0.100.xx when *true.* The default value of *false* avoids compatibility issues and is applicable for 3-level GAX role permissions.
GUI.Sites.ApplicationImage.MaxNestingLevels	50	Integer	After GUI re-start	When the graphical view of an application is large, an error, *Sorry, nesting level is too high in module [insert app or module name here]* is displayed. Increasing the value of the `MaxNestingLevels` field marginally will display the graphical view. Warning Increasing the value of this option can impact performance on the GUI server and could lead to OutOfMemoryErrors in the JVM. Important The **GUI.ApplicationIma** (default value 30) and **GUI.ApplicationIma** (default value 50) can also be used to control the **Solution View** modules diagram.
TurnBasedController.Active_Channel_preference	false	String: true or false	After GUI re-start	When set to `true`, GIA validates the customer input against the

Name	Default Value	Valid Values	Takes Effect	Description
				recognition phrases of the current active channel only. If the flag is set to `false`, GIA validates the customer inputs against the recognition phrases in all channels.
VuiPreferences.AvailableLanguages.ASR	en-gb,en-us,es-us,fr-fr		After GUI re-start	Add comma separated language code for ASR language support. If any new languages are added, add these following configuration options in server settings: <table><tr><th>Key</th><th>Sample Value (Update with values for the new supported languages)</th></tr><tr><td>Resources.Grammars[lang].Alph</td><td>[illegible]</td></tr><tr><td>Resources.Grammars[lang].Bool</td><td>[illegible]</td></tr><tr><td>Resources.Grammars[lang].Digi</td><td>[illegible]</td></tr></table>

Menu Contoller VXML Settings

Starting from 9.0.111.x, Intelligent Automation allows configuring the Menu Controller VXML options from the UI instead of configuring them in the **fish.properties** file.

Name	Default Value
MenuController.VXMLDefaults.speedvsaccuracy	0.5
MenuController.VXMLDefaults.bargeintype	speech
MenuController.VXMLDefaults.audiofetchhint	prefetch
MenuController.VXMLDefaults.audiomaxage	86400
MenuController.VXMLDefaults.audiomaxstale	0
MenuController.VXMLDefaults.documentfetchhint	safe

Name	Default Value
MenuController.VXMLDefaults.documentmaxage	0
MenuController.VXMLDefaults.documentmaxstale	0
MenuController.VXMLDefaults.grammarfetchhint	prefetch
MenuController.VXMLDefaults.grammarmaxage	86400
MenuController.VXMLDefaults.grammarmaxstale	0
MenuController.VXMLDefaults.objectfetchhint	prefetch
MenuController.VXMLDefaults.objectmaxage	86400
MenuController.VXMLDefaults.objectmaxstale	0
MenuController.VXMLDefaults.scriptfetchhint	prefetch
MenuController.VXMLDefaults.scriptmaxage	86400
MenuController.VXMLDefaults.scriptmaxstale	0
MenuController.VXMLDefaults.fetchaudiodelay	2000
MenuController.VXMLDefaults.fetchaudiominimum	2000
MenuController.VXMLDefaults.fetchtimeout	10000
MenuController.VXMLDefaults.universals	none
MenuController.VXMLDefaults.maxnotoconfirms	2
MenuController.VXMLDefaults.confirmation_incompletetimeout	500
MenuController.VXMLDefaults.confirmation_maxspeechtimeout	5000

Roles in Intelligent Automation

The role-based login system will be based primarily on the presence or absence of individual Permission IDs that are associated with a user. These permission IDs will be stored in the database and loaded into memory when a user logs in.

Roles are defined in the **roles** database table and each role is mapped to zero or more permissions via the **role_permissions** table.

Each role has a *level* which limits what other roles a particular admin user can assign to other users. For example, a **Company Administrator** with level 500 could create new users for their company with any level less than or equal to 500, such as **Application Designer**; they could not, however, create a user with the **Platform Administrator** role since it would have a higher level of 800. Note that only some roles have the ability to create users.

The initial roles table will look like this:

ID	Name	Level
1	CTI Agent	10
2	Reports Only User	30
3	Application Maintainer	70
4	Application Designer	200
5	Product Designer	300
6	Company Administrator	500
7	Platform Administrator	800
8	Full Administrator	1000

When you log in via the `User.login()` method, the GUI loads in all Permission IDs by joining the **users**, **roles**, and **role_permissions** tables.

An important consideration is that this set of available Permission IDs is further restricted by the Licensing Engine so, without the correct licenses, even a Full Administrator role would be unable to create new products or download raw report data.

Permission Mapping

The matrix below shows which individual permissions are assigned to each role.

Those permission IDs with an asterisk (*) are candidates for restriction via the Licensing Engine.

Some permission IDs are currently unused (legacy IDs) and are marked as such in the table. They may be re-assigned in future.

ID	Permission	CTI Agent	Report Viewer	Integration Dev	App Maintainer	App Designer	Product Designer	Company Admin	Platform Admin	Full Admin	Added in Version
1	list_users							X	X	X	Base
2	view_current_user	X	X	X	X	X	X	X	X	X	Base
3	view_other_user							X	X	X	Base
4	create_user							X	X	X	Base
5	update_current_user	X	X	X	X	X	X	X	X	X	Base
6	update_other_user							X	X	X	Base
7	delete_user							X	X	X	Base
8	switch_to_userless_company								X	X	Base
9	update_company_contact_details				X	X	X	X	X	X	Base
10	download_raw_report_data *		X		X	X	X	X	X	X	Base
11	deploy_to_production				X	X	X	X	X	X	Base
12	list_products						X		X	X	Base
13	view_product *						X		X	X	Base
14	create_product *						X		X	X	Base
15	update_product *						X		X	X	Base
16	delete_product						X		X	X	Base
17	list_standard_grammars								X	X	Base
18	view_standard_grammar								X	X	Base
19	create_standard_grammar								X	X	Base
20	update_standard_grammar								X	X	Base
21	delete_standard_grammar								X	X	Base
22	list_companies								X	X	Base
23	view_company				X	X	X	X	X	X	Base
24	create_company								X	X	Base
25	delete_company									X	Base
26	switch_company									X	Base

ID	Permission	CTI Agent	Report Viewer	Integration Dev	App Maintainer	App Designer	Product Designer	Company Admin	Platform Admin	Full Admin	Added in Version
27	list_clusters								X	X	Base
28	view_cluster								X	X	Base
29	create_cluster								X	X	Base
30	update_cluster								X	X	Base
31	delete_cluster								X	X	Base
32	list_servers								X	X	Base
33	view_server								X	X	Base
34	create_server								X	X	Base
35	update_server								X	X	Base
36	delete_server								X	X	Base
37	list_cpr_languages								X	X	Base
38	view_cpr_language								X	X	Base
39	add_or_update_cpr_language								X	X	Base
40	delete_cpr_language								X	X	Base
41	list_sites		X		X	X	X	X	X	X	Base
42	update_company_details								X	X	Base
43	create_site					X	X	X	X	X	Base
44	delete_site					X	X	X	X	X	Base
45	view_standard_reports *		X		X	X	X	X	X	X	Base
46	cti_viewer *	X	X	X	X	X	X	X	X	X	Base
47	call_monitor *					X	X	X	X	X	Base
48	view_callflow		X		X	X	X	X	X	X	Base
49	rename_block				X	X	X	X	X	X	Base
50	create_start_block					X	X	X	X	X	Base
51	view_start_block		X		X	X	X	X	X	X	Base
52	update_start_block					X	X	X	X	X	Base
53	delete_start_block					X	X	X	X	X	Base

ID	Permission	CTI Agent	Report Viewer	Integration Dev	App Maintainer	App Designer	Product Designer	Company Admin	Platform Admin	Full Admin	Added in Version
54	create_message_block				X	X	X	X	X	X	Base
55	view_message_block		X		X	X	X	X	X	X	Base
56	update_message_block				X	X	X	X	X	X	Base
57	delete_message_block				X	X	X	X	X	X	Base
58	create_menu_block				X	X	X	X	X	X	Base
59	view_menu_block		X		X	X	X	X	X	X	Base
60	update_menu_block				X	X	X	X	X	X	Base
61	delete_menu_block				X	X	X	X	X	X	Base
62	create_customquestion_block *					X	X	X	X	X	Base
63	view_customquestion_block		X		X	X	X	X	X	X	Base
64	update_customquestion_block				X	X	X	X	X	X	Base
65	delete_customquestion_block *					X	X	X	X	X	Base
66	create_script_block *					X	X	X	X	X	Base
67	view_script_block *					X	X	X	X	X	Base
68	update_script_block *					X	X	X	X	X	Base
69	delete_script_block *					X	X	X	X	X	Base
70	create_phonetransfer_block				X	X	X	X	X	X	Base
71	view_phonetransfer_block		X		X	X	X	X	X	X	Base
72	update_phonetransfer_block				X	X	X	X	X	X	Base
73	delete_phonetransfer_block				X	X	X	X	X	X	Base
74	create_urltransfer_block				X	X	X	X	X	X	Base
75	view_urltransfer_block		X		X	X	X	X	X	X	Base
76	update_urltransfer_block				X	X	X	X	X	X	Base
77	delete_urltransfer_block				X	X	X	X	X	X	Base

ID	Permission	CTI Agent	Report Viewer	Integration Dev	App Maintainer	App Designer	Product Designer	Company Admin	Platform Admin	Full Admin	Added in Version
78	create_end_block				X	X	X	X	X	X	Base
79	view_end_block		X		X	X	X	X	X	X	Base
80	update_end_block				X	X	X	X	X	X	Base
81	delete_end_block				X	X	X	X	X	X	Base
82	view_output_node		X		X	X	X	X	X	X	Base
83	update_output_node				X	X	X	X	X	X	Base
84	login	X	X	X	X	X	X	X	X	X	Base
85	view_dashboard *		X		X	X	X	X	X	X	Base
86	download_recordings		X		X	X	X	X	X	X	Base
87	test_script *					X	X	X	X	X	Base
88	view_defaults_block		X		X	X	X	X	X	X	Base
89	update_defaults_block				X	X	X	X	X	X	Base
90	view_prompts		X		X	X	X	X	X	X	Base
91	update_prompts				X	X	X	X	X	X	Base
92	add_and_delete_custom_prompts					X	X	X	X	X	Base
93	import_callflow					X	X	X	X	X	Base
94	import_callflow_audio					X	X	X	X	X	Base
95	import_data					X	X	X	X	X	Base
96	import_data_audio					X	X	X	X	X	Base
97	export_callflow					X	X	X	X	X	Base
98	export_callflow_audio					X	X	X	X	X	Base
99	export_data					X	X	X	X	X	Base
100	export_data_audio					X	X	X	X	X	Base
101	view_product_specific_editors		X		X	X	X	X	X	X	Base
102	update_product_specific_editors				X	X	X	X	X	X	Base
103	delete_product_specific_editors				X	X	X	X	X	X	Base
104	view_product_specific_reports *		X		X	X	X	X	X	X	Base

ID	Permission	CTI Agent	Report Viewer	Integration Dev	App Maintainer	App Designer	Product Designer	Company Admin	Platform Admin	Full Admin	Added in Version
105	view_cpr_prompt_list					X	X	X	X	X	Base
106	upload_cpr_prompts					X	X	X	X	X	Base
107	download_resource_files		X		X	X	X	X	X	X	Base
108	view_site_details					X	X	X	X	X	Base
109	update_site_details					X	X	X	X	X	Base
110	create_output_node				X	X	X	X	X	X	Base
111	delete_output_node				X	X	X	X	X	X	Base
112	save_callflow				X	X	X	X	X	X	Base
113	create_recording_block *					X	X	X	X	X	Base
114	view_recording_block *		X		X	X	X	X	X	X	Base
115	update_recording_block *					X	X	X	X	X	Base
116	delete_recording_block *					X	X	X	X	X	Base
117	view_settings								X	X	Base
118	update_settings								X	X	Base
119	use_shock_absorber *		X		X	X	X	X	X	X	Base
120	lock_unlock_callflow *						X		X	X	Base
121	update_locked_callflow *						X		X	X	Base
122	copy_production_site_to_test_site				X	X	X	X	X	X	Base
123	view_export					X	X	X	X	X	Base
124	view_import					X	X	X	X	X	Base
125	change_site_publicity *						X		X	X	Base
126	login_during_maintenance								X	X	Base
127	speak_tts				X	X	X	X	X	X	Base

ID	Permission	CTI Agent	Report Viewer	Integration Dev	App Maintainer	App Designer	Product Designer	Company Admin	Platform Admin	Full Admin	Added in Version
128	delete_cpr_folder					X	X	X	X	X	Base
129	test_inbound_sms_script *					X	X	X	X	X	Base
130	view_admin_menu						X		X	X	Base
131	list_inbound_sms_rules *		X		X	X	X	X	X	X	Base
132	update_inbound_sms_rule *					X	X	X	X	X	Base
133	delete_inbound_sms_rule *					X	X	X	X	X	Base
134	create_inbound_sms_rule *					X	X	X	X	X	Base
135	view_inbound_sms_rule *		X		X	X	X	X	X	X	Base
136	flush_server_settings								X	X	Base
137	list_roles								X	X	Base
138	view_role								X	X	Base
139	update_role								X	X	Base
140	create_role								X	X	Base
141	delete_role								X	X	Base
142	list_phone_number_rulesets								X	X	Base
143	view_phone_number_ruleset								X	X	Base
144	update_phone_number_ruleset								X	X	Base
145	create_phone_number_ruleset								X	X	Base
146	delete_phone_number_ruleset								X	X	Base
147	export_product *						X		X	X	Base
148	import_product						X		X	X	Base
149	list_opening_hours_rules				X	X	X	X	X	X	Base
150	view_opening_hours_rule				X	X	X	X	X	X	Base
151	update_opening_hours_rule				X	X	X	X	X	X	Base

ID	Permission	CTI Agent	Report Viewer	Integration Dev	App Maintainer	App Designer	Product Designer	Company Admin	Platform Admin	Full Admin	Added in Version
152	create_opening_hours_rule				X	X	X	X	X	X	Base
153	update_weights *					X	X	X	X	X	Base
154	export_standard_grammar								X	X	Base
155	import_standard_grammar								X	X	Base
156	create_interceptor_block *				X	X	X	X	X	X	Base
157	view_interceptor_block *		X		X	X	X	X	X	X	Base
158	update_interceptor_block *				X	X	X	X	X	X	Base
159	delete_interceptor_block *				X	X	X	X	X	X	Base
160	remember_input_block_results *				X	X	X	X	X	X	Base
161	list_outbound_campaigns *		X		X	X	X	X	X	X	3.0.1
162	create_outbound_campaigns *				X	X	X	X	X	X	3.0.1
163	update outbound_campaigns *				X	X	X	X	X	X	3.0.1
164	delete outbound_campaigns *				X	X	X	X	X	X	3.0.1
165	startstop outbound_campaigns *		X		X	X	X	X	X	X	3.0.1
166	view_callflow_in_tree		X		X	X	X	X	X	X	3.1.0
167	view_question		X		X	X	X	X	X	X	3.1.0
168	create_question				X	X	X	X	X	X	3.1.0
169	update_question				X	X	X	X	X	X	3.1.0
170	delete_question				X	X	X	X	X	X	3.1.0

ID	Permission	CTI Agent	Report Viewer	Integration Dev	App Maintainer	App Designer	Product Designer	Company Admin	Platform Admin	Full Admin	Added in Version
171	view_answer		X		X	X	X	X	X	X	3.1.0
172	create_answer				X	X	X	X	X	X	3.1.0
173	update_answer				X	X	X	X	X	X	3.1.0
174	delete_answer				X	X	X	X	X	X	3.1.0
175	change_site_callflow_editor *						X		X	X	3.1.0
176	store_input_block_results *				X	X	X	X	X	X	3.0.2
177	associate_input_block_with_task *				X	X	X	X	X	X	3.0.2
178	view_cli_data_settings								X	X	3.0.2
179	update_cli_data_settings								X	X	3.0.2
180	view_cpr_uploads				X	X	X	X	X	X	3.1.0
181	select_cpr_upload				X	X	X	X	X	X	3.1.0
182	download_cpr_upload				X	X	X	X	X	X	3.1.0
183	change_site_vertical_name						X		X	X	3.1.6
184	create_template_site *						X		X	X	3.1.6
185	create_mobileapp_block									X	3.1.6
186	update_mobileapp_block									X	3.1.6
187	view_mobileapp_block									X	3.1.6
188	delete_mobileapp_block									X	3.1.6
189	view_mobile_apps *									X	3.1.6
190	create_mobile_apps *									X	3.1.6
191	delete_mobile_apps *									X	3.1.6
192	edit_mobile_apps_flow									X	3.1.6
193	edit_mobile_apps_screen									X	3.1.6
194	lock_unlock_visualflow									X	3.1.6

ID	Permission	CTI Agent	Report Viewer	Integration Dev	App Maintainer	App Designer	Product Designer	Company Admin	Platform Admin	Full Admin	Added in Version
	*										
195	change_mobile_apps_publicity *									X	3.1.6
196	change_mobile_apps_active_status *									X	3.1.6
197	edit_mobile_apps_name									X	3.1.6
198	edit_mobile_apps_description									X	3.1.6
199	deploy_mobile_apps *									X	3.1.6
200	rollback_mobile_apps *									X	3.1.6
201	upload_theme *									X	3.1.6
202	edit_test_numbers *									X	3.1.6
203	list_mobile_apps *									X	3.1.6
204	export_callflow_grammars					X	X	X	X	X	3.1.7
205	import_callflow_grammars					X	X	X	X	X	3.1.7
206	view_customer_journey_reports *		X			X	X	X	X	X	3.1.7
207	lock_or_unlock_user							X	X	X	3.1.8
208	update_user_forcepassword_status							X	X	X	3.1.8
209	list_personas				X	X	X	X	X	X	3.2.0
210	view_persona				X	X	X	X	X	X	3.2.0
211	edit_persona					X	X	X	X	X	3.2.0
212	create_persona					X	X	X	X	X	3.2.0
213	delete_persona					X	X	X	X	X	3.2.0
214	assign_personas_to_site					X	X	X	X	X	3.2.0
215	reset_company_health_date *				X	X	X	X	X	X	3.2.0

ID	Permission	CTI Agent	Report Viewer	Integration Dev	App Maintainer	App Designer	Product Designer	Company Admin	Platform Admin	Full Admin	Added in Version
216	display_health_error_thresholds *				X	X	X	X	X	X	3.2.0
217	update_health_error_thresholds *				X	X	X	X	X	X	3.2.0
218	create_mobile_start_block *					X	X	X	X	X	3.2.0
219	update_mobile_start_block *					X	X	X	X	X	3.2.0
220	delete_mobile_start_block *					X	X	X	X	X	3.2.0
221	view_mobile_start_block *		X		X	X	X	X	X	X	3.2.0
222	create_mobile_settings_block *					X	X	X	X	X	3.2.0
223	update_mobile_settings_block *					X	X	X	X	X	3.2.0
224	delete_mobile_settings_block *					X	X	X	X	X	3.2.0
225	view_mobile_settings_block *		X		X	X	X	X	X	X	3.2.0
226	create_mobile_tabs_block *					X	X	X	X	X	3.2.0
227	update_mobile_tabs_block *					X	X	X	X	X	3.2.0
228	delete_mobile_tabs_block *					X	X	X	X	X	3.2.0
229	view_mobile_tabs_block *		X		X	X	X	X	X	X	3.2.0
230	create_mobile_link_block *					X	X	X	X	X	3.2.0
231	update_mobile_link_block *					X	X	X	X	X	3.2.0
232	delete_mobile_link_block					X	X	X	X	X	3.2.0

ID	Permission	CTI Agent	Report Viewer	Integration Dev	App Maintainer	App Designer	Product Designer	Company Admin	Platform Admin	Full Admin	Added in Version
	*										
233	view_mobile_link_block *		X		X	X	X	X	X	X	3.2.0
234	create_mobile_screen_block *					X	X	X	X	X	3.2.0
235	update_mobile_screen_block *					X	X	X	X	X	3.2.0
236	delete_mobile_screen_block *					X	X	X	X	X	3.2.0
237	view_mobile_screen_block *		X		X	X	X	X	X	X	3.2.0
238	view_mobile_apps_details *					X	X	X	X	X	3.2.0
239	update_mobile_apps_details *					X	X	X	X	X	3.2.0
240	view_mobileflow *		X		X	X	X	X	X	X	3.2.0
241	save_mobileflow *				X	X	X	X	X	X	3.2.0
242	update_locked_mobileflow *						X		X	X	3.2.0
243	create_template_mobile_apps *						X		X	X	3.2.0
244	create_mobile_logon_block *					X	X	X	X	X	3.2.0
245	update_mobile_logon_block *					X	X	X	X	X	3.2.0
246	delete_mobile_logon_block *					X	X	X	X	X	3.2.0
247	view_mobile_logon_block *		X		X	X	X	X	X	X	3.2.0
248	view_company_health_details *		X		X	X	X	X	X	X	3.2.0

ID	Permission	CTI Agent	Report Viewer	Integration Dev	App Maintainer	App Designer	Product Designer	Company Admin	Platform Admin	Full Admin	Added in Version
249	view_demo_menu *									X	3.2.1
250	populate_dashboard_demo_data *									X	3.2.1
251	delete_dashboard_demo_data *									X	3.2.1
252	view_cli_data_records		X			X	X	X	X	X	3.2.2
253	update_cli_data_records					X	X	X	X	X	3.2.2
254	clear_cli_data_records					X	X	X	X	X	3.2.2
255	view_integration			X	X			X	X	X	3.2.2
256	create_integration			X				X	X	X	3.2.2
257	update_integration			X				X	X	X	3.2.2
258	delete_integration			X				X	X	X	3.2.2
259	export_excel_prompt_list		X		X	X	X	X	X	X	3.2.2
260	import_excel_prompt_list				X	X	X	X	X	X	3.2.2
261	call_monitor_generate_cyara_testcase					X	X	X	X	X	3.2.2
262	list_certificates								X	X	3.2.3
263	update_certificate								X	X	3.2.3
264	delete_certificate								X	X	3.2.3
265	create_certificate								X	X	3.2.3
266	view_certificate								X	X	3.2.3
267	generate_certificate								X	X	3.2.3
268	virtual_call					X	X	X	X	X	3.2.3
269	edit_site_parameters					X	X	X	X	X	3.2.4
270	list_visual_apps		X		X	X	X	X	X	X	3.2.4
271	create_visual_apps					X	X	X	X	X	3.2.4
272	delete_visual_apps					X	X	X	X	X	3.2.4
273	view_visual_apps_details		X		X	X	X	X	X	X	3.2.4
274	update_visual_apps_details					X	X	X	X	X	3.2.4
275	list_themes		X		X	X	X	X	X	X	3.5.0

ID	Permission	CTI Agent	Report Viewer	Integration Dev	App Maintainer	App Designer	Product Designer	Company Admin	Platform Admin	Full Admin	Added in Version
276	create_themes					X	X	X	X	X	3.5.0
277	edit_themes					X	X	X	X	X	3.5.0
278	delete_themes					X	X	X	X	X	3.5.0
279	view_company_authentication_key					X	X	X	X	X	3.5.0
280	view_indy_sharedscript			X	X	X	X	X	X	X	9.0.003.00
281	create_indy_sharedscript			X		X	X	X	X	X	9.0.003.00
282	update_indy_sharedscript			X		X	X	X	X	X	9.0.003.00
283	delete_indy_sharedscript			X		X	X	X	X	X	9.0.003.00
284	view_indy_httpsettings			X	X	X	X	X	X	X	9.0.003.00
285	create_indy_httpsettings			X		X	X	X	X	X	9.0.003.00
286	update_indy_httpsettings			X		X	X	X	X	X	9.0.003.00
287	delete_indy_httpsettings			X		X	X	X	X	X	9.0.003.00
288	view_indy_envsettings			X	X	X	X	X	X	X	9.0.003.00
289	create_indy_envsettings			X		X	X	X	X	X	9.0.003.00
290	update_indy_envsettings			X		X	X	X	X	X	9.0.003.00
291	delete_indy_envsettings			X		X	X	X	X	X	9.0.003.00
292	create_indy_deployment			X		X	X	X	X	X	9.0.003.00
293	delete_indy_deployment			X		X	X	X	X	X	9.0.003.00
294	rollback_indy_deployment			X		X	X	X	X	X	9.0.003.00
295	import_indy_integration			X		X	X	X	X	X	9.0.003.00
296	export_indy_integration			X		X	X	X	X	X	9.0.003.00
297	indy_deploy_to_production			X	X	X	X	X	X	X	9.0.003.00
298	view_indy_jdbcconfig			X	X	X	X	X	X	X	9.0.003.00
299	create_indy_jdbcconfig			X		X	X	X	X	X	9.0.003.00
300	update_indy_jdbcconfig			X		X	X	X	X	X	9.0.003.00
301	delete_indy_jdbcconfig			X		X	X	X	X	X	9.0.003.00
302	configure_number_entry_retry_timeout_prompts					X	X	X	X	X	9.0.101.00
303	update_protected_callflow										9.0.104
304	view_phrasehints				X	X	X	X	X	X	9.0.108

ID	Permission	CTI Agent	Report Viewer	Integration Dev	App Maintainer	App Designer	Product Designer	Company Admin	Platform Admin	Full Admin	Added in Version
305	create_phrasehint_group					X	X	X	X	X	9.0.108
306	delete_phrasehint_group					X	X	X	X	X	9.0.108
307	create_phrasehint_upload					X	X	X	X	X	9.0.108
308	delete_phrasehint_upload					X	X	X	X	X	9.0.108
309	download_phrasehints				X	X	X	X	X	X	9.0.108

Document Change History

This page summarizes the changes made in this document for release 9.0.0.

Changes in release 9.0.004.01

- New section User Authentication and Authorization describing how to allow users created in Genesys Administrator and stored in the Configuration Database to access Intelligent Automation, and how passwords for these users can be changed by Intelligent Automation.
- Applied standard template to front page.

Changes in release 9.0.003.00

There were no changes in this document for release 9.0.003.00.

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. Some folders and path names might still include GAAP and/or SpeechStorm, but these will be rebranded as the product evolves.
- The new page Applying CSS to WebIVR MicroApps describes the CSS classes that are rendered into HTML at runtime for WebIVR MicroApps.
- The new page Using Bots provides a description of how Intelligent Automation provides chat bot-based access WebIVR applications.
- The new page SNMP Traps provides an introduction to SNMP traps and lists the traps that are currently available within Genesys Intelligent Automation.
- Hive off recovery processes are described in the new pages Hive Off Recovery Process for Oracle and Hive Off Recovery Process for SQL Server.