



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 Help

[Interface Overview](#)

12/14/2025

Contents

- [1 Interface Overview](#)
 - [1.1 Terminology](#)

Interface Overview

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

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

Terminology

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

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

Applications, modules and menus

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

Applications

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

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

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

Modules

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

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

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

Menus

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

Tip

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

Blocks

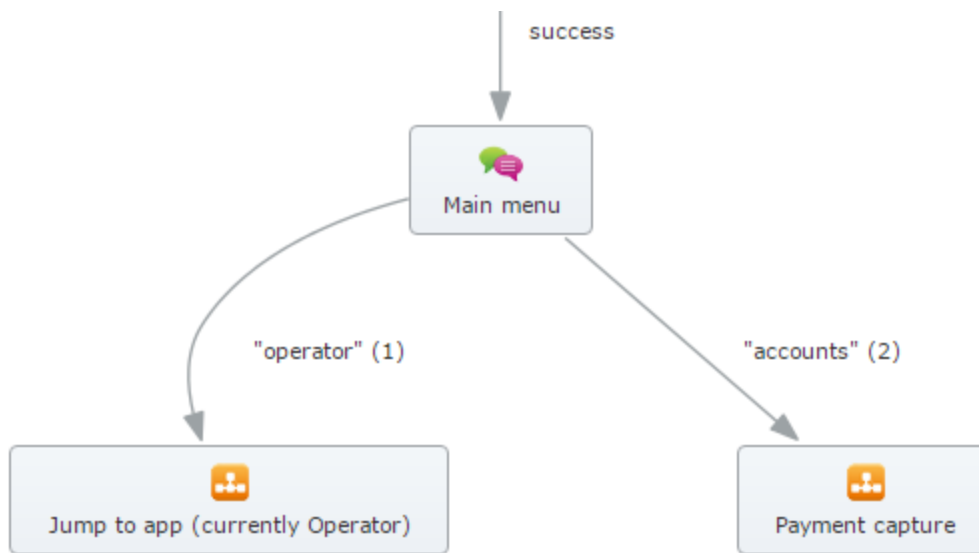
Each individual step in an application callflow is known as a *block*. Each block is represented by a rectangle on the callflow diagram in the [Callflow Editor](#) page.

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

Paths

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

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



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

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

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

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

Company

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

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