



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

Integration

5/4/2025

---

## Contents

- [1 Integration](#)
  - [1.1 Benefits of iHub](#)
  - [1.2 Terminology](#)

# Integration

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

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

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

## Benefits of iHub

iHub offers the following benefits when integrating systems:

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

## Limitations

iHub does have the following limitations:

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

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

## Terminology

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

- **Author**
- **Process**
- **Library**
- **Environment**

### Author

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

### Process

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

### Integration Script

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

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

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

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

For a list of the helper methods, see [Scripting Commands](#).

### Response Template

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

### Process Endpoint

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

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

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

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

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

### Library

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

### Environment

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

You cannot export environment settings, including:

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