



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.

Chat Server Administration Guide

Sample Workflow Deployment

12/19/2025

Sample Workflow Deployment

Use the following procedure to deploy the workflow sample:

1. Stop Interaction Server.
2. Upgrade the Interaction Server Database - execute script `AsyncChatSample.sql` from “workflow” subdirectory of Chat Server working directory.
Note: SQL script provided for Microsoft SQL Server only. You need to adjust it accordingly if you are using a different database for Interaction Server.
3. Create new interaction custom properties; these are used in the workflow sample for **view** conditions. In Configuration (using GAX):
 - Navigate to the **Business Attributes** folder and create new **Business Attribute** with:
 - type - Custom
 - name - InteractionCustomProperties
 - display name - Interaction Custom Properties
 - Navigate to **Attribute Values** and create the following values (**Note:** that **name** and **display name** must be the same and each value must have a *translation* section in the **Annex** tab with a “translate-to” option):
 - GCTI_Chat_AsyncStatus with translate-to=async_status
 - GCTI_Chat_AsyncCheckAt with translate-to=async_check_at
4. Start Interaction Server.
5. In Interaction Routing Designer (IRD), import `AsyncChatSample.wie`, located in **workflow** sub-directory of Chat Server working directory, and activate strategies.
6. Connect Chat Server endpoint with `async-chat-greet-queue` queue.
7. In Workspace Desktop Edition application configuration options (see [Configuring Asynchronous Chat in Workspace Desktop Edition](#) for more information), set:
 - value `chat.on-hold-queue` to `async-chat-return-queue`
 - value `workbin.chat.on-hold` to `async-chat-main-workbin`

Testing

While Genesys Chat Widget only supports async mode for CometD connections, the testing of back-end components (Chat Server, workflow and Agent Desktop) can be done using the Chat Widget without enabling CometD. Launch the Chat Widget with the following userdata:

```
{ GCTI_Chat_AsyncMode: "true"
}
```

Additionally you can provide the following key-value pairs:

Value	Description
Chat_Async_RoutingTimeout: "5"	allow to decrease routing wait time to 5 seconds (default value in workflow is 120 seconds).
Chat_Async_WorkflowDebug: "true"	forces workflow to send debug chat messages about workflow execution.