Chat Server Administration Guide

Chat Business Process Sample

4/24/2026

Contents

Chat Business Process Sample

Overview

Chat Business Process (BP) Sample provides a sample workflow (in other words, a set of URS strategies combined into a business process) which demonstrates how to process chat interactions both for regular and asynchronous chat for different channels, including web chat, Apple Business Chat (ABC), WhatsApp. Chat BP Sample is not intended to be used in production deployments without careful evaluation and adjustments according to customer business requirements. While initially Chat BP Sample was developed for async chat only (which explains the naming convention for queues and strategies), it was adjusted to support regular chat as well.

Deployment

To deploy Chat BP Sample, following must be done:

1. Stop Interaction Server.
2. Upgrade the Interaction Server Database - execute script `AsyncChatSample_<database name>.sql` from “script” folder of Chat BP Sample IP.
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 `ChatBusinessProcessSample.wie`, located in “workflow” folder of Chat BP Sample IP, 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 `<media-type-name>.on-hold-queue` to `async-chat-return-queue` (where `<media-type-`

name> must be replaced with a media type value such as chat, or whatsappsession.

- value workbin.chat.on-hold to async-chat-main-workbin.

Testing through web chat

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"	Allows the routing wait time to decrease to 5 seconds (default value in workflow is 120 seconds).
Chat_Async_WorkflowDebug: "true"	Forces the workflow to send debug chat messages about workflow execution.
Chat_Async_SendRichMessage: "true" **Introduced in: 8.5.309.12**	Forces the workflow to send the welcome rich message, "Welcome to Genesys chat".

Functional description

Functionally, Chat BP Sample can be divided into the following parts:

1. **Initialization.** The strategy (async-chat-greet-strategy):
   a. Sends greeting message to a customer
   b. Initializes routing parameters either for async or regular chat
   c. Moves interaction for processing into the main (async-chat-main-queue) queue
2. **Routing preparation and routing.** The strategy (async-chat-main-strategy):
   a. Checks if the routing rules are already assigned to the interaction (Chat_Async_LastAgentAttempts key/value pair in userdata)
   b. Tries to route to last handling agent for the specified number of attempts
   c. Continues routing to any available agent as soon as attempts are exhausted
3. **Processing interactions which are placed on hold by an agent.** The strategy (async-chat-return-strategy):
   a. Resets routing rules (Chat_Async_LastAgentAttempts)
   b. Tries to find last agent ID and place the interaction into agent's workbin (async-chat-main-workbin)

   c. Otherwise places the interaction into the main queue

4. **Processing stuck async interactions.** This happens only in very rare cases when chat session processing may get stuck due to failed components. The strategy (`async-chat-stuck-strategy`):
   a. Checks if chat session is still alive (by sending `GetSessionInfo` request) and places the interaction back into the main queue with a special async status value (to resume the routing)
   b. Otherwise moves the interaction to stop queue
5. **Implements transfer and conference routing.** The strategy (`async-chat-x-intercom-strategy`):
   a. Detects the type of target (from `IW_RoutingBasedTargetType`): **Skill**, **AgentGroup**, **InteractionQueue**.
   b. Routes the interaction according to detected type.
   c. Demonstrates how to send the ESP message correctly from the internal auxiliary (`InteractionSubtype=InternalConferenceInvite`) interaction.

> **Important**
>
> `async-chat-x-intercom-strategy` is introduced in **8.5.309.12**; the required Chat Server version is **8.5.312.10** or higher.

6. **Stopping of the completed chat**. The strategy (`async-chat-stop-strategy`):
   a. Stops the interaction
   b. Notifies UCS

Important notes

- The main processing queue (`async-chat-main-queue`) contains several views (used for selecting interaction for the processing by a strategy) with different conditions:
  - `async-chat-main-proc-view` fetches chat sessions which are not in a dormant state, thus locking the dormant chat sessions in the queue until a qualified event changes the status (these events can be a new interaction, a message from a customer, idle control notification, agent desktop failure, and others).
  - `async-chat-main-check-view` fetches stuck chat sessions.
  - `async-chat-main-stop-view` fetches chat interactions for which sessions were stopped by Chat Server while the interaction was sitting in the queue.
- In this Chat BP Sample, the agent workbin is associated with the main routing queue thus forcing the router to apply the same processing rules to workbin interactions. If you need to keep interactions in the agent workbin without forcing routing upon a qualified event in a chat session, you need to modify the workflow accordingly.
- In case of failure, strategies move the interaction to stop queue (`async-chat-stop-queue`). However, in production the workflow must consider making a few attempts before finally surrendering the interaction processing any further. For example, workflow may send an ESP message a few times during a Chat Server switchover and when chat session is being moved to another Chat Server. Chat BP Sample does not provide this logic for simplification reasons.

- Chat BP Sample contains `async-chat-media-proc-sub` subroutine which provides channel-specific logic for interaction processing. For ABC and WhatsApp channels, the strategy verifies if a customer's nickname ("_msg_ProfileNickname") is already assigned for the contact, and assigns the default nickname if needed.
- You can send Rich Text Messages from the workflow. For more information, see the **Example** in the "Key-value collection format specification" section of the How to send ESP requests to Chat Session from Workflow topic.