Workspace Desktop Edition Deployment Guide

Chat

12/22/2025

Chat

[**Modified:** 8.5.108.11, 8.5.113.11, 8.5.115.17, 8.5.118.10, 8.5.122.08, 8.5.128.07, 8.5.132.05, 8.5.140.08, 8.5.142.05, 8.5.145.06]

Workspace employs the following Chat privileges for all Chat interactions:

- Can Use Chat Media
- Can Decline Chat
- Can Release
- Can One Step Transfer
- Can One Step Conference
- Can Push Url
- Can Set Interaction Disposition
- Show Silent Monitoring
- Can Preview Customer Typing [**Added:** 8.5.108.11]
- Chat - Can Transfer File From File System [**Added:** 8.5.115.17]
- Chat - Can Transfer File From Standard Response [**Added:** 8.5.115.17]
- Chat - Can Save Attached files [**Added:** 8.5.115.17]

> ## Warning
>
> If the Chat - Can Preview Customer Typing privilege is also granted, and you have configured eServices to hide sensitive personal information that is entered by the contact during the chat, agents will be able to see the information as it is entered, but not after the contact sends it.
> [**Added:** 8.5.108.11]

Workspace displays a "Contact is typing a message" notice when the contact begins to type a reply message in your web site chat interface. The Chat - Can Preview Customer Typing privilege enables agents to see what the contact is typing before the contact clicks **Send**. This feature enables agents to anticipate their reply and therefore respond more quickly. However, Genesys recommends that you train your agents to wait until the contact has sent their message before responding to it. The text that is being typed by the contact is updated according to the updates that the web site sends to the Web API and Chat Server; therefore, your web site designer is responsible of the refresh rate of the content.

Chat server can be configured to disconnect a chat upon inactivity of engaged parties. Workspace warns the agents when a chat is about to expire due to inactivity and notifies when the session is automatically closed because of inactivity. [**Added:** 8.5.109.25]

Workspace notifies the agent through an audio alert or a visual alert when

another party is leaving (automatically, due to inactivity, or manually) or joining the chat session while the focus is not on the Workspace application or is on another area of Workspace.
Visual notification added: 8.5.113.11

The Chat feature supports spelling check. Refer to Spelling tab for information about configuring Spelling Check.

For Genesys Widget customers, Workspace supports Rich Messaging. [**Added: 8.5.142.05**]

You use the following options in the `interaction-workspace` section to configure Chat interactions:

- chat.simple-transcript: Specifies whether the chat transcript is displayed as simple lines of text or as colored blocks of text. [**Added:** 8.5.122.08]
- options.record-option-locally-only: Specifies whether the display settings for the agent are stored locally or in the agent annex.
- chat.pending-response-to-customer: Defines two alarm thresholds, in seconds, that warn agents that they have a pending response to a chat. Three levels are displayed: below the warning time, between the warning time and the maximum time, and above the maximum time. Agents are warned by the flashing of various elements in the user interface, including the taskbar, collapse/expand button, the interaction bar, and the pending response timer. If the agent places his or her mouse pointer on any of these flashing elements, a preview of the last received message from the contact is displayed.
- chat.show-unread-notification: Specifies whether the unread message notification is displayed in the chat transcript. When the message is read, the notification icon disappears. [**Added:** 8.5.122.08]
- chat.toast-information-key: Specifies whether the Information area is displayed in the Chat interaction notification. The option specifies the name of the attached data key that contains the information.
- chat.typing-isenabled: Specifies whether typing notification is enabled. It should be disabled for Chat Server lower than 8.0.1.
- chat.typing-timeout: Specifies the duration, in seconds, that the typing notification is displayed after the last keystroke and before the agent or contact sends their message.
- chat.new-message-bell: Specifies the new Chat sound configuration string.
- chat.reconnect-attempts—Defines the number of attempts to reconnect to the chat session. This applies to environments that implement Chat High Availability (HA) but also to simple environments if network disconnection occurs during a chat session.
- chat.reconnect-timeout—Defines the interval between each attempt to reconnect to the chat session. This applies to environments that implement Chat High Availability (HA) but also to simple environments if network disconnection occurs during a chat session.
- chat.nickname—Specifies that a nickname (pseudonym) is used in chat sessions instead of the agent's user name, and defines the nickname.
- display-format.chat-agent-name—Specifies the display format of agent identifiers in agent and team supervisor views.
- display-format.chat-customer-name—Specifies the display format of contact identifiers in agent and team supervisor views. [**Added:** 8.5.145.06]
- chat.auto-answer: Specifies whether a chat interaction is automatically answered when it is routed to an agent. This option can be overridden by a routing strategy. You can also configure auto-answer to display a timer that enables an agent to view case information before the interaction is automatically answered by using the chat.auto-answer.timer and chat.auto-answer.enable-reject options [**Added:**

8.5.105.12].

- chat.historical.maximum-age: Specifies the number of days of previous chat sessions with the current contact are to be displayed in the Chat interaction view before the current chat session. This reduces the need for agents to open the contact history to find previous chat interactions. Many chat sessions are conducted on mobile devices, meaning that the likelihood of timeout is very high. If a chat is resumed after a timeout, the agent sees the content of the previous sessions. [**Added:** 8.5.122.08]
- chat.rich-media-widget-width: Specifies the width, in pixels, of Rich Media in a chat interaction. The value of this option affects the minimum width of the Chat transcript view [**Added:** 8.5.150.06].
- chat.transcript-enable-history-filters: Specifies that the value specified for the contact.history.filters-<attribute> option is used to filter the history-based part of the chat transcript. Keys and values of the option are constructed like those of the `contact.history.filters-<attribute>` option. You can add these options to a routing strategy [**Added:** 8.5.132.05].
- chat.transcript-message-text-direction: Specifies whether messages in the chat transcript are displayed with a left-to-right (default) or a right-to-left reading layout. Use this option for chat interactions where contacts are using a right-to-left reading language [**Added:** 8.5.140.08].

Provisioning the Chat channel

[**Modified:** 8.5.115.17, 8.5.128.07]

1. Enabling an agent to chat with a contact

Purpose:

To enable an agent to use the Chat channel to chat with a contact whose information is stored in Universal Contact Server (UCS).

Prerequisites

- Genesys Administrator 8.0.2 or higher, configured to show Advanced View.
- A working knowledge of Genesys Administrator 8.
- A Workspace `Application` object exists in the Configuration Database.
- Workspace has a connection to Universal Contact Server, Chat Server, and Interaction Server.
- The Procedure: Enabling agents to manage contacts.

Start

1. Allow the Chat privileges (see Chat Access Privileges) for the role to which the agent is assigned (refer to the Procedure: Creating a Role and allowing a Workspace privilege and assigning a Role to an agent or agent group):
   - Can Use Chat Media
   - Can Decline Chat
   - Can Release
   - Can One Step Transfer

- Can Two Step Transfer
- Can One Step Conference
- Can Two Step Conference
- Can Set Interaction Disposition
- Can Preview Customer Typing [**Added:** 8.5.108.11]

2. Configure the Chat options in the `interaction-workspace` section of the Workspace `Application` object (refer to the Chat configuration option reference for a list of Chat options and a description of how to configure them).

3. [**Added:** 8.5.128.07] Configure the Chat options that control how Chat interactions are marked as done. When an agent manually marks a Chat interaction as done, a new Chat interaction can be directed to the agent. Sometimes agents do not mark interactions done to avoid getting new interactions, which affects the focus time. To have Workspace automatically mark a Chat as done, use the following configuration options:

   - chat.auto-mark-done-owner-agent: When set to `true`, the Chat interaction is automatically closed and marked as done as soon as the last agent (the owner) ends the Chat session or the contact disconnects. To prevent the Chat interaction from closing immediately — for example to allow for after-call work — you can specify how long the interaction window stays open by using the chat.auto-mark-done-owner-agent.timer option.
   - chat.auto-mark-done-non-owner-agent: When set to `true`, the Chat interaction on the non-owner's desktop is automatically closed and marked as done as soon as the non-owner leaves the Chat session or the contact disconnects. To prevent the Chat interaction from closing immediately, for example to allow for after-call work, specify how long the interaction window stays open by using the chat.auto-mark-done-non-owner-agent.timer option.

   If the value of the interaction.disposition.is-mandatory option is `true`, then an agent must specify an outcome for the interaction before it can be closed. In this case, the auto-close options are overridden and the agent must click **Done** to close the interaction.

   If the value of the `interaction-workspace/mandatory` option is set to `true`, then an agent must edit the case data for the interaction before it can be closed. In this case, the auto-close options are overridden and the agent must click **Done** to close the interaction.

4. To keep chats open after the last agent leaves the session, which enables an agent to rejoin the session until the session is marked as done, set up Asynchronous chat [**Added:** 8.5.128.07]. Allow the following Chat privileges:

   - Chat - Can Place On Hold
   - Chat - Can Release Async
   - Chat - Can Release

   Configure the following options according to your environment:

   - chat.on-hold-queue
   - keyboard.shortcut.interaction.chat.hold

5. You can specify which Chat Server messages are included as part of the chat transcript in the interaction history. To include notices about inactivity timeout:

   - Set up the Chat Server inactivity control configuration options.
   - Set the value of the transcript-save-notices Chat Server option to `selective2`. [**Added:** 8.5.115.17]

End

2. Enabling Chat HA

Purpose:

To enable Chat for High Availability (HA).
Prerequisites

- Genesys Administrator 8.0.2 or higher, configured to show Advanced View.
- A working knowledge of Genesys Administrator 8.
- A Workspace `Application` object exists in the Configuration Database.
- Workspace has a connection to Universal Contact Server and Interaction Server.
- Chat Server 8.1.0 or higher.
- The Procedure: Enabling agents to manage contacts.

> **Important**
>
> Chat Server 8.1.0 Limitation: Do not use the application cluster to connect the Web API Server Application to Chat Server.

Start

1. Configure Chat Server for Warm Stand-by. Set the following values for the following Chat Server options for both the primary and backup Chat Servers:
   - `session-restoration-mode = simple`
   - `transcript-auto-save = 2`
   - `transcript-resend-attempts = 10`
   - `transcript-resend-delay = 15`
   - `transcript-save-on-error = continue`

   Refer to the EServices documentation for more information on setting up Chat Server.
2. For the Web API Server application, add a connection to the primary Chat Server.
3. Configure the following options in the `interaction-workspace` section of the Workspace Application object:
   - `chat.reconnect-attempts`--Defines the number of attempts to reconnect to the chat session.
   - `chat.reconnect-timeout`--Defines the interval between each attempt to reconnect to the chat session.

   Refer to the Chat configuration option reference for information about how to configure these options.

> **Tip**
>
> For information about configuring Load Balancing and Business Continuity, refer to Runtime Connection Logic in the eServices Load Balancing Business Continuity section of the *Business Continuity and Disaster Recovery* topic. [**Added:** 8.5.109.16]

End

Configuring Chat Conference and Consultation with a Skill, Group, or Interaction Queue

Agents can use the Team Communicator to find an "Instant Chat Conference" and "Start Chat Consultation" target based on a skill, group, or interaction queue instead of searching for a specific individual or DN. The system router finds the next available target from a list of targets based on the skill, group, or interaction queue that is selected by the agent. A Business Process tries to route the call based on attached data. You can configure the contact attempt interval and the number of attempts to find an available target with the specified skill or in the specified agent group or interaction queue before the request times out. The requesting agent is informed if the request has timed out.

The following attached data keys are populated by Workspace:

- InternalConferenceInviteOwnerId—The `employeeId` of the agent who is requesting the conference or consultation.
- InternalConferenceInviteOwnerInteractionId—The `Interaction Id` of parent Interaction.

To enable this feature, allow the following privilege:

- Can One Step Conference

To configure the features of the Chat conference or consultation with a skill, an agent group, or an interaction queue, set the following configuration options:

- Set the value of the intercommunication.chat.conference.invite-timeout to specify the length of the interval before the conference invitation times out.
- Set the value of the intercommunication.chat.queue to the name of the interaction queue that is used by the routing based feature for chat.

Receiving Files from Contacts

[**Added:** 8.5.115.17]

Workspace lets your agents handled files that have been sent by contacts in chat interactions. By

default, agents enabled for chat can receive files, open them in an appropriate application, and print them (if applicable). You can configure Workspace to enable agents to save files on their local workstation that have been sent through chat by a contact. To enable this feature, allow the following privilege:

- Chat - Can Save Attached files

Configure the chat.show-attachment-image-thumbnail option to specify whether to display the icon of an attached image file in chat transcript as image thumbnail or as generic file) in the online session, the interaction history, or both.

Files are stored in the Contact History, so attached files can also be opened or saved from the Contact History view.

> **Warning**
> Workspace does not provide virus checking of files sent to your agents. You should have anti-virus software running on agent workstations to prevent infection from opening or saving infected files.

Controlling attachment read-only behavior

[**Added:** 8.5.118.10]

By default, all attachments opened by agents in an external program are read-only. This means that agents cannot update them and save the changes to their hard drive.

Use the general.writable-downloaded-attachment-file-types option to override this behavior for specific file types. Allowing agents to edit only certain file types preserves the data integrity of files that you do not want agents to modify. For example, you might allow agents to modify `.jpg` and `.png` files so that the orientation can be changed, but restrict the modification of `.docx`, `.xlsx`, and other file types. Or, you might want to ensure that only `.xlsx` files can be updated by agents.

Transferring files to contacts

[**Added:** 8.5.115.17]

You can enable your agents to transfer files to contacts in a chat interaction. You can specify whether agents can transfer files from their local workstation, from the Standard Response Library, or both. To enable this feature, allow one or more of the following privileges:

- Chat - Can Transfer File From File System
- Chat - Can Transfer File From Standard Response

A toolbar in the chat composition field enables agents to choose files from their local workstation.

Files that are transferred to a contact are also stored in the Contact History, so attached files can also

be opened or saved from the Contact History view.

Use the following configuration options to control the behavior of this feature:

- chat.max-file-size: Specifies the maximum size of the file, in kilobytes, that an agent can attach to a chat interaction.
- chat.max-attachments-size: Specifies the total number of megabytes of files that an agent can attach to a chat interaction.
- chat.max-attachments-files: Specifies the maximum number of files that an agent can attach to a chat interaction.
- chat.restricted-attachment-file-types: Specifies the list of file types (for example: `exe`, `bat`, and so on) that agents cannot attach to a chat interaction.
- chat.attachment-download-timeout: Specifies the time to wait for Universal Contact Server to download a chat attachment.
- chat.show-attachment-image-thumbnail: Specifies whether to display the icon of an attached image file in chat transcript (as image thumbnail or as generic file) in the online session, the interaction history, or both.

Enabling emojis

[**Added:** 8.5.115.17]

Prerequisites:

- Chat Server deployed in UTF-8 mode
- Interaction Server to handle attached data
- Universal Contact Server deployed in UTF-8 mode

If you want your agents to be able to send and receive *emojis* during chat interactions you must perform the following actions:

1. Allow the Chat - Can Use Emojis privilege.
2. Set the value of the chat.emojis-business-attribute option to the name of a Business Attribute that defines the emojis that you want to support.
3. Set the value of the gui.emoji-font-name option to specify which font should be used to display received emojis.

In chat interactions, emojis are handled as Unicode characters, which systems recognize and translate into an image. On computers, the image that is displayed is dependent on the font being used. Not all fonts support emojis and not all fonts support all emojis. On mobile devices, the image that is displayed is depending on the image set of the platform. Each mobile device substitutes the Unicode character with an image.

The list of emojis can be found at http://unicode-table.com/en/blocks/emoticons/. Click a Unicode character in the table to view how it looks on different platforms. For example: 1F601. Another

resource for emojis can be found at http://unicode.org/emoji/charts/full-emoji-list.html.

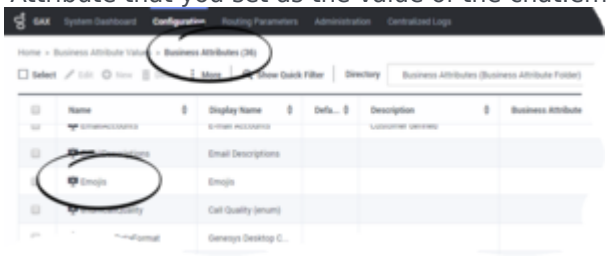
If an emoji is not supported by the font that you are using in your environment, it will be replaced by a rectangle in the chat transcript.

> **Tip**
> For more information about emojis in Genesys solutions, see this article.

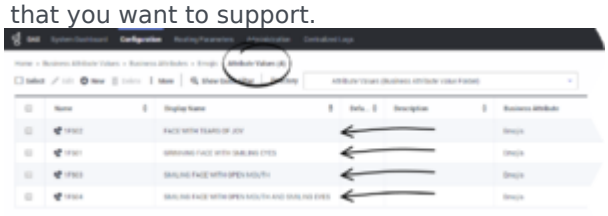
Creating the Business Attribute

Use the following steps to create a Business Attribute that defines the emojis that your agent can send:

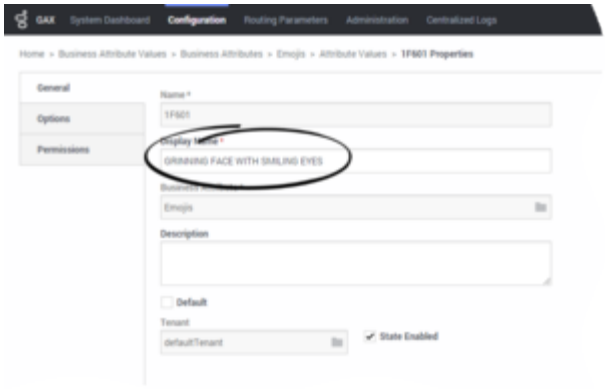
1. Create a new Business Attribute of type `Custom` with a name such as `Emojis`. This is the Business Attribute that you set as the value of the chat.emojis-business-attribute option.



2. In the **Attribute Value** tab of the Business Attribute, create one value for each emoji Unicode character that you want to support.



3. Name each value with a unique name, such as the code value of the Unicode character that you want to support. The `Display Name` that you define is displayed in a tooltip when an agent hovers their mouse pointer over an emoji before selecting it from the chat text field tools.



4. For each attribute value, configure the `interaction-workspace\code` option and assign to it the value of the emoji Unicode character. The value of the Unicode emoji format is 1F6nn where nn are digits in hex format; for example: 1F607