



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

Functional capabilities of chat protocol

Contents

- 1 Functional capabilities of chat protocol
 - 1.1 Direct Messages
 - 1.2 Enhancing security when joining a chat session
 - 1.3 Chat bot participant special treatment
 - 1.4 Notifications about detected and masked out PII data
 - 1.5 Read confirmation notice
 - 1.6 Nickname change
 - 1.7 Using rich messaging

Functional capabilities of chat protocol

This page describes various Chat Server protocol elements which can be used in the implementation of custom agent desktop applications.

Direct Messages

Chat Server can send so called direct (or private) messages and notices to a participant in chat session. Only chat basic protocol provides such functionality. In order to send a message or a notice which will be visible only to a certain participant in chat session, **ReceiverId** in methods **RequestMessage** and **RequestNotify** (defined in `Genesyslab.Platform.WebMedia.Protocols.BasicChat.Requests`) must be initialized with the `userId` of the intended participant (which can be obtained from the transcript event). In this case, only two participants will see this message in the transcript: the sender and the recipient.

When receiving a direct message, the transcript will contain either **MessageInfo** or **NoticeInfo** (defined in `Genesyslab.Platform.WebMedia.Protocols.BasicChat`) with corresponded **ReceiverId**.

Supported:

Chat Server	PSDK	Workspace (both) Edition	Chat Widget	GMS
8.5.108	8.5.1x	not supported	not supported	not supported

Enhancing security when joining a chat session

Using configuration option `session-password-enforce`, it is possible to force Chat Server to generate the crypto-random security token (we call it "session password") which will be associated with a chat session during its creation. In this case, Chat Server will require this session password each time a new participant sends a request to join an existing chat session (it must be provided in **GCTI_Chat_SessionPassword** key/value pair in userdata of **RequestJoin**). Chat Server attaches the session password to the userdata of the interaction (submitted to Interaction Server) in **ChatServerSessionPassword** key/value pair). Only in basic chat protocol it is possible to specify a user-defined session password by adding **GCTI_Chat_SessionPassword** key/value pair in userdata of **RequestJoin** when creating a chat session.

Supported:

Chat Server	PSDK	Workspace (both) Edition	Chat Widget	GMS
8.5.109	8.5.1x	not supported	joining an existing chat session is not supported	joining an existing chat session is not supported

Chat bot participant special treatment

Only the agent or supervisor in a chat session can be marked as "bot" participants. It happens when the userdata of **RequestJoin** (when participant joins chat session) contains **GCTI_Chat_SetPartyStyle** key/value pair with value "BOT". Chat Server attaches another key/value pair **GCTI_Chat_PartyStyle="BOT"** to the newParty' event in basic protocol chat transcript and **GCTI_SYSTEM/party-into/style="BOT"** in eventAttributes property (both in newParty event in basic protocol and in all events for bot participant in flex protocol).

For "bot" participants:

- Chat Server does not take such participants into account when processing after-action in **RequestReleaseParty** with value **CloseIfNoAgents**.
- Agent Desktop must not take such participants into account when making a decision to stop the processing of chat session and interaction.
- Reporting statistics (see [Chat Server Reporting Statistics](#)) will not count such participants as an agent or supervisor.

Supported:

Chat Server	PSDK	Workspace Desktop Edition	Workspace Web Edition	Chat Widget	GMS
8.5.109	8.5.1x for userdata location, 9.0.000.01 for eventAttributes	8.5.118	not supported	not supported	8.5.201.04

Notifications about detected and masked out PII data

Chat Server can be configured to detect and replace PII data in a chat session (see [Masking Sensitive Data](#)). If such PII data is detected according to the configuration provided, the message event (both in flex and basic chat transcripts) will contain information in the **eventAttributes** property about what parts of the message contains detected PII data, and how this data was masked out. In Chat Server logs it can be seen as (text is formatted for presentation):

```
eventAttributes={ 'GCTI_SYSTEM'={ 'pii-cleanup'={  
  'rule-0001'={  
    'description'='<rule-description>',  
    'id'='<rule id>',  
    'name'='<rule name>',  
    'positions'={  
      '70-81'={ 'replaced'='digits' }  
    }  
  }  
}}}}
```

Supported:

Chat Server	PSDK	Workspace (both) Edition	Chat Widget	GMS
8.5.109	9.0.000.01	not supported	not supported	8.5.201.04

Read confirmation notice

Chat Server provides the possibility for chat session participants to signal about messages being seen/read. For that, a participant must send **RequestNotify** with notice type **SYS_COMMAND** and notice text **read-confirm**. The userdata of the request must contain key-value pair with key **last-event-id**, and the value must contain the transcript event ID (which is being reported as being seen). Chat Server processes read confirmation notices as follows:

- Other chat participants will receive corresponding notification with provided last-event-id in userdata of the notice transcript event.
- The notice event will be saved in UCS transcript only if option transcript-save-notices = all.

Participant's read confirmation notice events get annihilated from transcript:

- When a participant leaves the session.
- When another read confirmation notice is received from the same participant.
- During the session restoration.

Supported:

Chat Server	PSDK	Workspace Desktop Edition	Workspace Web Edition	Chat Widget	GMS
8.5.105	8.5.1x	8.5.122.08	not supported	not supported	8.5.201.04

Nickname change

Chat Server provides the possibility for chat session participants to change their nickname during the session. For that, a participant must send **RequestNotify** with notice type **USER_UPDATE_NICK** and text containing a new nickname. The nickname of a participant can be changed more than once. Upon receiving such request:

- Chat Server updates the nickname for a participant.
- Chat Server adds this notice to the session transcript.
- Only when updated the nickname for the first time, Chat Server records the original nickname value in **GCTI_Original_Nickname** key-value pair of userdata of the initial newParty event for that participant.

Supported:

Chat Server	PSDK	Workspace (both) Edition	Chat Widget	GMS
8.5.0	8.1.1	not supported	not supported	8.5.201.04

Using rich messaging

Chat Server provides chat session participants (either an agent or chat bot) with the ability to use **Rich Messaging** (in other words to send structured messages) in a chat session. To do so, a participant must send `RequestMessage` (`com.genesyslab.platform.webmedia.protocol.basicchat`) with the following mandatory parameters:

- `MessageText` with plain text message. For backward compatibility, structured messages must be always accompanied with so called "fallback" plain text messages. This way, a component which does not know how to process the structured content will continue to use a regular plain text message supplied together with the structured content.
- `eventAttributes` with `KeyValueCollection` with the following content (nested nodes):
 - `"structured-content"` (of type `kvlist`)
 - `"<origin name>"` (of type `kvlist`)
 - `"content"` (of type `string`) - contains structured message as a valid JSON string (which varies by channel).
 - `"type"` (of type `string`) - contains the content type, and is mandatory for all channels (except for Widgets).

`"<origin name>"` can be one of the following supported channels:

Channel	<origin name> value
Apple Business Chat	applebc-session
WhatsApp	genesys-chat
Genesys Chat Widget Rich Messaging	genesys-chat

If supported by a channel, the reply message from a customer can contain a `"related-event-id"` attribute (under the `"general-properties"` node in the `eventAttributes`). It contains the corresponding `EventId` of the original outgoing (to a customer) structured message, which is returned in the `EventSessionInfo` when sending a `RequestMessage` in PSDK.

In **Chat API Version 2**, structured messages are delivered in the `messages` array, as shown below:

```
"messages": [
  {
    "from": {"nickname": "The Bot", "participantId": 3, "type": "Agent"},
    "index": 15,
    "text": "<plain text message (fallback version)>",
    "type": "Message",
    "utcTime": 1561059828000,
    "eventAttributes": {
```

```
"GCTI_SYSTEM":{"party-info":{"style":"BOT"}},
"structured-content":{"genesys-chat":{"type":"Quick replies","content":"<contains
structured message as a valid escaped JSON string>"}}
}
]
```

Supported versions

Component	Version
Chat Server	8.5.109.06
PSDK	9.0.000.01
Workspace Web Edition and Workspace Desktop Edition	Not supported
Chat Widget	9.0.006.02
Genesys Mobile Services	8.5.201.04