



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

How to send ESP requests to Chat Session from Workflow

How to send ESP requests to Chat Session from Workflow

Introduction

Genesys can send messages, notices (types are limited), and other requests to a chat session from a workflow (an URS [Universal Routing Server]/ORS [Orchestration Server] strategy).

For example, when a customer starts a chat session from the web page, the chat session is created in Chat Server and corresponding interaction is submitted in Interaction Server. At some point, the interaction is processed by the workflow, which can send a message like "agent will be with you shortly... " and then the routing starts (to find an agent to serve this chat communication).

Prerequisites

Interaction Server application (in configuration) must be connected to Chat Server application's "ESP" port.

How to Implement

The following steps are necessary in order to send a message or notice from the URS strategy:

1. Verify that the interaction is still online by checking that `UData['IsOnline'] != '0'`. If the interaction is offline, which means that the chat session is closed, there is no sense to send messages into it.
2. Extract from the interaction properties the name of the Chat Server application which is processing/handling the ongoing chat session. This can be achieved by assigning `UData['ChatServerAppName']` to a local variable.
3. Use the External Service block in the Data and Services palette in IR Designer (or the External Service block in the Server Side palette in Composer) to send a request. The following general parameters must be specified:
 - The Application type must be set to ChatServer.
 - The Application name must be set to a value obtained from the user data in step 2.
 - The Service name is set to Chat.
 - The Don't send user data must be unchecked.
4. Set the corresponding Method name to send one of the ESP requests, described below.
5. Provide mandatory parameters.

Tip

Each ESP request listed below requires the interaction ID to process the request. Chat Server receives the value for the interaction ID from the userdata. This userdata is supplied with each ESP request of the KVP with the **InteractionId** key. However, this behavior can be overwritten if an explicit parameter in each ESP request with the same **InteractionId** name is provided.

Example: In order for the workflow to process internal communication interactions used to invite another agent for a conference or a consult through the queue, the skills, or the agent group, the **InteractionId** parameter must be initialized with the value of the parent chat interaction UData['ParentId'].

Available in Chat Server since **8.5.314.02**.

Important

Chat Server processes values for all incoming parameters as a string value, even if it represents a number or a Boolean value. For Boolean parameters, only **true** is recognized as a value for **true**. All other values are recognized as **false**.

Method **Message**

Message – Submits a text message to a chat session. Provide the following parameters:

Parameter	Mandatory	Value Description
MessageText	yes	Message text to submit to a chat session
MessageType	optional	Specifies any arbitrary text as message type (transparent for Chat Server).
Nickname	optional	Specifies the nick name of a participant on behalf of whom the message will be shown in a chat session.
Visibility	optional	<p>Possible values:</p> <ul style="list-style-type: none">• ALL – message is visible to all chat participants (default value)• INT – message is visible to agents and supervisors only• VIP – message is visible to supervisors only <p>Use visibility wisely as not all components (including Genesys Workspace) may show it correctly.</p>

Parameter	Mandatory	Value Description
EventAttributes	optional	Specifies a nested list of attributes which are associated with the message provided. For more information, see Key-value collection format specification below with its Example . Available in Chat Server since 8.5.201.05 .

Event3rdServerResponse

Parameter	Value type	Mandatory	Value Description
OccuredAt	String	Yes	Timestamp for when the method was processed.
ScriptPos	Integer	Yes	Position of the message submitted in the chat transcript. Positioning starts from 0.

Method **Notice**

Notice – Sends a notification of the specified type to a chat session. Provide the following parameters:

Parameter	Mandatory	Value Description
NoticeType	yes	Possible values: <ul style="list-style-type: none"> USER_PUSHED_URL – to implement the "push URL" functionality (NoticeText must contain valid URL). USER_CUSTOM – can be used for any custom purpose (completely transparent for Chat Server).
NoticeText	optional	Any arbitrary text.
Nickname	optional	Specifies the nick name of a participant on behalf of whom the message will be shown in a chat session.
Visibility	optional	Possible values: <ul style="list-style-type: none"> ALL – message is visible to all chat participants (default value) INT – message is visible to agents and supervisors only

Parameter	Mandatory	Value Description
		<ul style="list-style-type: none">VIP – message is visible to supervisors only <p>Use visibility wisely as not all components (including Genesys Workspace) may show it correctly.</p>
EventAttributes	optional	<p>Specifies a nested list of attributes which are associated with the notice provided. For more information, see Key-value collection format specification below.</p> <p>Available in Chat Server since 8.5.201.05.</p>

Event3rdServerResponse

Parameter	Value type	Mandatory	Value Description
OccuredAt	String	Yes	Timestamp for when the method was processed.
ScriptPos	Integer	Yes	Position of the message submitted in the chat transcript. Positioning starts from 0.

Method **PlaceOnHold**

PlaceOnHold – Places the chat session on hold which sets GCTI_Chat_AsyncStatus to -2.

Important

This is only applicable for async chat sessions.

After calling this ESP method, the workflow must place the interaction into the **waiting** queue (in [Chat Business Process Sample](#) this is done through async-chat-return-queue). GCTI_Chat_AsyncStatus must not be set by the workflow itself since the Chat Server does not send a GCTI_Chat_AsyncStatus update upon the arrival of a new message from a customer, if Chat Server was not explicitly notified through this ESP method.

Request3rdServer

No parameters required.

Event3rdServerResponse

Parameter	Value type	Mandatory	Value Description
OccuredAt	String	Yes	Timestamp for when the method was processed.

Method **CloseSession**

CloseSession - Closes an alive session; Chat Server:

- Notifies active participants
- Updates the UCS record with the final transcript and chat closed date
- Updates IsOnline=0 and stat metrics (if configured) in Interaction Server
- Marks the chat session as Closed internally (but does not immediately remove it from memory, allowing a customer time to request the last transcript fetch).

Request3rdServer

Parameter	Value type	Mandatory	Value Description
CloSelfNoAgents	String	Optional	Possible values: <ul style="list-style-type: none">• true - Session closes only when there are no agents.• false (default) - Session closes even if participants are present in the session.
Purge	String	Optional	When this value is true, Chat Server removes the chat session from its memory without executing any closing actions (such as notifying participants, or updating the UCS and Interaction Server). This request is used internally when session-restore-do-purge=true. Available in Chat Server since 8.5.312.10.

Event3rdServerResponse

Parameter	Value type	Mandatory	Value Description
OccuredAt	String	Yes	Timestamp for when the method was processed.
IsClosed	Integer	Yes	Possible Values: <ul style="list-style-type: none">• 1 (Session closed)• 0 (Session alive)

Method **GetSessionInfo**

GetSessionInfo – Returns information regarding the session.

Request3rdServer

No parameters required.

Event3rdServerResponse

Parameter	Value type	Mandatory	Value Description
OccuredAt	String	Yes	Timestamp for when the method was processed.
SessionInfo	Key-value list	Yes	Lists the following info <ul style="list-style-type: none">• CreatedAt – Timestamp for when the session was created.• IsRestored – Specifies whether the session was restored. Possible values : 0 (false) or 1 (true).

Method **IdleControlConfigure**

IdleControlConfigure – allows to change the configuration for inactivity control monitoring for a given chat session. Provide the following parameters (while all parameters are optional, at least one parameter must be provided):

Parameter	Mandatory	Notes
reset-parameters	optional	Resets all inactivity control parameters to values provided in the Chat Server application configuration.

Parameter	Mandatory	Notes
		Valid values: true / false (default).
enabled	optional	
include-notices	optional	
message-alert	optional	
message-alert2	optional	Available starting with Chat Server release 8.5.107.11
message-close	optional	
timeout-alert	optional	
timeout-alert2	optional	Available starting with Chat Server release 8.5.107.11
timeout-close	optional	

Event3rdServerResponse

Parameter	Value type	Mandatory	Value Description
OccuredAt	String	Yes	Timestamp for when the method was processed.

Method **ConfigureSession**

ConfigureSession - allows you to change the language for the current chat session. At least one of the following parameters must be included:

Parameter	Mandatory	Value Description
async-idle-reset	optional	If the value 1 is provided, Chat Server resets async idle timeouts (in other words, starts counting from zero).
GCTI_LanguageCode	optional	If this parameter is present, other parameters are ignored . The parameter must contain the name of the language business attribute.
GCTI_LanguageName	optional	This parameter is used only if the GCTI_LanguageCode parameter is not present. Parameters are processed as described in the How Chat Server Associates Sessions with Languages section.

Event3rdServerResponse

Parameter	Value type	Mandatory	Value Description
OccuredAt	String	Yes	Timestamp for when the method was processed.

Key-value collection format specification

In order to provide data of type key-value collection in requests from URS workflow, the following parameters must be included in Interaction Routing Designer:

- The parameter Name should be prefixed with {} (lowercase L in curly brackets). This triggers URS to process the parameter value as a nesting list of key-value pairs.
- The parameter value can be specified in either of the following formats (Note: no line breaks allowed and whitespaces are ignored):
 - JSON-like style (can only be provided through the variable)

```
{ "key1": "value1", "key2": "value2", "key3": { "subkey1": "subvalue1" } }
```

- List-like style (can be provided directly in the parameter)

```
key1:value1|key2:value2|key3.subkey1:subvalue1
```

Example

In order to send a Rich Text message with Quick Replies, the following string must be provided as a value for the EventAttributes parameter in the Method Message:

```
{ "structured-content": { "genesys-chat": { "content": { "type": "Message", "contentType": "quick-replies", "content": [ { "id": 1, "type": "quick-reply", "action": "message", "text": "Black" }, { "id": 2, "type": "quick-reply", "action": "message", "text": "Green" }, { "id": 3, "type": "quick-reply", "action": "message", "text": "Mint" } ] }, "type": "Generic" } } }
```

It is important that the actual Rich Text element body JSON (specified in content) is escaped so it will be treated as a value instead of being converted into a nested Key/Value list structure.