A MITEL
PRODUCT
GUIDE

MiContact Center Enterprise

Chat Configurations – Operating Instructions

Release 9.8
Document Version 1.0

August 2025

Mitel
Powering connections

Notices

The information contained in this document is believed to be accurate in all respects but is not warranted by **Mitel Networks™ Corporation (MITEL®).**

The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes. No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

Trademarks

The trademarks, service marks, logos and graphics (collectively "Trademarks") appearing on Mitel's Internet sites or in its publications are registered and unregistered trademarks of Mitel Networks Corporation (MNC) or its subsidiaries (collectively "Mitel") or others. Use of the Trademarks is prohibited without the express consent from Mitel. Please contact our legal department at legal@mitel.com for additional information. For a list of the worldwide Mitel Networks Corporation registered trademarks, please refer to the website: http://www.mitel.com/trademarks.

INTRODUCTION

MiCC Enterprise may be integrated with chat functionality allowing customers to request chat sessions which will be routed to agents. This document will describe integrating the built-in customer chat client into your external Web site. Knowledge of basic Internet Web design and HTML is assumed.

CUSTOMER CHAT CLIENT

Chat functionality may be integrated into your Web site using the stock chat client or you may design your own.

STOCK CHAT CLIENT

During installation of MiCC Enterprise a CustomerChat Web application is added to the default Web site of Internet Information Server. The CustomerChat Web application was built using ASP.NET v4.0 and uses SignalR technology for communication between the client and server.

INTEGRATION

The chat toolbox can be integrated into a Web site using a single line of Javascript code. Add the following line to the end of the <head> section of the HTML page.

<script src="http[s]://<MICCEWEBSERVER[:port]>/CustomerChat/Chat/Toolbox" />

For example:

```
<head>
   <script src="https://MICCESERVER:443/CustomerChat/Chat/Toolbox" />
</head>
```

The toolbox will automatically be initialized when the page loads and will display as a popout tab

The CustomerChat Web application contains a sample Web page that integrates the chat toolbox. The page can be accessed using the following URL:

http://<MICCEWEBSERVER/CustomerChat/Chat

You may view the source of the Web page for sample code integrating the chat toolbox.

CONFIGURATION

WARNING: In the following sections that describe modifying configuration files, a backup of the existing files should be done before any changes are made. In addition, after modifying existing files or adding new ones, a backup should be made of the new or changed files. During upgrades of the software, these changes may be lost.

Web.config

Web.config contains settings that used by the back-end CustomerChat Web application as well as client side script. These are contained in the <appSettings> section of the config file. The file is located at:

<InstallDir>\Services\Web\CustomerChat\web.config

SETTING	PURPOSE
CustomerDisconnectTimeoutSeconds	When a customer leaves the current Web page containing the chat session, a disconnect occurs between the page and the back end Web application. If the customer returns to the page containing the chat session within this time, they will be reconnected to the current chat session. Otherwise, the chat session will be terminated. Default = 60.
AgentWarningCustomerDisconnectTimeoutSeconds	When a customer has left the current Web page containing the chat session and a disconnect occurs, the agent will receive a message X seconds before the session timeout will occur according to CustomerDisconnectTimeoutSeconds. For example, if CustomerDisconnectTimeoutSeconds is 60 and AgentWarningCustomerDisconnectTimeoutSeconds is 20, the agent will receive the message 40 seconds after the disconnect occurred. Default = 20.
IgnoreCustomerDetails	true or false. If true, details such as customer name, phone number and e-mail address are ignored when requesting a chat. Default = false.
ServiceGroupIDsToReject	A comma separated list if service group IDs that cannot request chats. Default = "".
CrossDomainAllowIpAddress	If the Chat toolbox is being invoked from a different domain than that of the toolbox, the IP address or domain name of the calling domain must be specified.

Mitel.ChatToolbox.Config

The Mitel.ChatToolbox.Config object contains the main settings that apply to UI state of the Chat toolbox and chat conversations. The object is defined in the file:

<InstallDir>\Services\Web\CustomerChat\Scripts\Chat\Config.js

The file may be modified directly which will apply as the default for all chat sessions.

Configuration may be set on a page basis by redefining the customizePage function. This function executes immediately after the toolbox is loaded. To redefine the function, insert the following Javascript code into your Web page:

```
<script>
        Mitel.ChatToolbox.customizePage = function() {
}
</script>
```

Inside the function modify any of the Mitel.ChatToolbox.Config properties.

Configuration may also be set on a request basis by redefining the customizeRequest function. This function executes immediately before the chat is requested. To redefine the function, insert the following Javascript code into your Web page:

```
<script>
        Mitel.ChatToolbox.customizeRequest = function() {
}
</script>
```

Inside the function modify any of the Mitel.ChatToolbox.Config properties. Some properties do not apply to each individual chat request and should not be modified in the customizeRequest function. For example, ToolboxAvailable which determines whether the toolbox is available does not apply to each chat session. Also, some properties such as ServiceGroupID must be known before a chat is requested as they are used to determine availability.

Properties

PROPERTY	PURPOSE
ToolboxAvailable	true or false. Indicates if the chat toolbox is available at all. Default = true. This property must not be modified in the customizeRequest function.
ShowWhenUnavailable	true or false. Indicates if the toolbox should be shown even when the defined service group is not available for chat requests. Default = true. This property must not be modified in the customizeRequest function.
StatePollInterval	Amount of time in seconds between polling for service group state and estimated wait time. Default = 20. This property must not be modified in the customizeRequest function.
ShowEstimatedWaitTime	true or false. Indicates if the estimated wait time should be displayed. Default = true.
SessionID	Identifier of the requested chat session. This can be left blank if the session does not need to be identified. Default = "".
CustomerName	Customer name used to initiate the chat session. This will be populated with the entry on the Request Chat page. This can be left blank indicating an anonymous customer. Default = "Customer".
CustomerID	Identifier of the customer used to initiate the chat session. This value is used strictly for archiving purposes and may be left blank. Default = "".
CustomerEmail	Customer e-mail address used to initiate the chat session. This will be populated with the entry on the Request Chat page. Default = "".
CustomerPhone	Customer phone number used to initiate the chat session. This will be populated with the entry on the Request Chat page. Default = "".
InitialMessage	Initial question the customer would like to ask. This will be populated with the entry on the Request Chat page. Default = "".
TenantID	Tenant ID. Required. Default = -1.
ServiceGroupID	ID of the service group used to request the chat session. Either this property or ServiceGroupName must be defined.

ServiceGroupName	Name of the service group used to request the chat session. Either this property or ServiceGroupID must be defined.
PrivateData	Any private data to be attached to the chat session. Default = "".
HideAgentName	true or false. If true, no agent name will be displayed to the customer. Default = false.
ShowPostChat	true or false. Indicates if the post chat page is to be shown where a new chat may be requested. Default = true.
TimeoutAfterConversation	Timeout in seconds after a chat conversation has ended before automatically showing the post chat page. If this value is 0, the post chat page will not be shown automatically. This only applies if ShowPostChat = true. Default = 0.
ToolboxPosition	Position of the toolbox as it is embedded in the Web page. Allowed values are: top-left top-right bottom-left bottom-right center-left center-right Default = center-right.
IdleTimeout	Timeout in seconds before an idle customer is disconnected. Default = 300.
WarningBeforeIdleTimeout	Time in seconds before an idle time is to occur before the customer receives a warning message. For example, if IdleTimeout is 300 and WarningBeforeIdleTimeout is 60, the customer will receive a warning 240 seconds after being becoming idle. 0 = no warning. Default = 60.
AllowMovingChatCard	true or false. If true, the chat card may be moved within the Web page. Default = true.
InitialMessageToAgent	If configured, this message will be sent to the agent when the chat is requested. The message will not be displayed to the customer.
PostedMessageAllowedOrigin	Messages can be posted to the chat client from web pages through the window.postMessage function. This value indicates the allowed Web origin that may post messages. Normally messages may only be posted by the Web page containing the chat toolbox. Default = "window.location.origin" indicating the same origin as the chat toolbox.
MaxMessageLength	Maximum length of the message that the customer may send. The maximum length will be the lesser of this value and the MaxMessageLength configured for the Chat service. Agents may still send messages up to the MaxMessageLength configured for the Chat service. This property must not be modified in the customizeRequest function.
MaxAttachmentSize	Maximum attachment size specified in KB. The maximum size will be the lesser of this value and the MaxAttachmentSize configured for the Chat service. If this value is 0, attachments are disabled. Agents may still send attachments to the customer if configured in the Chat service.
AllowedAttachmentTypes	List of file type extensions that are allowed to be uploaded. An empty list indicates that all types are allowed. Types must also be allowed by the Chat

	service. The list should be specified as a JSON string array. For example: ["jpg", "png", txt"].
RestrictedAttachmentTypes	List of file type extensions that are not allowed to be uploaded. An additional restriction may be imposed by the Chat service. This list is only checked if AllowedAttachmentTypes is not defined. The list should be specified as a JSON string array. For example: ["jpg", "png", txt"].

TEXT CHANGES AND LOCALIZATION

All displayed text strings are contained in the file:

<InstallDir>\Services\Web\CustomerChat\App_GlobalResources\Toolbox.resx

The text strings in this file may be modified to better suit the customer. The text strings may also be localized into additional languages. To add an additional language, copy the Toolbox.res file to a new file in the same folder using the following format:

Toolbox.<language>.resx

<language> should be the ISO 639 two-letter culture code and may contain an ISO 3166 two-letter uppercase subculture. For example, to create a general language for French, create the file:

Toolbox.fr.resx

To create a language for French with a subculture for Belgium, create the file:

Toolbox.fr-BE.resx

Toolbox.resx is an XML based language resource file. Be careful not to change the structure of the file. The following text strings may be modified:

NAME	PURPOSE
AnonymousAgentJoined	Message displayed when an anonymous agent has joined the conversation.
AnonymousAgentLeft	Message displayed when an anonymous agent has left the conversation.
AnonymousAgentTyping	Text displayed when an anonymous agent is typing.
AskAQuestion	Initial question prompt.
Browse	Browse button text.
ChatRequested	Default message displayed when a chat is requested if an initial queue message is not defined.
ChatUnavailable	Text displayed when chat is unavailable.
CustomerBack	Message displayed to the agent when the customer has navigated back to the chat conversation after viewing a different Web page.

CustomerBackReceivedMessage	Message displayed to the agent when the customer has navigated back to the chat conversation after viewing a different Web page and there are pending messages for the customer.
CustomerLeft	Message displayed to the agent when the customer has temporarily left the chat page.
CustomerLeftAutoSend	Message displayed to the agent when the agent sends a message to the customer and the customer has temporarily left the chat page.
CustomerLeftClosing	Message displayed to the agent when the customer has been gone from the chat page over the defined limit and the conversation will be terminated.
EnterEmail	Enter e-mail address prompt.
EnterName	Enter name prompt.
EnterPhone	Enter phone number prompt.
ErrorOccurred	Message displayed when a general error has occurred within the chat conversation.
ErrorRequestingChat	Message displayed when an error has occurred while requesting a chat session.
ErrorSendingMessage	Message displayed when an error has occurred while sending a chat message.
EstimatedWaitTime	Estimated wait time label.
InvalidAttachmentType	Message displayed if an attempt is made to upload an attachment type that is not allowed.
MaxAttachmentSizeExceeded	Message displayed when trying to upload an attachment that is larger than the maximum configured size.
Minute	Minute label when minute equals 1.
Minutes	Minute label when minute is not equal to 1.
NamedAgentJoined	Message displayed when a named agent has joined the conversation.
NamedAgentLeft	Message displayed when a named agent has left the conversation.
NamedAgentTyping	Text displayed when a named agent is typing.
No	No.
PostText	Text displayed in the post chat page.
PostUrl	Clickable URL displayed in the post chat page.
PostUrlText	Text of the clickable URL displayed in the post chat page.
Product	Product name.

ReallyClose	Message displayed to confirm closing the chat session.
RequestRejected	Message displayed when the chat request has been rejected.
Restart	Restart button text.
SelectFile	Select file prompt.
Send	Send button text.
SendTheMessage	Tooltip for the Send button.
SessionTimeout	Message displayed when the chat session has timed out.
SessionWillSoonBeClosed	Message displayed when the chat session will timeout soon.
StartChat	Start chat button text.
Terminated	Message displayed when the chat session has terminated.
Title	Title text of the chat toolbox.
TypeMessageHere	Prompt for the message text entry field.
Upload	Upload button text.
UploadComplete	Text displayed when uploading an attachment is complete.
UploadFailed	Text displayed when uploading an attachment has failed.
UploadProgress	Text displayed indicating upload progress.
UploadTheFile	Tooltip for the Upload button.
WelcomeText	Welcome text displayed in the pre-chat page.
Yes	Yes.

Note: Any placeholders such as {0} or {1} in the text strings must be maintained. The location in the text string may be moved, but it must exist.

POSTING MESSAGES TO THE CHAT CLENT

Messages may be posted to the chat client as if they were entered by the customer. Messages may only be posted when there is an active chat session and may only be posted by the Web origin indicated by the Mitel.ChatToolbox.Config.PostedMessagesAllowedOrigin setting. Following is an example of posting a message:

```
window.postMessage({
        apiType: "ChatToolbox",
        apiVersion: "2.2.0",
        methodName: "SendMessage",
        arguments: {
                message: "Message to Send",
```

```
            hideLocal: true
        }
}, window.location.origin);
```

hideLocal may be true or false. If true, the message will only be sent to the agent and will not be seen by the customer.

BROWSER SUPPORT

- Internet Explorer v9.0 or Later

- Current Version of Chrome

- Current Version of Firefox

Additional versions may be supported, however, only the current released version of Chrome and Firefox at the time of MiCC Enterprise release have been tested.

RUNNING STANDALONE ON A PUBLIC FACING WEB SERVER (DMZ)

The CustomeChat Web application may be used on public facing Web Server (DMZ). The following steps should be performed.

1. Install MiCC Enterprise on the DMZ.

    a. Select Custom installation.

    b. On the Select Features page, select only the Services/Web Services feature.

    c. Set the Broker Location to the main MiCC Enterprise server.

    d. Continue and complete the installation.

2. Remove unnecessary Web applications. The only necessary Web application is CustomerChat. The additional Web applications may be removed, however, they will never be used. If you wish to remove the Web applications, open IIS Manager and remove the following applications under the Default Web Site:

    MiCCEInstallation
    RTI
    scheduler
    seceventws
    seclogonws
    secreportws
    secsapicidriver
    SolidusACD
    WebAgent
    WebApps
    WebCallback

3. Run the MiCC Enterprise Setup Utility.

A. On the Web Server Location page, set the location to where the main MiCC Enterprise Web Services are installed. This is typically the same as the main MiCC Enterprise server. Set the appropriate options for connecting to the main MiCC Enterprise server and Web Server.

4. Open necessary ports. The CustomerChat Web application connects to the Broker Service and Chat Service on the main MiCC Enterprise server. The configured ports for these services must be open and accessible by the DMZ.

5. Ensure machine name resolution. The CustomerChat Web application retrieves the location of the Chat Service from the Broker Service. The location received is depending on how the Chat Service registers itself with the Broker. Typically, this is the simple machine name, but it may also be a full qualified domain name or an IP address. The DMZ must be able to access the Chat Service using that received location.

TEMPLATE MODIFICATION

The display of the Chat Client is based on HTML templates and in some cases, it may be possible to do slight modifications as necessary. In previous versions of MiCC Enterprise, these templates were contained in the file:

<InstallDir>\Services\Web\CustomerChat\Content\Templates.html

Slight modifications could be made to the templates with a direct result. For security reasons, as of MiCC Enterprise 9.7 SP1, direct modification is no longer possible and Templates.html no longer exists. Templates are still in use; however, they are now precompiled into the application. It may still be possible to modify the templates; however, it will require assistance from Mitel to compile these into a usable form. The templates are *.vue files supplied in the folder:

<InstallDir>\Services\Web\CustomerChat\Content\Templates

These files are not directly used by the application. You may make small changes to the templates and then you must contact Mitel Support and supply the modified templates to be compiled into a usable form by the application. Note that the changes that are possible is at the discretion of Mitel.

CUSTOM CHAT CLIENT

If the stock MiCC Enterprise Chat client does not suit your needs, you may create your own client. A REST service has been exposed from the MiCC Enterprise Chat service running on the MiCC Enterprise server. You may communicate with the service directly from client side script; however, this would require opening access to the MiCC Enterprise Chat service which is typically behind a firewall. A more typical scenario would be to communicate with the MiCC Enterprise Chat service from a back-end Web application such as from the code-behind in an ASP.NET application.

All requests to the Chat service use JSON data format and are made through the following URL:

http://<MiCCEServer[:port]>/chatservice/<Request>

For example:

The default port for the Chat service is 12616, but this can be changed on the MiCC Enterprise server through the MiCC Enterprise Setup Utility.

The following example shows calling the RequestChat method from Javascript using JQuery:

```
function RequestChatFromService()
{
        var requestParam = new Object();

        requestParam.SessionID = '{50B5D2FF-57BD-4948-A864-B044A6E9FEB7}';
        requestParam.CustomerID = '100';
        requestParam.CustomerName = 'John';
        requestParam.EmailAddress = 'john@somecompany.com';
        requestParam.PhoneNumber = '555-5555';
        requestParam.TenantID = -1;
        requestParam.ServiceGroupID = 0;
        requestParam.ServiceGroupName = 'Sales';
        requestParam.PrivateData = 'SomePrivateData';

        var DTO = {'request' : requestParam};

        $.ajax({
                type: "POST",
                contentType: "application/json; charset=utf-8",
                url: "http://MICC ENTERPRISESERVER:12616/ChatService/RequestChat",
                data: JSON.stringify(DTO),
                datatype: "json",
                success:
                function(data)
                {
                        return data.ChatID;
                }
        });
}
```

IDLE TIMEOUTS

An idle timeout scenario should be handled properly by the custom chat client to terminate the session. There is still a server side timeout as a fail-safe procedure. See the *Chat Service Configuration* section for the server side idle timeout setting.

CHAT SERVICE API

Methods

All methods will throw fault exceptions on failure except where noted.

RequestChatResponse RequestChat(RequestChatRequest request)

Requests a chat session. If the request fails due to the service group being closed, a fault will not be thrown and the group closed message will be returned if configured.

void LeaveChat(LeaveChatRequest request)

Leaves a chat session

void SendMessage(SendChatMessageRequest request)

Sends a message from the customer to the agent.

void SendTyping(SendTypingRequest request)

Sends a notification to the agent that the customer is typing.

GetEventsResponse GetEvents(GetEventsRequest request)

Gets pending events such as messages received from the agent.

GetQueueInfoResponse GetQueueInfo(GetQueueInfoRequest request)

Gets state information for a service group such as availability and estimated wait time.

GetChatInfoResponse GetChatInfo(GetChatInfoRequest request)

Gets state information for a chat session.

GetChatConfigurationResponse GetChatConfiguration()

Gets global configuration information related to chat sessions.

string GetVersion()

Returns current the MiCC Enterprise version.

Types

RequestChatRequest

NAME	TYPE	PURPOSE
SessionID	string	Unique ID of session. This may be used to tag the chat session during archive operations.
CustomerID	string	ID of customer. This may be used to tag the chat session during archive operations.
CustomerName	string	Name of customer.

EmailAddress	string	E-mail address of the customer.
PhoneNumber	string	Phone number of the customer.
TenantID	int	Tenant ID of the MiCC Enterprise service group where the chat session will be routed. This will be -1 for non-tenanted MiCC Enterprise systems.
ServiceGroupID	int	ID of the service group where the chat session will be routed. Either ServiceGroupID or ServiceGroupName must be specified.
ServiceGroupName	string	Name of the service group where the chat session will be routed. Either ServiceGroupID or ServiceGroupName must be specified.
PrivateData	string	Any character string that you wish to pass along to the agent. This value will not be displayed in the Agent's display, but it is provided to the Agent Integration interface.
TimeZoneOffset	int	Offset in minutes between the UTC and the local time zone.

RequestChatResponse

NAME	TYPE	PURPOSE
ChatID	string	Contains the unique ID associated to the new chat session. If the request fails due to the service group being closed, this value will be null and the group closed message will be returned in Message if configured.
Message	string	Initial queue or group closed message if any.
SessionID	int	Internal session ID.
QueuePosition	int	Position in the queue.
EstimatedWaitTime	int	Estimated wait time in seconds or -1 if an estimate could not be determined.

LeaveChatRequest

NAME	TYPE	PURPOSE
ChatID	string	Unique ID returned from the RequestChat method.

SendChatMessageRequest

NAME	TYPE	PURPOSE
ChatID	string	Unique ID returned from the RequestChat method.
Message	string	Message to send. If Attachment is specified, Message

NAME	TYPE	PURPOSE
		should contain the base filename of the attachment.
Attachment	byte[]	File data of attachment.

SendTypingRequest

NAME	TYPE	PURPOSE
ChatID	string	Unique ID returned from the RequestChat method.

GetEventsRequest

NAME	TYPE	PURPOSE
ChatID	string	Unique ID returned from the RequestChat method.
Timeout	int	Timeout in milliseconds to wait for events.

GetEventsResponse

NAME	TYPE	PURPOSE
Events	List<ChatEvent>	List of received events

GetChatInfoRequest

NAME	TYPE	PURPOSE
ChatID	string	Unique ID returned from the RequestChat method.

GetChatInfoResponse

NAME	TYPE	PURPOSE
State	ChatState	Current state.
QueuePosition	int	Position in the queue. Only applicable if State = Queued.

GetQueueInfoRequest

NAME	TYPE	PURPOSE
TenantID	int	Tenant ID of the MiCC Enterprise service. This will be -1 for non-tenanted MiCC Enterprise systems.
ServiceGroupID	int	ID of the service group. Either ServiceGroupID or

		ServiceGroupName must be specified.
ServiceGroupName	string	Name of the service group. Either ServiceGroupID or ServiceGroupName must be specified.

GetQueueInfoResponse

NAME	TYPE	PURPOSE
EstimatedWaitTime	int	Estimated wait time in seconds or -1 if an estimate could not be determined.
Open	bool	True if the service group is available.

GetChatConfigurationResponse

NAME	TYPE	PURPOSE
MaxMessageLength	int	Maximum length of message text that can be sent.
MaxAttachmentSize	Int	Maximum size in bytes of attachments that cane be sent. If 0, attachments cannot be sent.
AllowedAttachmentTypes	String[]	List of attachment types that are allowed to be uploaded. If null or blank, all types are allowed.
RestrictedAttachmentTypes	String[]	List of attachment types that are not allowed to be uploaded.

ChatEvent

NAME	TYPE	PURPOSE
EventType	ChatEventType	Type of the event. The type indicates what type of object will be contained in the Data member.
Data	object	Event data. See ChatEventType.

ChatEventType (enum int)

NAME	VALUE	DATA OBJECT TYPE	PURPOSE
MessageReceived	0	MessageReceivedEvent	Message was received from agent.
AgentJoined	1	AgentJoinedEvent	Agent has joined the conversation.
AgentLeft	2	AgentLeftEvent	Agent has left the conversation.
AgentTyping	3	AgentTypingEvent	Agent is typing.

ChatState	4	ChatStateEvent	Chat state has changed.

MessageReceivedEvent

NAME	TYPE	PURPOSE
ChatID	string	Unique ID of the chat session.
Name	string	Sender of the message.
Message	string	Message received. If Attachment is specified, Message will contain the base filename of the attachment.
System	bool	True if the message was generated from the system.
TimeStamp	DateTime	Time the message was received.
Attachment	byte[]	File data if this is an attachment.

AgentJoinedEvent

NAME	TYPE	PURPOSE
ChatID	string	Unique ID of the chat session.
Name	string	Name of the agent that joined the conversation.

AgentLeftEvent

NAME	TYPE	PURPOSE
ChatID	string	Unique ID of the chat session.
Name	string	Name of the agent that left the conversation.

AgentTypingEvent

NAME	TYPE	PURPOSE
ChatID	string	Unique ID of the chat session.
Name	string	Name of the agent that is typing.

ChatStateEvent

NAME	TYPE	PURPOSE
ChatID	string	Unique ID of the chat session.

State	ChatState	Current state.
QueuePosition	Int	Position in the queue. Only applicable if State = Queued.
EstimatedWaitTime	int	Estimated wait time in seconds or -1 if an estimate could not be determined. Only applicable if State = Queued.
TerminationCause	TerminationCause	If State = Terminated, specifies the cause of the termination.

ChatState (enum int)

NAME	VALUE	PURPOSE
Queued	0	Chat has been queued to a service group and is waiting for an agent to be allocated.
Handling	1	Chat is being handled by an agent.
Terminated	2	Chat session has been terminated.

TerminationCause (enum int)

NAME	VALUE	PURPOSE
General	0	Chat has been generally terminated.
GroupClosed	1	Chat has been terminated due to the call center closing.

CHAT SERVICE CONFIGURATION

Values in the registry for the CCChat service control global settings applicable to the communication interface between agents, the CCChat service and the customer chat client. These settings apply whether the stock chat client or a custom client is being used. All registry entries are stored under the following key.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\CCChat\Parameters

VALUE NAME	TYPE	PURPOSE
CustomerTimeout	REG_DWORD	Maximum amount of time in seconds that a customer may remain idle before being disconnected. The actual timeout will be the lesser of this value and the timeout configured in the customer chat client. Default = 900 seconds.
MaxMessageLength	REG_DWORD	Maximum length of a message that can be sent. Default = 1024 characters.
MaxAttachmentSize	REG_DWORD	Maximum size of attachments that can be sent specified in KB. Setting this value to 0 will disable attachment sending. Default = 5120 (5MB).

AllowedAttachmentTypes	REG_STRING	Comma separated list of file type extensions that are allowed to be uploaded. Blank indicates all types are allowed. Default = blank.
RestrictedAttachmentTypes	REG_STRING	Comma separated list of file type extensions that are not allowed to be uploaded. This list is only checked if AllowedAttachmentTypes is not defined. Default = exe, bat, cmd, com, vbs.
FromPriority	REG_STRING	Determines which field is displayed for the caller number in CDR and the From field in the Chat Viewer Form when viewing archived chats from the CDR or Customer Search. The customer ID, e-mail address or phone number may be displayed. This setting determines the order in which those data elements are checked for existence. Specify the fields in a comma delimited string. Possible field are: phone – Phone Number email – E-mail Address id – Customer ID For example, if the following is specified: phone, email, id First, the phone number field is checked for existence. If it doesn't exist, the e-mail address field is checked, and lastly, the customer ID field. The default order if this setting is not specified is email, phone, id

E-MAIL TRANSCRIPT

At the end of a chat session, the customer may be e-mailed a transcript of the conversation. This can be invoked manually from the Agent Chat form or opened automatically when the form is closed. Transcript properties are set on the Chat tab of the Service Group Properties in Configuration Manager.

The transcript is only available to the agent if the Sender E-mail Address is specified for the Service Group. The agent must also have the privilege to send e-mail plus an email license allocated.

If Open Transcript E-mail on Close is selected, the e-mail form with the transcript will be automatically opened when the agent closes the chat session form.

A custom subject and template may be used for the transcript subject and e-mail body. If these values are empty, a default subject and body will be used.

The transcript template may be any standard text base file (*.txt) or html format files (*.htm, *.html). The filename must be entered in UNC format.

Both the subject and template file may contain placeholders which will be replaced by data associated to the chat as well as the transcript itself. All identifiers are valid for the template file. All identifiers except $Transcript$ are available for the subject.

It is important to ensure that replaceable identifiers are entered in a continuous string in the template file. HTML editors such as Microsoft Word may split the text while inserting HTML format tags. This will prevent the identifiers from being replaced. This may occur if text is identified as a misspelled word. The underlining used in Microsoft Word to indicate the misspelled word will be stored in the HTML file as formatting information. Always ensure that replaceable identifiers are ignored for spell checking.

Replaceable Identifiers

IDENTIFIER	REPLACEMENT
$Transcript$	Chat transcript
$Date$	Current date formatted using the short date format of the current locale
$Time$	Current time formatted using the short time format of the current locale
$Received$	Date and time the e-mail was received formatted using the short date and short time formats of the current locale
$Received.Date$	Date the e-mail was received formatted using the short date format of the current locale
$Received.Time$	Time the e-mail was received formatted using the short time format of the current locale
$ServiceGroup$	Service group name
$ServiceGroup.Name$	Service group name
$ServiceGroup.Email$	Service group e-mail address.
$Agent$	Agent name
$Agent.Name$	Agent name
$Agent.FirstName$	Agent first name, if defined
$Agent.LastName$	Agent last name, if defined
$Agent.ChatName$	Agent Chat Display Name, if defined. Otherwise, set to agent name.
$Customer$	Customer name
$Customer.Name$	Customer name
$Customer.Email$	Customer e-mail address
$Customer.Number$	Customer phone number

CHAT RESPONSES

A response file may be setup for each service group allowing the agents to select predefined messages to be inserted into chat messages. If a response file is configured, a hierarchal list of the responses is displayed in the agent chat form. Refer to the section on E-mail, Chat and SMS Response Files in the document **3_1543-LXA119154 – Advanced Configurations** for the response file format.