Managing Configuration Data

This chapter includes the following topics:

Why Configuration Data Is Shared

Clients can acquire configuration data from the CTI server and use configuration messages to synchronize that data between Unified CCX and themselves. This includes an initial bulk upload of configuration data and subsequent dynamic updates whenever there is a configuration change made through the RMCM subsystem in the Unified CCX Administration software on Unified CCX.

Configuration Data That Can Be Shared

Currently, the Unified CCX CTI protocol supports the CTI client retrieval of only the following configuration data.

CSQ (Contact Service Queue)

A CSQ controls incoming calls by determining where an incoming call is to be placed in the queue and to which agent the call is sent.

Unified CCX Application

Unified CCX applications interact with contacts and perform such functions as prompting callers for information, transferring calls, providing information to callers, and so on.

Agent

An agent is the person in the contact center who handles customer calls, e-mail, and other types of correspondence between the customer and the company.

Device (agent extensions, application triggers and CTI ports)

Call connection points in the computer system.

Team

Agent team information. A team may include agents, CSQs, a primary supervisor, and a secondary supervisor.

Configuration Message Summary Descriptions

Table 1: Unified CCX CTI Configuration Message List, on page 2 lists the configuration messages and Table 2: Configuration Message Descriptions, on page 3 describes the messages. The rest of this chapter explains how to use them. See Message Type Definitions for detailed descriptions of the message contents.

Message List

Table 1: Unified CCX CTI Configuration Message List

Request	Response
CONFIG_REQUEST_KEY_EVENT	CONFIG_KEY_EVENT
CONFIG_REQUEST_EVENT	CONFIG_BEGIN_EVENT CONFIG_END_EVENT CONFIG_CSQ_EVENT CONFIG_APPLICATION_EVENT CONFIG_AGENT_EVENT CONFIG_DEVICE_EVENT
TEAM_CONFIG_REQ	TEAM_CONFIG_EVENT TEAM_CONFIG_CONF

Message Summary Descriptions

Table 2: Configuration Message Descriptions, on page 3 briefly describes the configuration messages. For complete definitions of these messages, see the referenced definition link for each message.

Table 2: Configuration Message Descriptions

Message	Sent by	Description
Configuration initialization messages		
CONFIG_KEY_EVENT	Client	Requests configuration data from the server. The type of update data received depends on the ConfigInfoMask used in this request. The ConfigInfoMasks used to indicate the type of data requested are: 1 = Application Information 2 = CSQ Information 4 = Agent Information 8 = Device Information 0 = No initial configuration information upload is wanted, but update configuration data is wanted. This is used when the client does not want the initial update messages but does want, at the time of this message, configuration update messages.
CONFIG_KEY_EVENT	Unified CCX	Response to the client CONFIG_REQUEST_KEY_EVENT message. Contains the requested Unified CCX configuration keys at the time of the CONFIG_REQUEST_KEY_EVENT.
CONFIG_REQUEST_EVENT	Client	Response to the server CONFIG_KEY_EVENT message. Tells the server if it wants the requested initial snapshot of the configuration or only updates.
Configuration data messages		
CONFIG_BEGIN_EVENT	Unified CCX	Indicates the beginning of configuration data (all data of the same type of the same key) from the server.
CONFIG_END_EVENT	Unified CCX	Indicates one of the following: • The end of a successful configuration upload. In this case, it follows configuration records preceded by a CONFIG_BEGIN_EVENT message. • No configuration for the items requested. • An error condition. This and the preceding message may be in response to a CONFIG_REQUEST_EVENT message.

Message	Sent by	Description
CONFIG_CSQ_EVENT	Unified CCX	Sends CSQ configuration data to a client.
CONFIG_APPLICATION_EVENT	Unified CCX	Sends application configuration data to a client.
CONFIG_AGENT_EVENT	Unified CCX	Sends agent configuration data to a client.
CONFIG_DEVICE_EVENT	Unified CCX	Sends device configuration data to a client.
Configuration team messages		
TEAM_CONFIG_REQ	Client	Requests team configuration from the server.
TEAM_CONFIG_EVENT	Unified CCX	Sends team configuration data to a client.
TEAM_CONFIG_CONF	Unified CCX	Indicates the end of team configuration data or an error condition.

Share Configuration Data

Note The following process applies to application, CSQ, agent, and device configuration. For agent team data, see Share Agent Team Configuration Data, on page 7.

For a client to get configuration data:

Procedure

Step 1 The client must include the CTI_SERVICE_CONFIG_EVENTS bit (0x00040000) in the ServiceRequested field of an OPEN_REQ message if the client wants configuration data. The Unified CCX CTI server does not provide configuration data to the client unless that bit is set. In the OPEN_REQ message, the client must also include the configuration message mask indicating the type of configuration data that it wants.
The client cannot send a configuration request before the server is online. The status of the server is shown in the OPEN_CONF reply to the OPEN_REQ message or in a SYSTEM_EVENT message.

Step 2 The client sends a CONFIG_REQUEST_KEY_EVENT message to the server to request a set of current configuration keys (see CONFIG_REQUEST_KEY_EVENT).

Step 3 In response to the CONFIG_REQUEST_KEY_EVENT message, the server replies with a CONFIG_KEY_EVENT message containing the requested keys (see CONFIG_KEY_EVENT).

Step 4 Based on the keys returned in the CONFIG_KEY_EVENT message, the client decides if it needs an initial snapshot of the configuration.

Step 5 If the client needs an initial snapshot of the configuration, it sends the CONFIG_REQUEST_EVENT message with the requested configuration mask. Otherwise, the client sends this message with the configuration mask set to zero, indicating that the client configuration is in sync with the server configuration and that it only needs configuration updates.

Step 6 If in Step 5, the initial snapshot of the configuration is requested, the server sends the initial snapshot of the configuration data to the client in a CONFIG_BEGIN_EVENT and CONFIG_END_EVENT block of messages. In between these two messages, the server sends the appropriate message for the type of data requested (see CONFIG_BEGIN_EVENT and CONFIG_END_EVENT).
If all the data does not fit in one message, the server sends multiple messages of the same type. In this case, each data message is called a record. The whole transaction from the begin event to the end event is called a data block.

The server sends configuration data messages in the following order:

CSQ data

Application data

Agent data

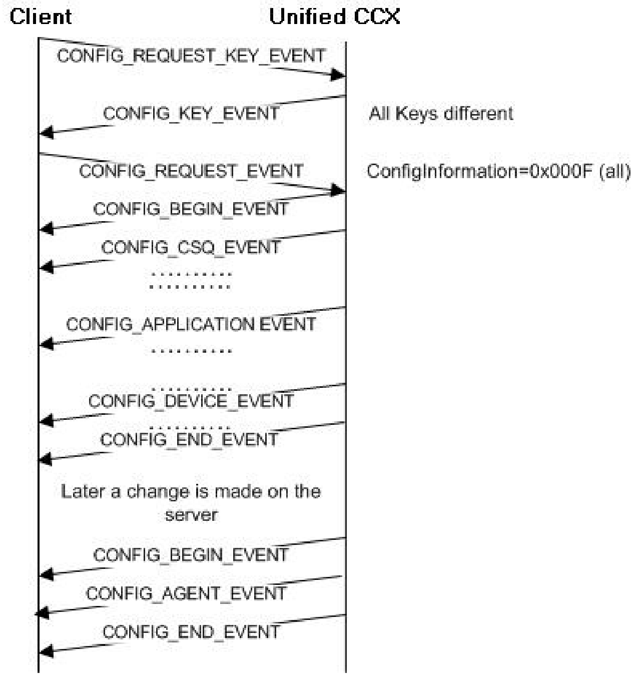
Device data

However, the order in which the client receives the data is irrelevant. The client may receive the data messages in any order depending on the timing and number of messages sent. What is important is that all the messages between a begin and end message contain consistent data. That is, for example, if a CSQ message is sent with an Agent message in the same data block, the agent referenced belongs to that CSQ.

Step 7 Configuration updates performed in the Unified CCX Administration web page are sent to clients in a CONFIG_BEGIN_EVENT and CONFIG_END_EVENT block of messages as well.

Figure 1: Configuration Message Flow, on page 6 illustrates the preceding steps

Figure 1: Configuration Message Flow



What to Do Next

Note During an initial configuration upload, if the server needs to break up the configuration into multiple CONFIG_BEGIN_EVENT/CONFIG_END_EVENT messages, a status 5 is listed in the CONFIG_END_EVENT message until all the data is sent. A status of 0 is included in the final CONFIG_END_EVENT message to indicate a successful data upload.

How Configuration Consistency is Maintained

The consistency of configuration data is managed through configuration keys. Unified CCX server always increments configuration keys when a configuration update happens in the Unified CCX Administration. The set of keys are sent to clients in the CONFIG_KEY_EVENT message and the CONFIG_BEGIN_EVENT message. Clients may store these keys.

The keys are 8-byte values and are stored as long data in Unified CCX. Configuration keys apply to the configuration events determined by the mask bit CTI_SERVICE_CONFIG_EVENTS of the ServicesRequested field of the OPEN_REQ message only.

If a client has the same set of keys as the server, it indicates that the client configuration is in sync with the server's configuration.

Note Agent Team configuration data is the exception and is not managed by these keys.

Share Agent Team Configuration Data

Procedure

Step 1 The client must include the CTI_SERVICE_SUPERVISOR bit (0x00000080) in the ServiceRequested field of an OPEN_REQ message if the client wants team configuration data. The CCX CTI server does not provide team update configuration data to the client unless that bit is set.

Step 2 When a client needs team configuration data, it sends a TEAM_CONFIG_REQ message to the Unified CCX server.
If the “configparam” field in the TEAM_CONFIG_REQ message is set to 0x1, it takes the initial configuration.

Note Clients must use the TEAM_CONFIG_REQ to request team configuration information. This information is not a part of the CONFIG_REQUEST_EVENT, as shown in Table 1: Unified CCX CTI Configuration Message List, on page 2, and must be requested explicitly.

Step 3 The server responds with zero or more TEAM_CONFIG_EVENT messages, followed by a TEAM_CONFIG_CONF message.
The TEAM_CONFIG_CONF messages mark the end of the initial configuration upload. After that, the client receives updates only for the message session.

Step 4 After this request/confirmation exchange, the Unified CCX server sends TEAM_CONFIG_EVENT messages to the client whenever there is a team configuration change in the Unified CCX Administration web page.

Error Handling

As Unified CCX and clients exchange configuration messages, Unified CCX may encounter errors. These errors are conveyed to clients through configuration messages or a generic FAILURE_CONF message and a FAILURE_EVENT message. For example, the status code in the CONFIG_KEY_EVENT message and the CONFIG_END_EVENT message may indicate an error condition in Unified CCX. Client applications are expected to check these error code and error messages. When a client detects an error condition in Unified CCX, the client needs to start over the configuration data sharing process. It is required that the client waits for at least 10 seconds before starting the process.

Although the configuration data sharing process is designed to maintain configuration consistency using the configuration key, a client developer may want to design a mechanism to overwrite the configuration key mechanism in case of unforeseen conditions in which the Unified CCX configuration and client configuration are out of sync. For example, a client developer may create a button such that whenever that button is clicked, the client starts the configuration data sharing process. In this process, the result of key comparison is ignored and CONFIG_REQUEST_EVENT message is always sent.