Messaging Conventions

CTI Message Convention

The CTI client and the CTI Server communicate by exchanging messages. Cisco's CTI Server message set is modeled after the Computer-Supported Telecommunications Applications (CSTA) messaging conventions defined by the European Computer Manufacturers Association. CTI Server messages, in general, follow CSTA naming conventions and the request/confirmation and unsolicited event paradigms. However, CTI Server messages use a simpler set of data types than those defined by CSTA.

In the CSTA model, one party acts as a server and the other as a client. In the Cisco interface, as the names suggest, the CTI client takes the client role and issues requests to the Unified CCE. The Unified CCE CTI Server takes the server role, responding to requests from the CTI clients and originating unsolicited events.

Message Types

This table defines the complete CTI server message set. The messages are described in greater detail in the remainder of this document. The length of the largest possible message (including the message header) defined by this protocol is 12500 bytes.

Table 1: Message Set

Number	Message Type	Purpose
1	FAILURE_CONF	Negative confirmation; may be sent in response to any request.
2	FAILURE_EVENT	Unsolicited notification of a failure or error.

Number	Message Type	Purpose
3	OPEN_REQ	Communication session establishment request.
4	OPEN_CONF	Communication session establishment confirmation.
5	HEARTBEAT_REQ	Communication session maintenance request.
6	HEARTBEAT_CONF	Communication session maintenance confirmation.
7	CLOSE_REQ	Communication session termination request.
8	CLOSE_CONF	Communication session termination confirmation.
9	CALL_DELIVERED_EVENT	Notification of inbound call arrival.
10	CALL_ESTABLISHED_EVENT	Notification of answering of inbound call.
11	CALL_HELD_EVENT	Notification of call placed on hold.
12	CALL_RETRIEVED_EVENT	Notification of call taken off hold.
13	CALL_CLEARED_EVENT	Notification of call termination.
14	CALL_CONNECTION_CLEARED_EVENT	Notification of the termination of a conference party connection.
15	CALL_ORIGINATED_EVENT	Notification of outbound call initiation.
16	CALL_FAILED_EVENT	Notification of inability to complete call.
17	CALL_CONFERENCED_EVENT	Notification of tandem connection of two calls.
18	CALL_TRANSFERRED_EVENT	Notification of call transfer.
19	CALL_DIVERTED_EVENT	Notification of call changing to a different service.
20	CALL_SERVICE_INITIATED_EVENT	Notification of the initiation of telecommunications service at a device (“dial-tone”).
21	CALL_QUEUED_EVENT	Notification of call being placed in a queue pending the availability of some resource.
22	CALL_TRANSLATION_ROUTE_EVENT	Notification of call context data for a call that has been routed to the peripheral by a translation route.
23	BEGIN_CALL_EVENT	Notification that a call has been associated with the CTI client.
24	END_CALL_EVENT	Notification that a call is no longer associated with a CTI client.
25	CALL_DATA_UPDATE_EVENT	Notification of a change in a call’s context data.
26	SET_CALL_DATA_REQ	Request to update one or more call variables or call wrap-up data.

Number	Message Type	Purpose
27	SET_CALL_DATA_CONF	Response confirming a previous SET_CALL_DATA request.
28	RELEASE_CALL_REQ	Notification that all call data updates are complete.
29	RELEASE_CALL_CONF	Response confirming a previous RELEASE_CALL request.
30	AGENT_STATE_EVENT	Notification of new agent state.
31	SYSTEM_EVENT	Notification of a PG Status change.
32	CLIENT_EVENT_REPORT_REQ	Request to report a CTI client event.
33	CLIENT_EVENT_REPORT_CONF	Response confirming a previous CLIENT_EVENT_REPORT request.
34	CALL_REACHED_NETWORK_EVENT	Notification of outbound call being connected to the network.
35	CONTROL_FAILURE_CONF	Response indicating the failure of a proceeding control request.
36	QUERY_AGENT_STATE_REQ	Request to obtain the current state of an agent position.
37	QUERY_AGENT_STATE_CONF	Response to a QUERY_AGENT_STATE request.
38	SET_AGENT_STATE_REQ	Request to alter the current state of an agent position.
39	SET_AGENT_STATE_CONF	Response confirming a previous SET_AGENT_STATE request.
40	ALTERNATE_CALL_REQ	Request to alternate between a held and an active call.
41	ALTERNATE_CALL_CONF	Response confirming a previous ALTERNATE_CALL request.
42	ANSWER_CALL_REQ	Request to answer an alerting call.
43	ANSWER_CALL_CONF	Response confirming a previous ANSWER_CALL request.
44	CLEAR_CALL_REQ	Request to release all devices from a call.
45	CLEAR_CALL_CONF	Response confirming a previous CLEAR_CALL request.
46	CLEAR_CONNECTION_REQ	Request to release a single device from a call.
47	CLEAR_CONNECTION_CONF	Response confirming a previous CLEAR_CONNECTION request.
48	CONFERENCE_CALL_REQ	Request to conference a held call with an active call.

Number	Message Type	Purpose
49	CONFERENCE_CALL_CONF	Response confirming a previous CONFERENCE_CALL request.
50	CONSULTATION_CALL_REQ	Request to hold an active call and start a new call.
51	CONSULTATION_CALL_CONF	Response confirming a previous CONSULTATION_CALL request.
52	DEFLECT_CALL_REQ	Request to move an alerting call to a different device.
53	DEFLECT_CALL_CONF	Response confirming a previous DEFLECT_CALL request.
54	HOLD_CALL_REQ	Request to place a call connection in the held state.
55	HOLD_CALL_CONF	Response confirming a previous HOLD_CALL request.
56	MAKE_CALL_REQ	Request to start a new call between two devices.
57	MAKE_CALL_CONF	Response confirming a previous MAKE_CALL request.
58	MAKE_PREDICTIVE_CALL_REQ	Request to start a new predictive call.
59	MAKE_PREDICTIVE_CALL_CONF	Response confirming a previous MAKE_PREDICTIVE_CALL request.
60	RECONNECT_CALL_REQ	Request to clear a connection and retrieve a held call.
61	RECONNECT_CALL_CONF	Response confirming a previous RECONNECT_CALL request.
62	RETRIEVE_CALL_REQ	Request to reconnect a held call.
63	RETRIEVE_CALL_CONF	Response confirming a previous RETRIEVE_CALL request.
64	TRANSFER_CALL_REQ	Request to transfer a held call to an active call.
65	TRANSFER_CALL_CONF	Response confirming a previous TRANSFER_CALL request.
66 to 77	Reserved	Reserved
78	QUERY_DEVICE_INFO_REQ	Request to obtain general device information.
79	QUERY_DEVICE_INFO_CONF	Response to a previous QUERY_DEVICE_INFO request.
80 to 81	Reserved	Reserved
82	SNAPSHOT_CALL_REQ	Request to obtain information about a specified call.

Number	Message Type	Purpose
83	SNAPSHOT_CALL_CONF	Response to a previous SNAPSHOT_CALL request.
84	SNAPSHOT_DEVICE_REQ	Request to obtain information about a specified device.
85	SNAPSHOT_DEVICE_CONF	Response to a previous SNAPSHOT_DEVICE request.
86	CALL_DEQUEUED_EVENT	Notification of call being removed from a queue.
87 to 90	Reserved	Reserved
91	SEND_DTMF_SIGNAL_REQ	Request to send a sequence of DTMF tones.
92	SEND_DTMF_SIGNAL_CONF	Response to a previous SEND_DTMF_SIGNAL_REQ request.
93	MONITOR_START_REQ	Request to start monitoring of a given call or device.
94	MONITOR_START_CONF	Response to a previous MONITOR_START request.
95	MONITOR_STOP_REQ	Request to terminate monitoring of a given call or device.
96	MONITOR_STOP_CONF	Response to a previous MONITOR_STOP request.
97	CHANGE_MONITOR_MASK_REQ	Request to change the message masks of a given call or device monitor.
98	CHANGE_MONITOR_MASK_CONF	Response to a previous CHANGE_MONITOR_MASK request.
99	CLIENT_SESSION_OPENED_EVENT	Notification that a new CTI Client session has been opened.
100	CLIENT_SESSION_CLOSED_EVENT	Notification that a CTI Client session has been terminated.
101	SESSION_MONITOR_START_REQ	Request to start monitoring of a given CTI Client session.
102	SESSION_MONITOR_START_CONF	Response to a previous SESSION_MONITOR_START request.
103	SESSION_MONITOR_STOP_REQ	Request to terminate monitoring of a given CTI Client session.
104	SESSION_MONITOR_STOP_CONF	Response to a previous SESSION_MONITOR_STOP request.
105	AGENT_PRE_CALL_EVENT	Advance notification of a call routed to an Enterprise Agent.

Number	Message Type	Purpose
106	AGENT_PRE_CALL_ABORT_EVENT	Cancellation of advance notification of a call routed to an Enterprise Agent.
107	USER_MESSAGE_REQ	Request to send a message to other CTI Server clients.
108	USER_MESSAGE_CONF	Response to a previous USER_MESSAGE_REQ request.
109	USER_MESSAGE_EVENT	Notification of a message sent by another CTI Server client.
110	REGISTER_VARIABLES_REQ	Request to register call context variables used by application.
111	REGISTER_VARIABLES_CONF	Response to a previous REGISTER_VARIABLES_REQ request.
112	QUERY_AGENT_STATISTICS_REQ	Request for current agent call handling statistics.
113	QUERY_AGENT_STATISTICS_CONF	Response to a previous QUERY_AGENT_STATISTICS_REQ request.
114	QUERY_SKILL_GROUP_STATISTICS_REQ	Request for current skill group call handling statistics.
115	QUERY_SKILL_GROUP_STATISTICS_CONF	Response to a previous QUERY_SKILL_GROUP_STATISTICS_REQ request.
116	RTP_STARTED_EVENT	Indicates that an RTP input has been started.
117	RTP_STOPPED_EVENT	Indicates that an RTP input has been stopped.
118	SUPERVISOR_ASSIST_REQ	An agent requests for assistance to their supervisor.
119	SUPERVISOR_ASSIST_CONF	Response to a previous SUPERVISOR_ASSIST_REQ request.
120	SUPERVISOR_ASSIST_EVENT	Notification of a supervisor assist request sent by a CTI Server client.
121	EMERGENCY_CALL_REQ	An agent declaring an emergency situation to their supervisor.
122	EMERGENCY_CALL_CONF	Response to a previous EMERGENCY_CALL_REQ request.
123	EMERGENCY_CALL_EVENT	Notification of an emergency call request sent by a CTI Server client.
124	SUPERVISE_CALL_REQ	A supervisor request to perform monitor or barge-in operations.

Number	Message Type	Purpose
125	SUPERVISE_CALL_CONF	Response to a previous SUPERVISE_CALL_REQ request.
126	AGENT_TEAM_CONFIG_REQ	Request sent by client to CTI Server, to change agent team configuration.
127	AGENT_TEAM_CONFIG_CONF	Response to a previous AGENT_TEAM_CONFIG_REQ request.
128	AGENT_TEAM_CONFIG_EVENT	Notification of passing the team member list.
129	SET_APP_DATA_REQ	Request to update one or more application variables.
130	SET_APP_DATA_CONF	Response confirming a previous SET_APP_DATA request.
131	AGENT_DESK_SETTINGS_REQ	Request to obtain Agent Desk Settings.
132	AGENT_DESK_SETTINGS_CONF	Response to a previous AGENT_DESK_SETTINGS_REQ request.
133	LIST_AGENT_TEAM_REQ	Request to obtain a list of Agent Teams.
134	LIST_AGENT_TEAM_CONF	Response to a previous LIST_AGENT_TEAM_REQ request.
135	MONITOR_AGENT_TEAM_START_REQ	Request to start monitoring an Agent Team.
136	MONITOR_AGENT_TEAM_START_CONF	Response to a previous MONITOR_AGENT_TEAM_START_REQ request.
137	MONITOR_AGENT_TEAM_STOP_REQ	Request to stop monitoring an Agent Team.
138	MONITOR_AGENT_TEAM_STOP_CONF	Response to a previous MONITOR_AGENT_TEAM_STOP_REQ request.
139	BAD_CALL_REQ	Request to mark a call as having poor voice quality.
140	BAD_CALL_CONF	Response to a previous BAD_CALL_REQ request.
141	SET_DEVICE_ATTRIBUTES_REQ	Request to set the default attributes of a calling device.
142	SET_DEVICE_ATTRIBUTES_CONF	Response to a previous SET_DEVICE_ATTRIBUTES_REQ request.
143	REGISTER_SERVICE_REQ	Request to register service for the server application.
144	REGISTER_SERVICE_CONF	Response to a previous REGISTER_SERVICE_REQ request.
145	UNREGISTER_SERVICE_REQ	Request to unregister service for the server application.

Number	Message Type	Purpose
146	UNREGISTER_SERVICE_CONF	Response to a previous UNREGISTER_SERVICE_REQ request.
147	START_RECORDING_REQ	Request to start recording.
148	START_RECORDING_CONF	Response to a previous START_RECORDING_REQ request.
149	STOP_RECORDING_REQ	Request to stop recording.
150	STOP_RECORDING_CONF	Response to a previous STOP_RECORDING_REQ request.
151	MEDIA_LOGIN_REQ	Report agent sign in to MRD.
152	MEDIA_LOGIN_RESP	Response to MEDIA_LOGIN_REQ.
153	MEDIA_LOGOUT_IND	Report agent sign out from MRD.
154	MAKE_AGENT_ROUTABLE_IND	Make agent routable for MRD request.
155	MAKE_AGENT_NOT_ROUTABLE_REQ	Make agent not routable for MRD request.
156	MAKE_AGENT_NOT_ROUTABLE_RESP	Response to MAKE_AGENT_NOT_ROUTABLE_REQ.
157	MAKE_AGENT_READY_IND	Report agent made ready.
158	MAKE_AGENT_NOT_READY_REQ	Report agent made not ready.
159	MAKE_AGENT_NOT_READY_RESP	Response to MAKE_AGENT_NOT_READY_REQ.
160	OFFER_TASK_IND	Report agent has been offered task, agent selected by Unified CCE.
161	OFFER_APPLICATION_TASK_REQ	Report agent has been offered task, agent not selected by Unified CCE.
162	OFFER_APPLICATION_TASK_RESP	Response to OFFER_APPLICATION_TASK_REQ.
163	START_TASK_IND	Report agent has begun task, agent selected by Unified CCE.
164	START_APPLICATION_TASK_REQ	Report agent has begun task, agent not selected by Unified CCE.
165	START_APPLICATION_TASK_RESP	Response to START_APPLICATION_TASK_REQ.
166	PAUSE_TASK_IND	Report agent has paused task.
167	RESUME_TASK_IND	Report agent has resumed task.
168	WRAPUP_TASK_IND	Report agent has entered wrap-up for task.

Number	Message Type	Purpose
169	END_TASK_IND	Report agent has ended task.
170	AGENT_MADE_NOT_ROUTABLE_EVENT	Notify client that agent made not routable for MRD.
171	AGENT_INTERRUPT_ADVISORY_EVENT	Notify client that agent has been interrupted by noninterruptible task.
172	AGENT_INTERRUPT_ACCEPTED_IND	Report acceptance of the interrupt.
173	AGENT_INTERRUPT_UNACCEPTED_IND	Report nonacceptance of the interrupt.
174	AGENT_INTERRUPT_DONE_ADVISORY_EVENT	Notify client that interrupt has been ended.
175	AGENT_INTERRUPT_DONE_ACCEPTED_IND	Report acceptance of interrupt end.
176	CHANGE_MAX_TASK_LIMIT_REQ	Change the maximum number of simultaneous tasks for the agent MRD combination.
177	CHANGE_MAX_TASK_LIMIT_RESP	Response to CHANGE_MAX_TASK_LIMIT_REQ.
178	OVERRIDE_LIMIT_REQ	Request a task assignment even though it would exceed agent's maximum number of simultaneous tasks for the MRD.
179	OVERRIDE_LIMIT_RESP	Response to OVERRIDE_LIMIT_REQ.
180	UPDATE_TASK_CONTEXT_IND	Update Unified CCE task context.
181	BEGIN_AGENT_INIT_IND	Report begin agent and task resynchronization.
182	AGENT_INIT_REQ	Report agent's current state.
183	AGENT_INIT_RESP	Response to AGENT_INIT_REQ.
184	END_AGENT_INIT_IND	Report end of agent and task resynchronization.
185	TASK_INIT_IND	Report task's state.
186	AGENT_INIT_READY_EVENT	Notify client that Unified CCE is ready to receive agent and task resynchronization messages.
187	GET_PRECALL_MESSAGES_REQ	Request any pending PRE-CALL messages.
188	GET_PRECALL_MESSAGES_RESP	Response to GET_PRECALL_MESSAGES_REQ.
189	AGENT_LEGACY_PRE_CALL_EVENT	Current task context.
190	FAILURE_RESP	Failure response to ARM indication messages.
191	BEGIN_TASK_EVENT	Indicates that the specified task has entered the system, either queued, offered, or begun.
192	QUEUED_TASK_EVENT	Indicate that the specified task has been queued in the router.

Number	Message Type	Purpose
193	DEQUEUED_TASK_EVENT	Indicate that the specified task has been dequeued from the router.
194	OFFER_TASK_EVENT	Indicates that the specified agent has been reserved to handle the specified task.
195	START_TASK_EVENT	Indicates that the specified agent has started handling the task.
196	PAUSE_TASK_EVENT	Indicates that the specified agent has temporarily suspended handling of the specified task.
197	RESUME_TASK_EVENT	Indicates that the specified agent has resumed handling of the specified task after having previously sent a Pause Task message.
198	WRAPUP_TASK_EVENT	Indicates that the specified agent is no longer actively handling the task but is doing followup work related to the task.
199	END_TASK_EVENT	Indicates that the specified agent has ended handling of the specified task.
200	TASK_DATA_UPDATE_EVENT	Update task context for the specified task.
201	TASK_MONITOR_START_REQ	Request to start the task monitor with the task mask in the request message.
202	TASK_MONITOR_START_CONF	Response to TASK_MONITOR_START_REQ.
203	TASK_MONITOR_STOP_REQ	Request to stop the task monitor with the monitor ID in the request message.
204	TASK_MONITOR_STOP_CONF	Response to TASK_MONITOR_STOP_REQ.
205	CHANGE_TASK_MONITOR_MASK_REQ	Request to change the task monitor mask with the new mask in the request message.
206	CHANGE_TASK_MONITOR_MASK_CONF	Response to CHANGE_TASK_MONITOR_MASK_REQ.
207	MAX_TASK_LIFETIME_EXCEEDED_EVENT	Unified CCE terminated a task which had exceeded its configured maximum lifetime. The result is equivalent to the task ending due to an end task but with a special reason code in the Termination Call Detail record.
208	SET_APP_PATH_DATA_IND	Set or update the application path-specific data variables available to routing scripts.
209	TASK_INIT_REQ	Report task's state. Use this when a Unified CCE taskID is not yet assigned to the task because the task began when the ARM client interface was down.

Number	Message Type	Purpose
210	TASK_INIT_RESP	Response to the TASK_INIT_REQ message.
211	ROUTE_REGISTER_EVENT	Register to receive route requests.
212	ROUTE_REGISTER_REPLY_EVENT	Reply to registration message.
213	ROUTE_REQUEST_EVENT	Route request for a destination for a call.
214	ROUTE_SELECT_EVENT	Supplies a route destination for a route request.
215	ROUTE_END_EVENT	End Routing dialog.
216 to 229	Reserved	Reserved
230	CONFIG_REQUEST_KEY_EVENT	Sent by client to CTI Server, to request configuration keys for different items.
231	CONFIG_KEY_EVENT	Response to previous CONFIG_REQUEST_KEY_EVENT request.
232	CONFIG_REQUEST_EVENT	Sent by client to CTI Server, to receive configuration.
233	CONFIG_BEGIN_EVENT	Signifies the beginning of configuration
234	CONFIG_END_EVENT	Signifies the end of configuration
235	CONFIG_SERVICE_EVENT	Sent by the CTI Server to client, to update information about a service or application.
236	CONFIG_SKILL_GROUP_EVENT	Sent by the CTI Server to client, to update information about skill group configuration.
237	CONFIG_AGENT_EVENT	Request sent by the CTI Server to client, to update information about agent.
238	CONFIG_DEVICE_EVENT	Request sent by the CTI Server to client, to update information about a device.
239 to 241	Reserved	Reserved
242	TEAM_CONFIG_REQ	Request sent by client to CTI server, to request team configuration data.
243	TEAM_CONFIG_EVENT	Response to previous TEAM_CONFIG_REQ request.
244	TEAM_CONFIG_CONF	Sent by the CTI Server to client, to mark end of team configuration data.
245	CONFIG_CALL_TYPE_EVENT	Sent by the CTI server to client, to provide information about a call type.
246 to 247	Reserved	Reserved
248	CALL_AGENT_GREETING_EVENT	Status Notification of Agent Greeting request.

Number	Message Type	Purpose
249	AGENT_GREETING_CONTROL_REQ	Stop the greeting that is playing; disable or enable the Agent Greeting feature for this current sign-in session.
250	AGENT_GREETING_CONTROL_CONF	Confirmation of AGENT_GREETING_CONTROL_REQ.
251 to 253	Reserved	Reserved
254	CONFIG_MRD_EVENT	Sent by the CTI server to client, to provide information about a Media Routing Domain.
255	GET_AGENT_TASKS_REQ	Request sent to obtain an agent's Tasks list in a specified MRD. The message acts as an indication to a PG that the client has reconnected; the PG then recalculates the agent's state based on the Tasks the agent has. If there are no tasks, the agent state is Not-Ready.
256	AGENT_TASKS_RESP	Sent by the CTI Server to client, as a response to a previous GET_AGENT_TASKS_REQ message.
257	SNAPSHOT_TASK_REQ	Request sent to obtain information about a specified agent's task.
258	SNAPSHOT_TASK_RESP	Sent by the CTI Server to client, as a response to a previous SNAPSHOT_TASK_REQ message.
259	Reserved	Reserved
260	CONFIG_PERIPHERAL_EVENT	Configuration message for peripheral devices.
261	CONFIG_AGENT_DESK_SETTINGS_EVENT	Configuration message for Agent Desk Settings.
262 -267	Reserved	Reserved.

Data Types

This table lists the data types that define fields within messages. All numeric data longer than 1 byte are sent in order of most significant byte to least significant byte. This is the canonical network byte order defined by TCP/IP standards.

Table 2: Data Types

Data Type	Meaning	Byte Size
CHAR	Signed integer, –128 to 127.	1
UCHAR	Unsigned integer, 0 to 255.	1
SHORT	Signed integer, –32,768 to 32,767.	2

Data Type	Meaning	Byte Size
USHORT	Unsigned integer, 0 to 65,535.	2
INT	Signed Integer, –2,147,483,648 to 2,147,483,647.	4
UINT	Unsigned Integer, 0 to 4,294,967,295.	4
BOOL	Boolean (False = 0, True = 1).	2
STRING[n]	ASCII string of length n.	n
UNSPEC[n]	Unspecified data occupying n consecutive bytes.	n
TIME	A date/time, expressed as the number of seconds since midnight January 1, 1970 Coordinated Universal Time (UTC).	4
MHDR	Message header	8
NAMEDVAR	A named call context variable	3 … 251
NAMEDARRAY	A named call context array element	4 … 252
TASKID	Task group identifier	12
APPPATHID	Application path identifier	5

MHDR Data Type

The MHDR data type is a common message header that precedes all messages exchanged between a CTI client and the CTI Server. This table defines the message header format.

Table 3: Message Header (MHDR) Format

Field Name	Value	Data Type	Byte Size
MessageLength	The length of the message in bytes, excluding the size of the message header (the first 8 bytes).	UINT	4
MessageType	The type of message. This value determines the format of the remainder of the message.	UINT	4

NAMEDVAR Data Type

The NAMEDVAR data type is a call context variable that is defined in the Unified CCE Expanded_Call_Variable_Table. This variable-length data type may appear in the floating part of a message and has the format shown in this table:

Table 4: Named Call Context Variable (NAMEDVAR) Format

Subfield	Value	Data Type	Max. Size
Tag	NAMED_VARIABLE_TAG (= 82). The floating field tag that indicates that the following data is a named call context variable.	UCHAR	2
FieldLength	The total length of the VariableName and Variable Value fields, including the null-termination bytes. The value of this field may range from 3 to 251.	UCHAR	2
VariableName	The null-terminated defined name of the variable.	STRING	33
VariableValue	The null-terminated value of the variable.	STRING	211

NAMEDARRAY Data Type

The NAMEDARRAY data type is a call context variable that is defined in the Unified CCE Expanded_Call_Variable_Table. This variable length data type may appear in the floating part of a message and has the format shown in this table:

Table 5: Named Call Context Array Variable (NAMEDARRAY) Format

Subfield	Value	Data Type	Max. Size
Tag	NAMED_ARRAY_TAG (= 83). The floating field tag that indicates that the following data is a named call context array variable.	UCHAR	2
FieldLength	The total length of the VariableIndex, Variable Name, and VariableValue fields, including the null-termination bytes. The value of this field may range from 4 to 252.	UCHAR	2
VariableIndex	The index of the array variable.	UCHAR	1
VariableName	The null-terminated defined name of the array variable.	STRING	33
VariableValue	The null-terminated value of the array variable.	STRING	211

TASKID Data Type

This table defines the TASKID field format.

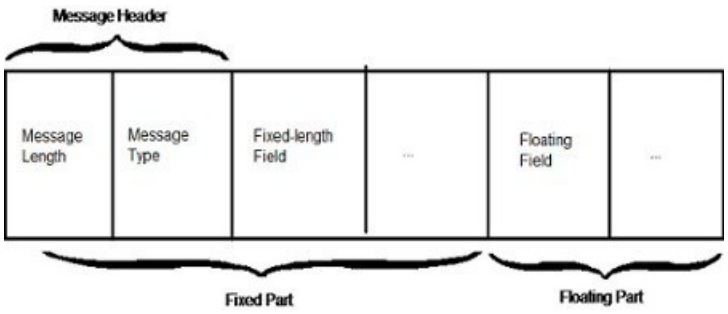
Table 6: TASKID Format

Field Name	Value	Data Type	Byte Size
TaskGroupHigh	The most significant 4 bytes of the Task Group ID. The Task Group ID links multiple Termination Call Detail (TCD) records together for reporting purposes. Use this when the same customer interaction involves multiple tasks over time. For example, this might happen if an agent stops the work and then another agent restarts it.	INT	4
TaskGroupLow	The least significant 4 bytes of the Task Group ID.	INT	4
SequenceNumber	The Task Group ID is unchanged for the lifetime of all tasks that are related to the group. The combination of Task Group ID and Sequence Number is unique for every termination record.	INT	4

Message Formats

Messages contain either a fixed part only or a fixed part and a floating part. The fixed part of a message contains the message header and all required, fixed length fields. The variable part of a message immediately follows the fixed part. It contains one or more floating fields that are optional and/or variable in length. The message type field in the message header determines the format of the message, and therefore indicates if the message includes a floating part and what types of floating fields may appear within it.

Figure 1: CTI Server Message Format

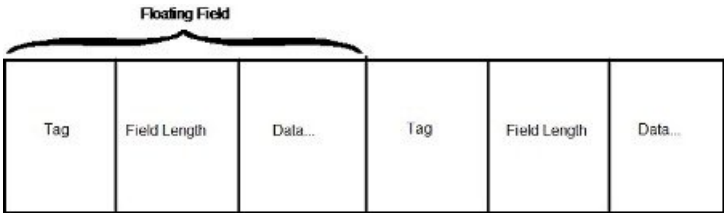


Floating Fields

Each floating field has the same format. The field begins with a two-byte tag, which identifies the field type. Following the tag is a two-byte field length, which indicates the number of bytes of data in the field (excluding the tag and field length). The data immediately follows the FieldLength. The maximum size listed for each floating field is the maximum number of data bytes allowed. It does not include the tag and field length bytes. For string data, it includes the null termination byte.

Floating fields are packed together in the floating part of the message. The tag of one floating field immediately follows the data of the previous field. The message length (in the message header) indicates the end of the message. This figure shows the format of a floating field.

Figure 2: CTI Server Floating Field Format



Within the floating part, floating fields may appear in any order. In general, each floating field appears only once unless the field is a member of a list. In this case, a fixed field in the message indicates the number of list entries present. This table defines the format of the floating field:

Table 7: Floating Field Subfields

Subfield	Value	Data Type	Byte Size
Tag	The type of the floating field	USHORT	2
FieldLength	The number of bytes, n, in the Data subfield of the floating field.	USHORT	2
Data	The data	Depends on field type	n

For a list of possible floating field tag values, see the Tag Values table.

Related Topics

Tag Values

Call Event Data

The Cisco CTI Interface presents Call Event data using a CSTA-like model; however, the underlying ACD datalink may or may not conform to this model. This means that, depending upon the type of ACD being used, some Call Event messages may not be generated, and some of the CSTA message data for other events may not be available. Be aware that the interpretation of Call Event data is very peripheral-specific, particularly when multiple ACD types are being used.

For a discussion of peripheral-specific considerations, see the *CTI OS Developer Guide for Cisco Unified ICM*.

Device IDs

The Call Event messages detailed later in this document typically provide several different device ID fields. Depending upon the type of peripheral and the nature of the event, the device ID may represent a Trunk number, a Trunk Group number, or an agent teleset number (extension). Some peripheral types may not provide a device ID for one or more fields. To handle these situations, the Call Event messages provide device

IDs using two fields: a fixed field indicating whether or not the device ID was provided and enumerating the type of device identified, and a floating field containing the device ID (if provided).

CTI Client History

The Call Event messages also provide a list of CTI clients associated with the current call (if any). This information is provided using a separate floating field for each CTI client in the list, and a fixed field providing a count of the number of entries in the list. Each list entry's floating field uses the same tag value.

Event Cause Codes

Most Call Event messages include an EventCause fixed field that may provide a reason for the occurrence of the event. Usually no event cause information is supplied (CEC_NONE).

For a list of EventCause codes that may be reported, see the EventCause Values table.

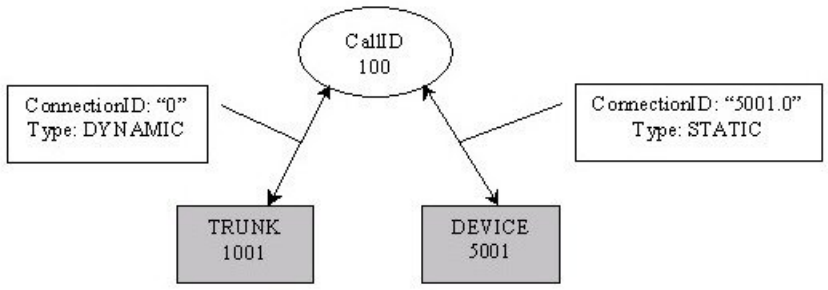
Related Topics

EventCause Values

Call Identification

CTI Server uses the CSTA method of identifying calls. A numeric ConnectionCallID identifies a call; each connection of a device to that call is identified by a ConnectionDeviceID string and an enumerated ConnectionDeviceIDType value. All call related messages identify the ConnectionCallID as well as the ConnectionDeviceIDType and ConnectionDeviceID of the call connection that is the subject of the event.

Figure 3: Sample CSTA Call/Device/ConnectionID Values



A ConnectionDeviceID uniquely identifies a call connection. However, it cannot directly identify the connected device; use other event message fields for that purpose. In some cases, the ConnectionDeviceID may simply be the ID of the connected device, the connected deviceID with additional identifying data included, or a string that does not contain the deviceID at all. A valid CTI Server application can make no assumption about the content or format of a ConnectionDeviceID.

Occasionally, both the ConnectionDeviceID and the numeric ConnectionCallID are required in order to properly identify the subject call. This occurs when the ACD uses the ConnectionCallID value from an ACD call as the ConnectionCallID value for any related consultative calls. This poses two particularly significant requirements for applications: they must be able to keep track of two calls with the same numeric ConnectionCallID value, and they must be able to decide which of the two calls is being referenced by any given call event message. These requirements are relatively easy to implement by keeping track of the

ConnectionDeviceIDs associated with each call. The call that has a ConnectionDeviceID that matches the ConnectionDeviceID provided in the call event message is the call that is the subject of the event. The only difficult case is determining which call is the subject when a new call connection is created. For this case, the following rule applies:

- When more than one call with the same ConnectionCallID value exists, the connection being created by a CALL_ESTABLISHED_ EVENT shall apply to the call that does not yet have a destination connection established.

Typically, when this occurs, one call will have been the subject of a prior CALL_ESTABLISHED_EVENT and will have two connections; the other will have only one originating connection. The CALL_ESTABLISHED_EVENT will therefore create the second connection on that call. It should never be the case that both calls have already been the subject of a CALL_ESTABLISHED_EVENT.

Failure Indication Messages

The CTI Server may indicate errors to the CTI client using the FAILURE_CONF and FAILURE_EVENT messages. The CTI Server may use the FAILURE_CONF message in response to any request message from the CTI client. The CTI Server sends the FAILURE_CONF message instead of the positive confirmation message specific to the request. The format of the FAILURE_CONF message is defined in this table:

Table 8: FAILURE_CONF Message Format

Field Name	Value	Data Type	Byte Size
MessageHeader	Standard message header. MessageType = 1.	MHDR	8
InvokeID	Set to the value of the InvokeID from the corresponding request message.	UINT	4
Status	A status code indicating the cause of the failure. The possible status codes are defined in the Failure Indication Message status code table.	UINT	4

The CTI Server may use the FAILURE_EVENT message to asynchronously indicate a failure or error condition to the CTI client. The format of the FAILURE_EVENT message is defined in this table:

Table 9: FAILURE_EVENT Message Format

Field Name	Value	Data Type	Byte Size
MessageHeader	Standard message header. MessageType = 2.	MHDR	8
Status	A status code indicating the cause of the failure. The possible status codes are defined in the Failure Indication Message status code table.	UINT	4

Related Topics

Failure Indication Message Status Codes