Event Interfaces and Events

Event Interfaces and Events

This chapter describes the CTI OS Client Interface Library event publications mechanism. Programs written to take advantage of CTI interfaces are generally event driven, meaning that a code module in the application is executed when an external event arrives. The CIL interface provides a rich set of event interfaces and events for use by client programmers.

Events are generated asynchronously, either by the telephony equipment (for example, phone, PBX, and ACD) or by the CTI environment (CTI Server, or CTI OS Server). Each event passes an Arguments structure of key-value pairs that contains all of the event parameters. These parameters are discussed in greater detail in this chapter.

Event Publication Model

Note The CIL event interfaces discussed in this section and the following sections apply only to the C++, COM, and VB interfaces. For more information about a discussion of Java CIL counterpart events and event handling in the Java CIL, see Events in Java CIL, on page 115. For more information about a discussion of .NET CIL event handling, see Events in .NET CIL, on page 116.

The Client Interface Library provides a publisher-subscriber model for notifying clients of events. Client applications using the CIL can subscribe to one or more of the CIL event interfaces. For more information and examples on how to subscribe and unsubscribe for events, see Building Your Custom CTI Application

The published CIL event interfaces are organized around the different classes of CTI objects that the CIL provides. The event interfaces described in this chapter are:

- **ISessionEvents**. This interface publishes the events that relate to actions on the Session object.
- **ICallEvents**. This interface publishes the events that relate to actions on Call objects.
- **IAgentEvents**. This interface publishes the events that relate to actions on Agent objects.
- **ISkillGroupEvents**. This interface publishes the events that relate to actions on SkillGroup objects.
- **IButtonEnablementEvents**. This interface publishes the events that relate to changes in the enable-disable status of softphone buttons.
- **ISilentMonitorEvents**. This interface sends events to subscribers of the Silent Monitor interface.
- **IMonitoredAgentEventsInterface**. This interface fires Agent events to a supervisor for his team members.
- **IMonitoredCallEventsInterface**. This interface sends Call events to a supervisor for one of his agent team members.
- **LogEventsAdapter** (Java only). This class provides the default implementation for the message handlers in ILogEvents.
- **IGenericEvents**. This interface sends generic events to subscribers of the IGenericEvents interface.

The remainder of this chapter provides the detailed description of each event interface available from the CIL.

Note The data type listed for each keyword is the standardized data type discussed in the section CTI OS CIL Data Types in CIL Coding Conventions For more information about the appropriate language specific types for these keywords, see Table 1.

ISessionEvents Interface

The Session object fires events on the ISessionEvents interface. The following events are published to subscribers of the ISessionEvents interface.

OnConnection

The OnConnection event is generated after the Connect method succeeds. It returns the name of the connected server and the connection time of day. The client application need not take any special action but can use it to display connection status.

Syntax

```
C++: void OnConnection(Arguments& args)
COM: void OnConnection (IArguments * args)
VB:  session_OnConnection (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 1: ISession Event Parameters

Keyword	Type	Description
EventTime	INT	Integer value with time of day expressed in milliseconds.
CurrentServer	STRING	Name or TCP/IP address of the current connected CTI OS server.

OnConnectionClosed

The OnConnectionClosed message is generated when a connection is terminated by the client. This message has no fields. This event indicates successful completion of an action that the client (CIL or application using the CIL) initiated. By contrast, the OnConnectionFailure event is generated when the connection terminated for reasons other than that the client closed the connection.

OnConnectionFailure

The OnConnectionFailure event is generated when an established connection fails. It returns the name of the failed connected server and the failure time of day. Retry is automatic and is followed by an OnConnection event when connection is successfully reestablished. The client application need not take any special action but can use this event to display connection status.

Syntax

C++

```
void OnConnectionFailure(Arguments& args)
```

COM

```
void OnConnectionFailure (IArguments * args)
```

VB

```
session_OnConnectionFailure (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 2: OnConnectionFailure Event Parameters

Keyword	Type	Description
EventTime	INT	Integer value with time of day expressed in milliseconds.
FailedServer	STRING	Name or TCP/IP address of the server that failed to respond. See ReasonCode.
ReasonCode	SHORT	Reason code 0 : eProtocolMismatch Reason code 1 : eMissedHeartbeats Reason code 2 : eTransportError Reason code 3 : eConnectFail Reason code 4 : eOtherError

OnConnectionRejected

The OnConnectionRejected event indicates that the client tried to make a connection using incompatible versions of the CTI OS Server and CTI OS CIL.

Syntax

```
C++:  void OnConnectionRejected (Arguments& args)
COM:  void OnConnectionRejected (IArguments * args)
VB:   Session_OnConnectionRejected (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Not currently used, reserved for future use.

OnCTIOSFailure

The OnCTIOSFailure event indicates that the CTI Server fired either a FailureConf or a SystemEvent.

Note CTI OS CIL sends the disconnect request to CTI OS Server when the login attempt fails. Hence, CTI OS Server closes the client connection.

Syntax

```
C++:  void OnCTIOSFailure (Arguments& args)
COM:  void OnCTIOSFailure (IArguments * args)
VB:   Session_OnCTIOSFailure (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 3: OnCTIOSFailure Event Parameters

Keyword	Type	Description
FailureCode	INT	A value according to an enumerated value, as shown immediately following this table.
SystemEventID	INT	Present only if FailureCode equals ServerConnectionStatus. Contains a value according to an enumerated value, as shown immediately following this table.
SystemEventArg1	INT	Present only if SystemEventID equals SysPeripheralOnline or SysPeripheralOffline. Contains the peripheral ID of the affected peripheral.
ErrorMessage	STRING	An error message.

Following are the enumerated values for FailureCode:

```
enum enumCTIOS_FailureCode
{
eDriverOutOfService = 1,
eServiceNotSupported = eDriverOutOfService + 1,
eOperationNotSupported = eServiceNotSupported + 1,
eInvalidPriviledge = eOperationNotSupported + 1,
eUnknownRequestID = eInvalidPriviledge + 1,
eUnknownEventID = eUnknownRequestID + 1,
eUnknownObjectID = eUnknownEventID + 1,
eRequiredArgMissing = eUnknownObjectID + 1,
eInvalidObjectState = eRequiredArgMissing + 1,
eServerConnectionStatus = eInvalidObjectState + 1,
eInconsistentAgentData = eServerConnectionStatus + 1,
eAgentAlreadyLoggedIn = eInconsistentAgentData + 1,
eForcedNotReadyForConfigError = eAgentAlreadyLoggedIn + 1
eMonitorModeConnectionDenied = eForcedNotReadyForConfigError + 1
};
```

Following are the enumerated values for SystemEventID:

```
enum enumCTIOS_SystemEventID
{ eSysCentralControllerOnline = 1,
eSysCentralControllerOffline = 2,
```

```
eSysPeripheralOnline = 3,
eSysPeripheralOffline = 4,
eSysTextFYI = 5,
eSysPeripheralGatewayOffline = 6,
eSysCtiServerOffline = 7,
eSysCTIOSServerOnline = 8,
eSysHalfHourChange = 9,
eSysInstrumentOutOfService = 10,
eSysInstrumentBackInService = 11,
eSysCtiServerDriverOnline = eSysInstrumentBackInService + 1,
eSysCtiServerDriverOffline = eSysCtiServerDriveOnline + 1,
eSysCTIOSServerOffline = eSysCtiServerDriverOffline + 1,
eSysCTIOSServerOnline = eSysCTIOSServerOffline + 1,
eSysAgentSummaryStatusOnline = eSysCTIOSServerOnline + 1,
eSysAgentSummaryStatusOffline = eSysAgentSummaryStatusOnline + 1
}
```

Remarks

See the descriptions of the CtiOs_Enums.FailureCode and CtiOs_Enums.SystemEvent interfaces in the Javadoc for information on Java CIL enumerations.

OnCurrentAgentReset

The OnCurrentAgentReset message is generated when the current agent is removed from the session.

Syntax

C++

```
void OnCurrentAgentReset(Arguments& args)
```

COM

```
void OnCurrentAgentReset (IArguments * args)
```

VB

```
session_OnCurrentAgentReset (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 4: OnCurrentAgentReset Parameters

Keyword	Type	Description
UniqueObjectID	STRING	Unique object ID (if any) of the old current agent that was just removed.

OnCurrentCallChanged

The OnCurrentCallChanged message is generated when the current call changes to another call.

Syntax

C++

```
void OnCurrentCallChanged(Arguments& args)
```

COM

```
void OnCurrentCallChanged (IArguments * args)
```

VB

```
session_OnCurrentCallChanged (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 5: OnCurrentCallChanged Parameters

Keyword	Type	Description
UniqueObjectID	STRING	Unique object ID (if any) of the new current call.

OnFailure Event

Not supported.

OnGlobalSettingsDownloadConf

You can configure the client after you are in the CTI OS Server and then download this configuration to each CTI OS client desktop. When an application executes the RequestDesktopSettings method call on the Session, an eGlobalSettingsDownloadRequest event is sent to the server.

In response, the server sends an OnGlobalSettingsDownloadConf event back to the calling application. The Arguments object passed as a parameter in this event contains the Desktop Settings configuration information. The Arguments object is an array that can contain up to seven elements, each of which has the value of a nested Arguments array in a hierarchy that closely matches that of the CTI OS server configuration in the Windows registry.

Each of these Arguments arrays is a packed version of the configuration contained in the CTI OS Server. For more information, see *CTI OS System Manager Guide for Cisco Unified ICM*.

This section describes the contents of the Arguments array returned in the OnGlobalSettingsDownloadConf event. Custom applications can add values at the lowest level under each key. Custom values added in this way are passed to the client in this event. This section also identifies which keys and values in the CTI OS registry are passed to the client in this event.

For more information about what is available and how to configure these items, see the following sections in the *CTI OS System Manager Guide for Cisco Unified ICM*:

- MainScreen
- Defining Connection Profiles
- Declaring ECC Variables

- Configuring the Call Appearance Grid
- Automatic Agent Statistics Grid Configuration
- Automatic Skill Group Statistics Grid Configuration

Syntax

C++

```
void OnGlobalSettingsDownloadConf(Arguments & args)
```

COM

```
void OnGlobalSettingsDownloadConf(IArguments * args)
```

VB

```
session_OnGlobalSettingsDownloadConf(ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

An Arguments array containing the Enterprise Desktop Settings configuration from a CTI OS server. For more information about the Enterprise Desktop Settings values listed below, see the *CTI OS System Manager Guide for Cisco Unified ICM*.

The following are the top level elements in the Enterprise Desktop Settings registry key. The CTI OS server passes configuration data for these elements to the client in the OnGlobalSettingsConf event:

- ECC (Expanded Call Context) variables
- Grid
- IPCCSilentMonitor
- Login
- ScreenPreferences
- SoundPreferences

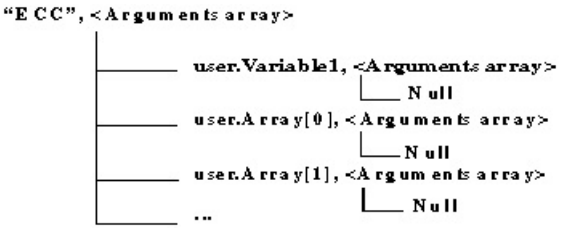
Other keys or values that are added to the EnterpriseDesktopSettings/All Desktops key in the CTI OS server registry are passed to the client in the DesktopSettings Arguments array as empty Arguments arrays.

The following sections describe the contents of the args array:

- **ECC** – Arguments array that contains the Expanded Call Context (ECC) variables declared on the CTI OS server in the "ECC/Name" registry subtree (the following figure).

  The CTI OS server does not send any registry information contained in the CTI OS registry keys representing the ECC scalar and array names. Thus the ECC Arguments arrays are empty in the OnGlobalSettingsDownloadConf event, regardless of the contents of those keys.

Figure 1: ECC Arguments Array



Each ECC scalar configured in the CTI OS server registry is represented as an empty Arguments array with keyword "user.<name>", where <name> is the ECC name as configured on CTI OS server.

Each ECC array configured in the CTI OS server registry is represented as multiple empty Arguments arrays with keywords "user.<name>[0]" to "user.<name>[n-1]", where <name> is the ECC name as configured on the CTI OS server and n is the size of the array as configured on the CTI OS server.

- **Grid** – Arguments array contains information from the CTI OS server registry's Grid subtree. The grid element contains an Arguments array of up to three Arguments arrays:
  - AgentStatistics
  - CallAppearance
  - SkillGroupStatistics

Each of these arrays contains the keyword "columns," an Arguments array that contains multiple nested Arguments arrays with key=<column_number>, where <column_number> corresponds to the name of a key within the Columns/Number registry subtree.

The configuration information for any key or value added to the SkillGroupStatistics, AgentStatistics, or CallAppearance registry keys is not passed to the client in the OnGlobalSettingsDownloadConf event.

The value for each column number in the AgentStatistics and SkillGroupStatistics element is an Arguments array containing the following key-value pairs:

Table 6: Agent Statistics Column Number: Key Values

Keyword	Data Type
Type	string
Header	string
Custom values[1]	custom

[1] Other registry values added to the <column_number> registry key are passed in the OnGlobalSettingsDownloadConf event. Subkeys added to the <column_number> registry key are not passed in this event.

The value for each column number in the CallAppearance element is an Arguments array containing the following key-value pairs:

Table 7: CallAppearance Column Number: Key Values

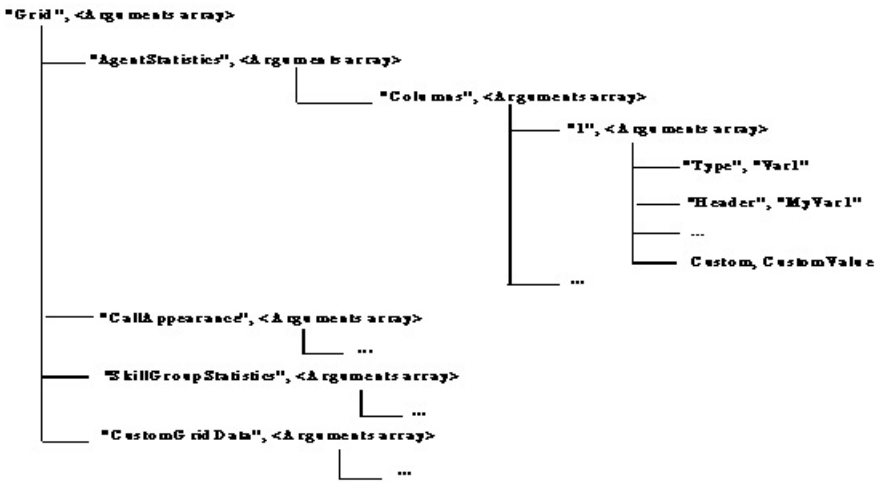
Keyword	Data Type
Type	string
Header	string
editable	boolean
maxchars	integer
Custom values[2]	custom

[2] Other registry values added to the <column_number> registry key are passed in the OnGlobalSettingsDownloadConf event. Subkeys added to the <column_number> registry key are not passed in this event.

You can add custom keys in the CTI OS Server registry's Grid subtree at the same level as the SkillGroupStatistics, AgentStatistics, and CallAppearance keys. The Grid Arguments array (see the following figure) within this event contain items corresponding to these custom keys. Any custom element that you add must follow the same hierarchy in the registry as that used by the existing top level elements.

The custom element hierarchy format is as follows:

Figure 2: Grid Arguments Array



- **IPCCSilentMonitor** – Arguments array that contains configuration information from the CTI OS server registry's IPCCSilentMonitor/ Name subtree.

  The IPCCSilentMonitor Arguments array contains a nested Arguments array with key=“settings.” This array contains the following key-value pairs:

Table 8: IPCCSilentMonitor: Key Values

Keyword	Value
MediaTerminationPort	integer

Keyword	Value
HeartBeatInterval	integer
TOS	boolean
MonitoringIPPort	integer
HeartbeatTimeout	integer
CCMBasedSilentMonitor	boolean

Configuration information for registry values added to the IPCCSilentMonitor/Settings registry key is passed to the client in the OnGlobalSettingsConf event. Configuration information for subkeys added to the Settings registry key is not passed in this event.

You can add custom keys to the CTI OS registry in the IPCCSilentMonitor subtree at the same level as the Settings key. The IPCCSilentMonitor Arguments array within this event contain items corresponding to these custom keys. Any custom element that you add must follow the same hierarchy in the registry as that used by the existing top level elements.

Two silent monitoring types are supported for Unified CCE:

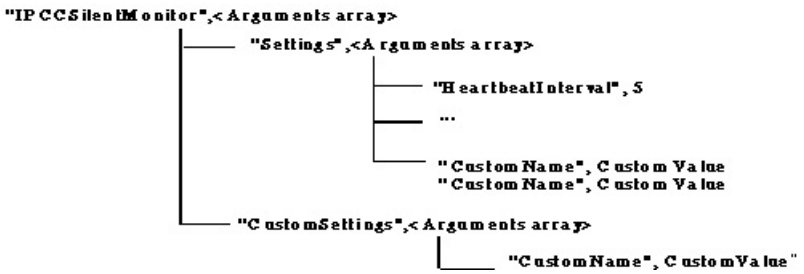
- CTI OS based
- CCM based

You configure the silent Monitor type used by CTI OS using the CCMBasedSilentMonitor registry key.

If CCMBasedSilentMonitor is present and set to true, CTI OS is using Call Manager's silent monitor implementation. When this is the case, supervisor applications must initiate silent monitor using the Agent.SuperviseCall() method. Agent applications do not need to do anything. If CCMBasedSilentMonitor is not present or set to 0, CTI OS implementation of silent monitor is in use. When this is the case, supervisor and agent applications must invoke silent monitor using the SilentMonitorManager object.

The format of the IPCCSilentMonitor Arguments array is shown in the following figure.

Figure 3: IPCCSilentMonitor Arguments Array



- **Login** – Arguments array that contains the information from the CTI OS server registry's Login subtree. This array contains a nested Arguments array with key="ConnectionProfiles" and with an Arguments array value for each connection profile. The keyword of each array is the name for the Connection Profile listed in the CTI OS server's registry. The value is another Arguments array.

  The following key-value pairs are contained in each connection profile Arguments array:

Table 9: Unified CCE Agent Statistics: Key Values

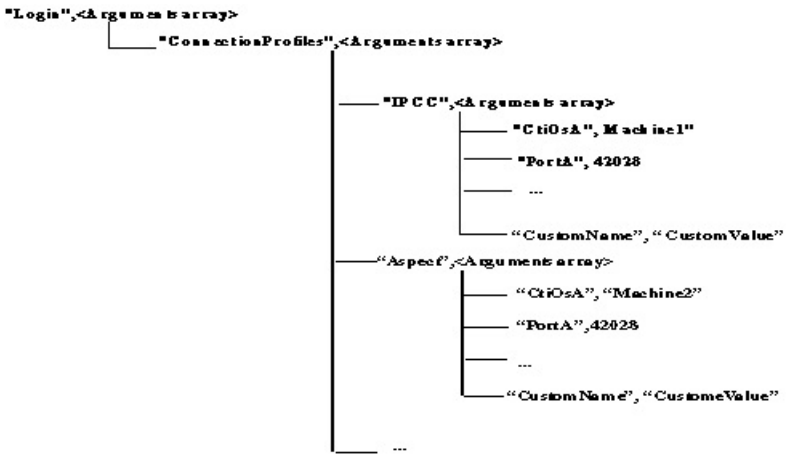
Keyword	Value
CtiOsA	string
CtiOsB	string
PortA	integer
PortB	integer
Heartbeat	integer
MaxHeartbeats	integer
AutoLogin	boolean
WarnIfAlreadyLoggedIn	boolean
ShowFieldBitMask	integer
RejectIfAlreadyLoggedIn	boolean
PeripheralID	integer
IPCCSilentMonitorEnabled	boolean
TOS	boolean
SwitchCapabilityBitMask	integer
WarnIfSilentMonitored	boolean
RasCallMode[3]	integer

[3] Applicable only to RAS enabled Unified CCE Profiles

Configuration information for keys or values that are added to the Login registry key in the CTI OS server's registry does not appear in the Login Arguments array.

The format of the Login Arguments array is shown in the following figure.

Figure 4: Login Arguments Array



SilentMonitorService Subkey

The <profile_name>/SilentMonitorService subkey contains parameters that clients use to connect to one of a set of silent monitor services. It contains the following keys:

Note The SilentMonitorService subkey is only applicable to CTI OS based silent monitor.

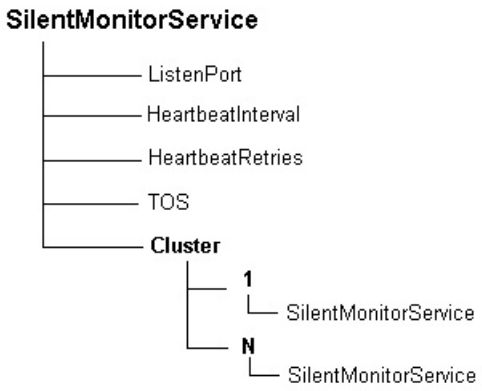
Table 10: SilentMonitorService Parameters

Keyword	Value	Description
ListenPort	integer	Port on which the silent monitor service is listening for incoming connections.
TOS	integer	QOS setting for the connection.
HeartbeatInterval	integer	Amount of time in milliseconds between heartbeats.
HeartbeatRetries	integer	Number of missed heartbeats before the connection is abandoned.

Keyword	Value	Description
Cluster		A key that contains a list of silent monitor services to which the CIL tries to connect. The CIL randomly chooses one of the services in this list. This key contains two subkeys: • 1 - index of the first silent monitor service • N - index of the Nth silent monitor service All subkeys contain the following keyword: • SilentMonitorService - host name or IP adress of the silent monitor service.

The following figure illustrates the hierarchy of the SilentMonitorService subkey.

Figure 5: SilentMonitorService Subkey Hierarchy



- **ScreenPreferences** – Arguments array that contains the information configured in the CTI OS server registry's ScreenPreferences/Name subtree. The ScreenPreferences array contains an element MainScreen, which is an Arguments array that contains the following key-value pairs:

Table 11: ScreenPreferences: Key Values

Keyword	Value
AgentStatisticsIntervalSec	integer
BringToFrontOnCall	boolean
FlashOnCall	boolean

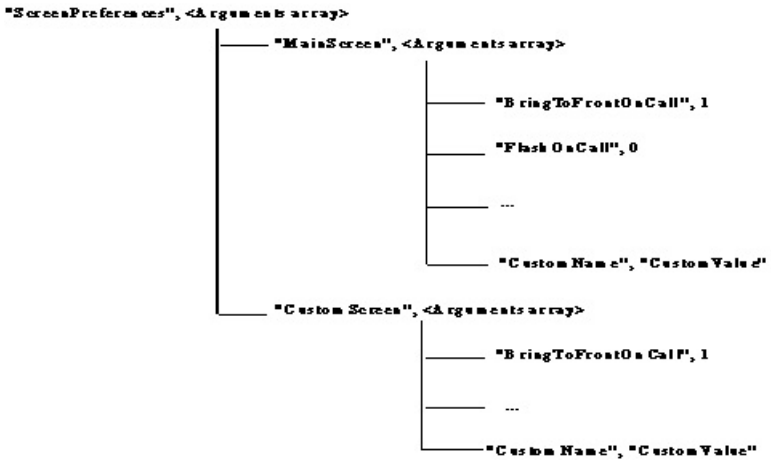
Keyword	Value
RecordingEnabled	boolean

You can add custom keys to the CTI OS registry in the ScreenPreferences subtree at the same level as the "MainScreen" key. The ScreenPreferences Arguments array within this event contains items corresponding to these custom keys. Any custom key that you add must follow the same hierarchy in the registry as that used by the existing top level keys.

Registry values added to the MainScreen registry key on the CTI OS server are passed to the client in the OnGlobalSettingsDownloadConf event. Subkeys added to the MainScreen registry key are not passed in this event.

The format of the ScreenPreferences Arguments array is shown in the following figure.

Figure 6: ScreenPreferences Arguments Array



- **SoundPreferences** – Arguments array that contains information configured on the CTI OS server in the SoundPreferences/Name subtree. This array includes a nested Arguments array that includes a setting for each sound, including .wav files to be played, and whether or not each one is mute. It can also include custom name/value pairs for a custom application.

  The SoundPreferences array contains the following key-value pairs:

  Table 12: SoundPreferences: Key Values

Keyword	Value	Subtree
DTMF*	Arguments array	SoundPreferences/Name/DTMF
DialTone*	Arguments array	SoundPreferences/Name/DialTone
OriginatingTone*	Arguments array	SoundPreferences/Name/OriginatingTone
RingInTone*	Arguments array	SoundPreferences/Name/RingInTone
All*	Arguments array	SoundPreferences/Name/All

  * Registry values added to this registry key in the CTI OS server registry are included in the Arguments array. Subkeys added to this registry key are not present.

The DTMF, DialTone, OriginatingTone, RingInTone, and All arrays each contain the keyword Mute, which has a boolean value. Custom registry values added to the DialTone DTMF, DialTone, OriginatingTone, RingInTone, and All registry keys are present in the array. Subkeys added to the these registry keys are not present in the array.

You can add custom keys in the SoundPreferences subtree at the same level as the All, DTMF, DialTone, OriginatingTone, and RingInTone keys. The SoundPreferences array contains items corresponding to these custom keys. Any custom element that you add must follow the same hierarchy in the registry as that used by the existing top level elements.

The format of the SoundPreferences Arguments array is shown in the following figure.

Figure 7: SoundPreferences Arguments Array

This configuration is stored in the Windows System Registry database and many of the values are set when the CTI OS Server Setup is run. You can set custom configuration at a later time by using the Windows Registry Editor.

OnHeartbeat

The OnHeartbeat event is generated when a heartbeat response is received from a CTI OS server. It returns the time of day.

Syntax

C++

```
void OnHeartbeat(Arguments& args)
```

COM

```
void Onheartbeat (IArguments * args)
```

VB

```
session_OnHeartbeat (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 13:

Keyword	Type	Description
EventTime	INT	Integer value with time of day expressed in milliseconds.

OnMissingHeartbeat

The OnMissingHeartbeat event is generated when an expected heartbeat is not received. It returns the number of consecutive heartbeats missed and time of day. When the number of heartbeats missed equals or exceeds the maximum number of heartbeats allowed (set in the MaxHeartbeats property), an OnConnectionFailure event is generated instead of an OnMissingHeartbeat event, and the CIL automatically attempts to reconnect to the CTI OS server, alternating between the CtiosA and CtiosB servers passed as parameters in the Connect method.

Syntax

C++

```
void OnMissingHeartbeat(Arguments& args)
```

COM

```
void OnMissingHeartbeat (IArguments * args)
```

VB

```
session_OnMissingHeartbeat (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 14: OnMissingHeartbeat Parameters

Keyword	Type	Description
EventTime	INT	Integer value with time of day expressed in milliseconds.
Consecutive MissedHeartbeats	INT	Integer value with the number of heartbeats missed.
HeartbeatInterval	INT	Integer value with the heartbeat interval, in milliseconds.

OnMonitorModeEstablished

The OnMonitorModeEstablished event is generated when Monitor Mode is established.

Syntax

C++

```
void OnMonitorModeEstablished(Arguments& args)
```

COM

```
void OnMonitorModeEstablished (IArguments * args)
```

VB

```
session_OnMonitorModeEstablished (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 15: OnMonitorModeEstablished Parameters

Keyword	Type	Description
CIL ConnectionID	STRING	ID of the client connection on the server.
StatusSystem	ARGUMENTS	Arguments array containing the following elements: • StatusCTIServer • StatusCtiServerDriver • StatusCentralController • StatusPeripherals (Arguments array with a peripheral ID for each key and a boolean true/false value indicating if that peripheral is online.)

OnSnapshotDeviceConf

The OnSnapshotDeviceConf confirmation message is fired to the client as part of a snapshot operation. For AgentMode clients, the OnSnapshotDeviceConf arrives at startup time, after the OnQueryAgentStateConf message. The OnSnapshotDeviceConf indicates the number of calls present at the device, and their UniqueObjectIDs.

Syntax

C++

```
void OnSnapshotDeviceConf (Arguments & args);
```

COM

```
HRESULT OnSnapshotDeviceConf ([in] IArguments * args);
```

VB

```
OnSnapshotDeviceConf (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 16: OnSnapshotDeviceConf Parameters

Keyword	Description	Type
UniqueObjectID	Unique ID of the device object on the server. There are no device objects in the CIL, so this keyword cannot be used to retrieve a device object at this point.	STRING
NumCalls	The number of active calls associated with this device, up to a maximum of 16.	SHORT
ValidCalls	An Arguments array containing the list of calls on the device. The Unique ObjectID of each call is a key in the Arguments object. The value is a boolean indicating if the call is valid. Calls not listed are not valid calls on the device.	ARGUMENTS

Remarks

The CIL uses this event to rectify the list of calls on a device when logging in after a failover, in case the status of calls on the device changes during the failure period. An example of such a scenario is an agent talking on a call on a hardphone and a CTI failure occurs. The agent hangs up the call before CTI is recovered. After CTI and the CIL recover, they use the snapshot to discover that the call it currently has is no longer on the device. CTI then fires an EndCall event to remove the call from its call list.

OnSnapshotSkillGroupList

Not supported.

OnTranslationRoute

The OnTranslationRoute event is a pre-call indication. The event indicates the pending arrival of a call, and provides early access to the call context information. From a call flow perspective, this event can be used to begin an application or database lookup for the call context data before the call actually arrives at the agent's teleset.

The contact is uniquely identified by the `ICMEnterpriseUniqueID`, which is a field based on the Unified ICM 64-bit unique key (RouterCallKeyDay and RouterCallKeyCallID). This event does not indicate the creation of a Call object on the CTI OS server—only that the contact is being tracked. This is sufficient to get and set data, which enables some powerful data-prefetching applications. When a OnCallBeginEvent follows for this same contact, the ICMEnterpriseUniqueID field is sent with the call data. At that point, a custom application can set the call data on the appropriate call object.

Syntax

C++

```
void OnTranslationRoute(Arguments& args)
```

COM

```
oid OnTranslationRoute(IArguments * args)
```

VB

```
session_OnTranslationRoute(ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 17: OnTranslationRoute Parameters

Keyword	Type	Description
ICMEnterpriseUniqueID	STRING	This string is a globally unique key for this contact, which corresponds to the Unified ICM 64 bit key. You can use this parameter to match this contact to a follow-on call event.
RouterCallKeyDay	INT	Together with the RouterCallKeyCallID field forms the unique 64-bit key for locating this call's records in the Unified ICM database. Only provided for Post-routed and Translation-routed calls.
RouterCalKeyCallID	INT	The call key created by the Unified ICM . The Unified ICM resets this counter at midnight.
RouterCallKey SequenceNumber	INT	Together with RouterCallKeyDay and RouterCallKeyCallID fields forms the TaskID.
NumNamedVariables	SHORT	Number of Named variables.
NumNamedArrays	SHORT	Number of Named Arrays.
ANI	STRING	The calling line ID of the caller.
UserToUserInfo	STRING	The ISDN user-to-user information element.
DNIS	STRING	The DNIS number to which this call will arrive on the ACD/PBX.
DialedNumber	STRING	The number dialed.

Keyword	Type	Description
CallerEnteredDigits	STRING	The digits entered by the caller in response to VRU prompting.
CallVariable1	STRING	Call-related variable data.
...	STRING	...
CallVariable10	STRING	Call-related variable data.
ECC	ARGUMENTS	A nested Arguments structure of key-value pairs for all of the ECC data arriving with this call.

ICallEvents Interface

The Call object fires events on the ICallEvents interface. The following events are published to subscribers of the ICallEvents interface.

Note Many of the parameters that CTI OS receives from the CTI Server are inconsequential to most customer applications. The most important parameters for doing a screenpop are included with the events described in this section. The more inconsequential parameters are suppressed at the CTI OS Server, to minimize network traffic to the clients. However, you can enable the complete set of available event arguments by setting the following registry setting:

Note [HKLM\Cisco Systems\CTIOS\Server\CallObject\MinimizeEventArgs = 0].

OnAgentPrecallEvent

Note The OnAgentPrecallEvent event is applicable to Unified CCE only. The equivalent on all other TDM events is TranslationRouteEvent.

The OnAgentPrecallEvent event is a pre-call indication that indicates the pending arrival of a call and provides early access to the call context information. From a call flow perspective, you can use this event to begin an application or database lookup for the call context data before the call actually arrives at the agent's teleset.

The contact is uniquely identified by the ICMEnterpriseUniqueID, which is a field based on the Unified ICM 64-bit unique key (RouterCallKeyDay and RouterCallKeyCallID). This event does not indicate the creation of a Call object on the CTI OS server—only that the contact is being tracked. This is sufficient to get and set data, which enables some powerful data-prefetching applications. When an OnCallBeginEvent follows for this same contact, the ICMEnterpriseUniqueID field is sent along with the call data. At that point, a custom application can set the call data on the appropriate call object.

Syntax

C++

```
void OnAgentPrecallEvent(Arguments& args)
```

COM

```
void OnAgentPrecallEvent (IArguments * args)
```

VB

```
session_OnAgentPrecallEvent (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 18: OnAgentPrecallEvent Parameters

Keyword	Type	Description
ICMEnterpriseUniqueID	STRING	This string is a globally unique key for this contact, which corresponds to the Unified ICM 64 bit key. You can use this parameter to match this contact to a follow-on call event.
RouterCallKeyDay	INT	Together with the RouterCallKeyCallID field forms the unique 64-bit key for locating this call's records in the Unified ICM database. Only provided for Post-routed and Translation-routed calls.
RouterCalKeyCallID	INT	The call key created by the Unified ICM. The Unified ICM resets this counter at midnight.
AgentInstrument	STRING	The agent instrument that the call is routed to.
NumNamedVariables	SHORT	Number of Named variables.
NumNamedArrays	SHORT	Number of Named Arrays.
ServiceNumber	INT	The service that the call is attributed to, as known to the peripheral.
ServiceID	INT	The Unified ICM ServiceID of the service that the call is attributed to.

Keyword	Type	Description
SkillGroupNumber	INT	An optional, user-defined number of the agent SkillGroup the call is attributed to, as known to the peripheral.
SkillGroupID	INT	The system-assigned identifier of the agent SkillGroup the call is attributed to.
SkillGroupPriority	SHORT	The priority of the skill group, or 0 when skill group priority is not applicable or not available.
ANI	STRING	The calling line ID of the caller.
UserToUserInfo	STRING	The ISDN user-to-user information element.
DNIS	STRING	The DNIS number to which this call will arrive on the ACD/PBX.
DialedNumber	STRING	The number dialed.
CallerEnteredDigits	STRING	The digits entered by the caller in response to VRU prompting.
CallVariable1	STRING	Call-related variable data.
...	STRING	...
CallVariable10	STRING	Call-related variable data.
ECC	ARGUMENTS	A nested Arguments structure of key-value pairs for all of the ECC data arriving with this call.
CallTypeIDTag	INT	Specifies CallType of the call and indicates that the agent is reserved via LegacyPreCall.
PreCallInvokeIDTag	INT	Specifies the invoking of the LegacyPreCall.

OnAgentPrecallAbortEvent

Note The OnAgentPrecallAbortEvent event is applicable to Unified CCE only.

The OnAgentPrecallAbortEvent event is received only if a previously indicated routing (OnAgentPrecallEvent) decision is reversed. The contact is uniquely identified by the ICMEnterpriseUniqueID, which is a field based on the Unified ICM 64-bit unique key (RouterCallKeyDay and RouterCallKeyCallID). Upon receipt of an OnAgentPrecallAbortEvent, any data pre-fetch work that was started on an OnAgentPrecallEvent should be cleaned up.

Syntax

C++

```
void OnAgentPrecallAbortEvent(Arguments& args)
```

COM

```
void OnAgentPrecallAbortEvent (IArguments * args)
```

VB

```
session_OnAgentPrecallAbortEvent (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 19: OnAgentPrecallAbortEvent Parameters

Keyword	Type	Description
ICMEnterpriseUniqueID	STRING	This string is a globally unique key for this contact, which corresponds to the Unified ICM 64 bit key. You can use This parameter to match this contact to a follow-on call event.
RouterCallKeyDay	INT	Together with the RouterCallKey CallID field forms the unique 64-bit key for locating this call's records in the Unified ICM database. Only provided for Post-routed and Translation- routed calls.
RouterCalKeyCallID	INT	The call key created by the Unified ICM. The Unified ICM resets this counter at midnight.
AgentInstrument	STRING	The agent instrument that the call will be routed to.

OnAlternateCallConf

The OnAlternateCallConf event is fired to the client to indicate that an Alternate request was received by the CTI Server

Syntax

C++

```
void OnAlternateCallConf (Arguments & args);
```

COM

```
HRESULT OnAlternateCallConf ([in] IArguments * args);
```

VB

```
Session_ OnAlternateCallConf (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following field.

Table 20: OnAlternateCallConf Parameters

Keyword	Type	Description
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.

OnAnswerCallConf

The OnAnswerCallConf event is fired to the client to indicate that an Answer request was received by the CTI Server.

Syntax

C++

```
void OnAnswerCallConf (Arguments & args);
```

COM

```
HRESULT OnAnswerCallConf ([in] IArguments * args);
```

VB

```
Session_ OnAnswerCallConf (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following field.

Table 21: OnAnswerCallConf Parameters

Keyword	Type	Description
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.

OnCallBegin

The OnCallBegin event is generated at the first association between a call and the CTI Client. The event passes the call identifier and the initial call context data. The ConnectionCallID identifies the call. This message always precedes any other event messages for that call.

Subsequent changes to the call context data (if any) are signaled by an OnCallDataUpdate event containing the changed call data.

Note There can be multiple calls with the same ConnectionCallID value.

Syntax

C++

```
void OnCallBegin(Arguments& args)
```

COM

```
void OnCallBegin (IArguments * args)
```

VB

```
session_OnCallBegin (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 22: OnCallBegin Parameters

Keyword	Type	Description
PeripheralID	INT	The Unified ICM PeripheralID of the ACD where the call activity occurred.
PeripheralType	SHORT	The type of the peripheral.
CallType	SHORT	The general classification of the call type.
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.
RouterCallKeyDay	INT	Together with the RouterCallKeyCallID field forms the unique 64-bit key for locating this call's records in the Unified ICM database. Only provided for Post-routed and Translation-routed calls.

Keyword	Type	Description
RouterCalKeyCallID	INT	The call key created by the Unified ICM . The Unified ICM resets this counter at midnight.
RouterCallKey SequenceNumber	INT	Together with RouterCallKeyDay and RouterCallKeyCallID fields forms the TaskID.
ConnectionCallID	UINT	The Call ID value assigned to this call by the peripheral or the Unified ICM .
ANI (optional)	STRING	The calling line ID of the caller.
DNIS (optional)	STRING	The DNIS provided with the call.
UserToUserInfo (Optional)	STRING	The ISDN user-to-user information element. unspecified, up to 131 bytes.
DialedNumber (Optional)	STRING	The number dialed.
CallerEnteredDigits (Optional)	STRING	The digits entered by the caller in response to VRU prompting.
ServiceNumber (Optional)	INT	The service that the call is attributed to, as known to the peripheral. May contain the special value NULL_SERVICE when not applicable or not available.
ServiceID (Optional)	INT	The Unified ICM ServiceID of the service that the call is attributed to. May contain the special value NULL_SERVICE when not applicable or not available.
SkillGroupNumber (Optional)	INT	An optional, user-defined number of the agent SkillGroup the call is attributed to, as known to the peripheral. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
SkillGroupID (Optional)	INT	The system-generated identifier of the agent SkillGroup the call is attributed to. May contain the special value NULL_SKILL_GROUP when not applicable or not available.

Keyword	Type	Description
SkillGroupPriority (Optional)	SHORT	The priority of the skill group, or 0 when skill group priority is not applicable or not available.
CallWrapupData (Optional)	STRING	Call-related wrap-up data.
CallVariable1 (Optional)	STRING	Call-related variable data.
...	STRING	...
CallVariable10 (Optional)	STRING	Call-related variable data.
CallStatus (optional)	SHORT	The current status of the call.
ECC (optional)	ARGUMENTS	Arguments array that contains all of the Expanded Call Context variables in use; for example: user.ArrayVariable[0]user.ArrayVariable[1] ...user.ArrayVariable[n]user.ScalarVariable
CTIClients (optional)	ARGUMENTS	Arguments array that contains the information about the number of clients that are using the Call object; for example: CTIClient[1] CTIClientSignatureCTIClientTimestamp
ICMEnterprise UniqueID (optional)	STRING	Required only when the call is pre-routed.

OnCallCleared

An OnCallCleared event is generated when the voice portion of all parties on a call is terminated, normally when the last device disconnects from a call. With this event the connection status becomes LCS_NULL.

Note If the CallCleared event is received after having received a CallFailed event, the event does not include a CallStatus because it is important to preserve the fact that the call failed (maintaining the CallStatus of LSC_Fail). Because of this exception, the CallStatus of the CallCleared event is optional.

Syntax

C++

```
void OnCallDelivered(Arguments& args)
```

COM

```
void OnCallCleared (IArguments * args)
```

VB

```
session_OnCallCleared (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 23: OnCallCleared Parameters

Keyword	Type	Description
EnablementMask	INT	Contains the bit-mask that specifies what buttons can be enabled or disabled when this call is the current call.
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.
CallStatus	SHORT	The current status of the call.
ICMEnterprise UniqueID (Optional)	STRING	Required only when the call is pre-routed.

OnCallConnectionCleared

An OnCallConnectionCleared event is generated when a party drops from a call. With this event the connection status becomes LCS_NULL.

Note If the CallConnectionCleared event is received after having received a CallFailed event, the event does not include a CallStatus because it is important to preserve the fact that the call failed (maintaining the CallStatus of LSC_Fail). Because of this exception, the CallStatus of the CallConnectionCleared event is optional.

Syntax

C++

```
void OnCallConnectionCleared(Arguments& args)
```

COM

```
void OnCallConnectionCleared (IArguments * args)
```

VB

```
session_OnCallConectionCleared (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 24: OnCallConnectionCleared Parameters

Keyword	Type	Description
EnablementMask	INT	Contains the bit-mask that specifies what buttons can be enabled or disabled when this call is the current call.
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.
CallStatus	SHORT	The current status of the call.
ICMEnterprise UniqueID (Optional)	STRING	Required only when the call is pre-routed.

OnCallConferenced

The joining of calls into a conference call or the adding of a new call joining a conference can generate an OnCallConferenced event. With this event, the connections at the controller's device merge to become one connection with a status of LCS_CONNECT, and the status of the connections at the original caller's device and at the consulted device remain unchanged.

Syntax

C++

```
void OnCallConferenced(Arguments& args)
```

COM

```
void OnCallConferenced (IArguments * args)
```

VB

```
session_OnCallConferenced (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 25: OnCallConferenced Parameters

Keyword	Type	Description
PeripheralID	INT	The Unified ICM PeripheralID of the ACD where the call activity occurred.
PeripheralType	SHORT	The type of the peripheral.
CallType	SHORT	The general classification of the call type.

Keyword	Type	Description
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.
RouterCallKeyDay	INT	Together with the RouterCallKeyCallID field forms the unique 64-bit key for locating this call's records in the Unified ICM database. Only provided for Post-routed and Translation-routed calls.
RouterCalKeyCallID	INT	The call key created by the Unified ICM. The Unified ICM resets this counter at midnight.
ConnectionCallID	UINT	The Call ID value assigned to this call by the peripheral or the Unified ICM.
ANI (optional)	STRING	The calling line ID of the caller.
DNIS (optional)	STRING	The DNIS provided with the call.
UserToUserInfo (Optional)	STRING	The ISDN user-to-user information element. unspecified, up to 131 bytes.
DialedNumber (Optional)	STRING	The number dialed.
CallerEnteredDigits (Optional)	STRING	The digits entered by the caller in response to VRU prompting.
ServiceNumber (Optional)	INT	The service that the call is attributed to, as known to the peripheral. May contain the special value NULL_SERVICE when not applicable or not available.
ServiceID (Optional)	INT	The Unified ICM ServiceID of the service that the call is attributed to. May contain the special value NULL_SERVICE when not applicable or not available.

Keyword	Type	Description
SkillGroupNumber (Optional)	INT	An optional, user-defined number of the agent SkillGroup the call is attributed to, as known to the peripheral. as known to the peripheral. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
SkillGroupID (Optional)	INT	The system-assigned identifier of the agent SkillGroup the call is attributed to. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
SkillGroupPriority (Optional)	SHORT	The priority of the skill group, or 0 when skill group priority is not applicable or not available.
CallWrapupData (Optional)	STRING	Call-related wrap-up data.
CallVariable1 (Optional)	STRING	Call-related variable data.
...	STRING	...
CallVariable10 (Optional)	STRING	Call-related variable data.
CallStatus (optional)	SHORT	The current status of the call.
ECC (optional)	ARGUMENTS	Arguments array that contains all of the Expanded Call Context variables in use; for example: user.ArrayVariable[0]user.ArrayVariable[1] ...user.ArrayVariable[n]user.ScalarVariable
CTIClients (Optional)	ARGUMENTS	Arguments array that contains the information about the number of clients that are using the Call object; for example: CTIClient[1] CTIClientSignatureCTIClientTimestamp
ICMEnterpriseUnique ID (Optional)	STRING	Required only when the call is pre-routed.

OnCallDataUpdate

Changes to the call context data generate an OnCallDataUpdate event. Only the changed items are in the event argument array. The initial call context is provided in the OnCallBegin event.

Syntax

C++

```
void OnCallDataUpdate(Arguments& args)
```

COM

```
void OnCallDataUpdate (IArguments * args)
```

VB

```
session_OnCallDataUpdate (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 26: OnCallUpdate Parameters

Keyword	Type	Description
PeripheralID	INT	The Unified ICM PeripheralID of the ACD where the call activity occurred.
PeripheralType	SHORT	The type of the peripheral.
CallType	SHORT	The general classification of the call type.
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.
RouterCallKeyDay	INT	Together with the RouterCallKeyCallID field forms the unique 64-bit key for locating this call's records in the Unified ICM database. Only provided for Post-routed and Translation-routed calls.
RouterCalKeyCallID	INT	The call key created by the Unified ICM. The Unified ICM resets this counter at midnight.
RouterCallKey SequenceNumber	INT	Together with RouterCallKeyDay and RouterCallKeyCallID fields forms the TaskID.

Keyword	Type	Description
ConnectionCallID	UINT	The Call ID value assigned to this call by the peripheral or the Unified ICM.
ANI (optional)	STRING	The calling line ID of the caller.
DNIS (optional)	STRING	The DNIS provided with the call.
UserToUserInfo (Optional)	STRING	The ISDN user-to-user information element. unspecified, up to 131 bytes.
DialedNumber (Optional)	STRING	The number dialed.
CallerEnteredDigits (Optional)	STRING	The digits entered by the caller in response to VRU prompting.
ServiceNumber (Optional)	INT	The service that the call is attributed to, as known to the peripheral. May contain the special value NULL_SERVICE when not applicable or not available.
ServiceID (Optional)	INT	The Unified ICM ServiceID of the service that the call is attributed to. May contain the special value NULL_SERVICE when not applicable or not available.
SkillGroupNumber (Optional)	INT	An optional, user-defined number of the agent SkillGroup the call is attributed to, as known to the peripheral. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
SkillGroupID (Optional)	INT	The system-assigned identifier of the agent SkillGroup the call is attributed to. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
SkillGroupPriority (Optional)	SHORT	The priority of the skill group, or 0 when skill group priority is not applicable or not available.
CallWrapupData (Optional)	STRING	Call-related wrap-up data.
CallVariable1 (Optional)	STRING	Call-related variable data.

Keyword	Type	Description
...	STRING	...
CallVariable10 (Optional)	STRING	Call-related variable data.
CallStatus (optional)	SHORT	The current status of the call.
ECC (optional)	ARGUMENTS	Arguments array that contains all of the Expanded Call Context variables in use; for example: user.ArrayVariable[0]user.ArrayVariable[1]...user.ArrayVariable[n]user.ScalarVariable
CTIClients (Optional)	ARGUMENTS	Arguments array that contains the information about the number of clients that are using the Call object; for example: CTIClient[1] CTIClientSignatureCTIClientTimestamp
ICMEnterprise UniqueID (Optional)	STRING	Required only when the call is pre-routed.

OnCallDelivered

The OnCallDelivered event may be generated when the call arrives at the agent's teleset. Both parties (call connections) receive this event. With this event, the called party's connection status becomes LCS_ALERTING but the calling party's connection status remains LCS_INITIATE.

Note With certain switches, when a call is made outside of the ACD, this event may not be received. For more information, see OnCallReachedNetwork.

Syntax

C++

```
void OnCallDelivered(Arguments& args)
```

COM

```
void OnCallDelivered (IArguments * args)
```

VB

```
session_OnCallDelivered (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 27: OnCallDelivered Parameters

Keyword	Type	Description
ServiceNumber	INT	The service that the call is attributed to, as known to the peripheral. May contain the special value NULL_SERVICE when not applicable or not available.
ServiceID	INT	The Unified ICM ServiceID of the service that the call is attributed to. May contain the special value NULL_SERVICE when not applicable or not available.
SkillGroupNumber (Optional)	INT	An optional, user-defined number of the agent SkillGroup the call is attributed to, as known to the peripheral. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
SkillGroupID (Optional)	INT	The system-assigned identifier of the agent SkillGroup the call is attributed to. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
SkillGroupPriority (Optional)	SHORT	The priority of the skill group, or 0 when skill group priority is not applicable or not available.
LineType	SHORT	Indicates the type of the teleset line.
EnablementMask	INT	Contains the bit-mask that specifies what buttons can be enabled or disabled when this call is the current call. See Table 1.
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.
CallStatus	SHORT	The current status of the call.
ICMEnterpriseUniqueID (Optional)	STRING	Required only when the call is pre-routed.
TrunkNumber (optional)	INT	The number representing a trunk.
TrunkGroup Number (optional)	INT	The number representing a trunk group.

OnCallDequeuedEvent

The explicit removal of a call from the ACD queue can generate a OnCallDequeuedEvent message to the CTI Client.

Syntax

C++

```
void OnCallDequeuedEvent(Arguments& args)
```

COM

```
void OnCallDequeuedEvent (IArguments * args)
```

VB

```
session_OnCallDequeuedEvent (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 28: OnCallDequeuedEvent Parameters

Keyword	Type	Description
Connection DeviceID	INT	The identifier of the connection between the call and the device.
ConnectionDevice IDType	SHORT	Indicates the type of the connection identifier supplied in the ConnectionDeviceID.
LocalConnection State	SHORT	The state of the local end of the connection.
EventCause	SHORT	Indicates a reason or explanation for the occurrence of the event.
LineHandle	SHORT	Identifies the teleset line being used.
LineType	SHORT	Indicates the type of the teleset line.
ServiceID	INT	The Unified ICM ServiceID of the service that the call is attributed to.
ServiceNumber	INT	The service that the call is attributed to, as known to the peripheral.
NumQueued	SHORT	The number of calls in the queue for this service.

Keyword	Type	Description
NumSkillGroups	SHORT	The number of Skill Groups that the call has been removed from, up to a maximum of 99.

OnCallDiverted

The removal of a call from one delivery target and forwarded to a different target can generate an OnCallDiverted event.

Syntax

C++

```
void OnCallDiverted(Arguments& args)
```

COM

```
void OnCallDiverted (IArguments * args)
```

VB

```
session_OnCallDiverted (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 29: OnCallDiverted Parameters

Keyword	Type	Description
UniqueObjectID	STRING	Unique reference generated for a call at client.
PeripheralID	INT	The Unified ICM PeripheralID of the ACD where the call activity occurred.
PeripheralType	SHORT	The type of the peripheral.
ConnectionDevice IDType	SHORT	Indicates the type of ConnectionDeviceID value.
Connection DeviceID	INT	The device identifier of the connection between the call and the device.
ConnectionCallID	UINT	The Call ID value assigned to this call by the peripheral or the Unified ICM.

Keyword	Type	Description
ServiceNumber	INT	The service that the call is attributed to, as known to the peripheral. May contain the special value NULL_SERVICE when not applicable or not available.
ServiceID	INT	The Unified ICM ServiceID of the service that the call is attributed to. May contain the special value NULL_SERVICE when not applicable or not available.
DivertingDevice Type	SHORT	Indicates the type of the device identifier supplied in the DivertingDeviceID field.
CalledDeviceType	SHORT	Indicates the type of the device identifier supplied in the CalledDeviceID field.
LocalConnection State	SHORT	The state of the local end of the connection.
EventCause	SHORT	Indicates a reason or explanation for the occurrence of the event.
DivertingDeviceID (Optional)	STRING	The device identifier of the device from which the call was diverted.
CalledDeviceID (Optional)	STRING	The device identifier of the device to which the call was diverted.

OnCallEnd

The OnCallEnd event is generated when the association between a call and the CTI Client is dissolved. The OnCallEnd event is the last event received for a Call.

Syntax

C++

```
void OnCallEnd(Arguments& args)
```

COM

```
void OnCallEnd (IArguments * args)
```

VB

```
session_OnCallEnd (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 30: OnCallEnd Parameters

Keyword	Type	Description
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.
CallStatus (optional)	SHORT	The current status of the call.
ICMEnterprise UniqueID (optional)	STRING	Required only when the call is pre-routed.

OnCallEstablished

The OnCallEstablished event is generated when the call is answered at the agent's teleset. Both parties (call connections) receive this event when the call is answered. With this event, the call status of both parties becomes LCS_CONNECT.

Note With certain switches, when a call is made outside of the ACD, this event may not be received. See OnCallReachedNetwork for more detail.

Syntax

C++

```
void OnCallEstablished(Arguments& args)
```

COM

```
void OnCallEstablished (IArguments * args)
```

VB

```
session_OnCallEstablished (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 31: OnCallEstablished Parameters

Keyword	Type	Description
ServiceNumber	INT	The service that the call is attributed to, as known to the peripheral. May contain the special value NULL_ SERVICE when not applicable or not available.
ServiceID	INT	The Unified ICM ServiceID of the service that the call is attributed to. May contain the special value NULL_SERVICE when not applicable or not available.
SkillGroupNumber (Optional)	INT	An optional, user-defined number of the agent SkillGroup the call is attributed to, as known to the peripheral. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
SkillGroupID (Optional)	INT	The system-assigned identifier of the agent SkillGroup the call is attributed to. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
SkillGroupPriority (Optional)	SHORT	The priority of the skill group, or 0 when skill group priority is not applicable or not available.
LineType	SHORT	Indicates the type of the teleset line.
EnablementMask	INT	Contains the bit-mask that specifies what buttons can be enabled or disabled when this call is the current call. See Table 1.
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.
CallStatus	SHORT	The current status of the call.
ICMEnterpriseUniqueID (Optional)	STRING	Required only when the call is pre-routed.
TrunkNumber (optional)	INT	The number representing a trunk.
TrunkGroup Number (optional)	INT	The number representing a trunk group.

OnCallFailed

The OnCallFailed event is generated when a call is not completed. With this event the connection status becomes LCS_FAIL. This usually happens as a result of a MakeCall or a MakeConsultCall request, but can occur at any other point in the call's lifetime if the call fails on an ACD. In this case, you should perform any required cleanup prior to arrival of an EndCall event.

Note The events (CallConnectionCleared and CallCleared) received after the CallFailed event does not include a CallStatus because, until the call has ended, it is important to preserve the fact that this is a failed call.

Syntax

C++

```
void OnCallFailed(Arguments& args)
```

COM

```
void OnCallFailed (IArguments * args)
```

VB

```
session_OnCallFailed (ByVal args As CtiosCLIENTLib.IArguments
```

Parameters

args

Arguments array containing the following fields.

Table 32: OnCallFailed Parameters

Keyword	Type	Description
EnablementMask	INT	Contains the bit mask that specifies what buttons can be enabled or disabled when this call is the current call.
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.
CallStatus	SHORT	The current status of the call.

OnCallHeld

Placing a call on hold at the agent's teleset can generate an OnCallHeld event. With this event the connection status becomes LCS_HELD.

Syntax

C++

```
void OnCallHeld(Arguments& args)
```

COM

```
void OnCallHeld (IArguments * args)
```

VB

```
session_OnCallHeld (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 33: OnCallHeld Parameters

Keyword	Type	Description
EnablementMask	INT	Contains the bit-mask that specifies what buttons can be enabled or disabled when this call is the current call.
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.
CallStatus	SHORT	The current status of the call.
ICMEnterpriseUniqueID (Optional)	STRING	Required only when the call is pre-routed.

OnCallOriginated

The initiation of a call from the peripheral can generate an OnCallOriginated event. Only the connection making the call receives this event. With this event the connection status becomes LCS_INITIATE.

Syntax

C++

```
void OnCallOriginated(Arguments& args)
```

COM

```
void OnCallOriginated (IArguments * args)
```

VB

```
session_OnCallOriginated (ByVal args As CtiosCLIENTLib.IArguments
```

Parameters

args

Arguments array containing the following fields.

Table 34: OnCallOriginated Parameters

Keyword	Type	Description
ServiceNumber	INT	The service that the call is attributed to, as known to the peripheral. May contain the special value NULL_SERVICE when not applicable or not available.
ServiceID	INT	The Unified ICM ServiceID of the service that the call is attributed to. May contain the special value NULL_SERVICE when not applicable or not available.
SkillGroupNumber (Optional)	INT	The user-defined number of the agent SkillGroup the call is attributed to, as known to the peripheral. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
SkillGroupID (Optional)	INT	The system-assigned identifier of the agent SkillGroup the call is attributed to. May contain the special value NULL_SKILL_ GROUP when not applicable or not available.
SkillGroupPriority (Optional)	SHORT	The priority of the skill group, or 0 when skill group priority is not applicable or not available.
LineType	SHORT	Indicates the type of the teleset line.
EnablementMask	INT	Contains the bit-mask that specifies what buttons can be enabled or disabled when this call is the current call.
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.
CallStatus	SHORT	The current status of the call.

OnCallQueuedEvent

The placing of a call in a queue pending the availability of some resource can generate an OnCallQueuedEvent message to the CTI Client. Clients with Client Events Service can receive this message when an outbound

call is queued waiting for a trunk or other resource. Clients with All Events Service can also receive this message when inbound calls are queued.

Syntax

C++

```
void OnCallQueuedEvent(Arguments& args)
```

COM

```
void OnCallQueuedEvent (IArguments * args)
```

VB

```
session_OnCallQueuedEvent (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 35: OnCallQueuedEvent Parameters

Keyword	Type	Description
Connection DeviceID	INT	The identifier of the connection between the call and the device.
ConnectionDevice IDType	SHORT	Indicates the type of the connection identifier supplied in the ConnectionDeviceID.
QueuedDeviceID	STRING	The device identifier of the queuing device.
QueuedDeviceIDType	SHORT	Indicates the type of the device identifier supplied in the QueuedDeviceID.
CallingDeviceID	STRING	The device identifier of the calling device.
CallingDeviceIDType	SHORT	Indicates the type of the device identifier supplied in the CalledDeviceID.
CalledDeviceID	STRING	The device identifier of the called device.
CalledDeviceIDType	SHORT	Indicates the type of the device identifier supplied in the CalledDeviceID.
LastRedirectedDeviceID	STRING	The device identifier of the redirecting device.

Keyword	Type	Description
LastRedirected DeviceIDType	SHORT	Indicates the type of the device identifier supplied in the LastRedirectDeviceID.
LocalConnection State	SHORT	The state of the local end of the connection.
EventCause	SHORT	Indicates a reason or explanation for the occurrence of the event.
LineHandle	SHORT	Identifies the teleset line being used.
LineType	SHORT	Indicates the type of the teleset line.
ServiceID	INT	The Unified ICM ServiceID of the service that the call is attributed to.
ServiceNumber	INT	The service that the call is attributed to, as known to the peripheral.
NumQueued	SHORT	The number of calls in the queue for this service.
NumSkillGroups	SHORT	The number of Skill Group queues that the call has queued to, up to a maximum of 50.

OnCallReachedNetworkEvent

The connection of an outbound call to another network can generate an OnCallReachedNetworkEvent. With some switches outside the ACD, this can be the last event the outbound connection receives. For these switches, you cannot assume that when the called party receives and answers the call that the OnCallDelivered and OnCallEstablished events is received.

Syntax

C++

```
void OnCallReachedNetworkEvent(Arguments& args)
```

COM

```
void OnCallReachedNetworkEvent (IArguments * args)
```

VB

```
session_OnCallReachedNetworkEvent (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 36: OnCallReachedNetworkEvent Parameters

Keyword	Type	Description
Connection DeviceID	STRING	The identifier of the connection between the call and the device.
ConnectionDevice IDType	SHORT	Indicates the type of the connection identifier supplied in the ConnectionDeviceID.
TrunkUsedDeviceID	STRING	The device identifier of the selected trunk.
TrunkUsedDeviceIDType	SHORT	Indicates the type of the device identifier supplied in the TrunkUsedDeviceID.
CalledDeviceID	STRING	The device identifier of the called device.
CalledDeviceIDType	SHORT	Indicates the type of the device identifier supplied in the CalledDeviceID.
LocalConnection State	SHORT	The state of the local end of the connection.
EventCause	SHORT	Indicates a reason or explanation for the occurrence of the event.
LineHandle	SHORT	Identifies the teleset line being used.
LineType	SHORT	Indicates the type of the teleset line.
TrunkNumber (optional)	INT	The number representing a trunk.
TrunkGroup Number (optional)	INT	The number representing a trunk group.

OnCallRetrieved

Resuming a call previously placed on hold at the agent's teleset can generate an OnCallRetrieved event. With this event the connection status becomes LCS_CONNECT.

Syntax

C++

```
void OnCallRetrieved(Arguments& args)
```

COM

```
void OnCallRetrieved (IArguments * args)
```

VB

```
session_OnCallRetrieved (ByVal args As CtiosCLIENTLib.IArguments
```

Parameters

args

Arguments array containing the following fields.

Table 37: OnCallRetrieved Parameters

Keyword	Type	Description
EnablementMask	INT	Contains the bit-mask that specifies what buttons can be enabled or disabled when this call is the current call.
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.
CallStatus	SHORT	The current status of the call.

OnCallServiceInitiatedEvent

The initiation of telecommunications service (“dial tone”) at the agent's teleset can generate an OnCallServiceInitiatedEvent to the CTI Client. However, when the call is made through the software, there is no way to detect the equivalent of the phone off hook. Therefore, after a call is made this event is received in sequence along with the OnCallOriginated and OnCallDelivered events. With this event the connection status becomes LCS_INITIATE.

Syntax

C++

```
void OnCallServiceInitiatedEvent(Arguments& args)
```

COM

```
void OnCallServiceInitiatedEvent (IArguments * args)
```

VB

```
session_OnCallServiceInitiatedEvent (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 38: OnCallServiceInitiatedEvent Parameters

Keyword	Type	Description
ServiceNumber	INT	The service that the call is attributed to, as known to the peripheral. May contain the special value NULL_SERVICE when not applicable or not available.
ServiceID	INT	The Unified ICM ServiceID of the service that the call is attributed to. May contain the special value NULL_SERVICE when not applicable or not available.
SkillGroupNumber (Optional)	INT	The optional, user-defined number of the agent SkillGroup the call is attributed to, as known to the peripheral. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
SkillGroupID (Optional)	INT	The system-assigned identifier of the agent SkillGroup the call is attributed to. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
SkillGroupPriority (Optional)	SHORT	The priority of the skill group, or 0 when skill group priority is not applicable or not available.
LineType	SHORT	Indicates the type of the teleset line.
EnablementMask	INT	Contains the bit-mask that specifies what buttons can be enabled or disabled when this call is the current call.
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.
CallStatus	SHORT	The current status of the call.

OnCallStartRecordingConf

The OnCallStartRecordingConf event is fired to the client to indicate that the CTI server received a StartRecord request.

Syntax

C++

```
void OnCallStartRecordingConf (Arguments & args);
```

COM

```
HRESULT OnCallStartRecordingConf ([in] IArguments * args);
```

VB

```
Session_ OnCallStartRecordingConf (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following field.

Table 39: OnCallStartRecordingConf Parameters

Keyword	Type	Description
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.

OnCallStopRecordingConf

The OnCallStopRecordingConf event is fired to the client to indicate that a the CTI server received a StopRecord request.

Syntax

C++

```
void OnCallStopRecordingConf (Arguments & args);
```

COM

```
HRESULT OnCallStopRecordingConf ([in] IArguments * args);
```

VB

```
Session_ OnCallStopRecordingConf (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following field.

Table 40: OnCallStopRecordingConf Parameters

Keyword	Type	Description
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.

OnCallTransferred

The transfer of a call to another destination can generate an OnCallTransferred event. With this event the two connections at the controller's device end and the status of the connections at the original caller's device and the consulted device are unchanged.

Syntax

C++

```
void OnCallTransferred(Arguments& args)
```

COM

```
void OnCallTransferred (IArguments * args)
```

VB

```
session_OnCallTransferred (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 41: OnCallTransferred Parameters

Keyword	Type	Description
PeripheralID	INT	The Unified ICM PeripheralID of the ACD where the call activity occurred.
PeripheralType	SHORT	The type of the peripheral.
CallType	SHORT	The general classification of the call type.
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.
RouterCallKeyDay	INT	Together with the RouterCallKeyCallID field forms the unique 64-bit key for locating this call's records in the Unified ICM database. Only provided for Post-routed and Translation-routed calls.
RouterCalKeyCallID	INT	The call key created by the Unified ICM. The Unified ICM resets this counter at midnight.
ConnectionCallID	UINT	The Call ID value assigned to this call by the peripheral or the Unified ICM.

Keyword	Type	Description
ANI (optional)	STRING	The calling line ID of the caller.
DNIS (optional)	STRING	The DNIS provided with the call.
UserToUserInfo (Optional)	STRING	The ISDN user-to-user information element. unspecified, up to 131 bytes.
DialedNumber (Optional)	STRING	The number dialed.
CallerEnteredDigits (Optional)	STRING	The digits entered by the caller in response to VRU prompting.
ServiceNumber (Optional)	INT	The service that the call is attributed to, as known to the peripheral. May contain the special value NULL_SERVICE when not applicable or not available.
ServiceID (Optional)	INT	The Unified ICM ServiceID of the service that the call is attributed to. May contain the special value NULL_SERVICE when not applicable or not available.
SkillGroupNumber (Optional)	INT	The optional, user-defined number of the agent SkillGroup the call is attributed to, as known to the peripheral. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
SkillGroupID (Optional)	INT	The system-assigned identifier of the agent SkillGroup the call is attributed to. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
SkillGroupPriority (Optional)	SHORT	The priority of the skill group, or 0 when skill group priority is not applicable or not available.
CallWrapupData (Optional)	STRING	Call-related wrap-up data.
CallVariable1 (Optional)	STRING	Call-related variable data.
...	STRING	...
CallVariable10 (Optional)	STRING	Call-related variable data.

Keyword	Type	Description
CallStatus (Optional)	SHORT	The current status of the call.
ECC (optional)	ARGUMENTS	Arguments array that contains all of the Expanded Call Context variables in use; for example: user.ArrayVariable[0]user.ArrayVariable[1] ...user.ArrayVariable[n]user.ScalarVariable
CTIClients (Optional)	ARGUMENTS	Arguments array that contains the information about the number of clients that are using the Call object; for example: CTIClient[1] CTIClientSignatureCTIClientTimestamp
ICMEnterpriseUniqueID (Optional)	STRING	Required only when the call is pre-routed.

OnClearCallConf

The OnClearCallConf event is fired to the client to indicate that the CTI server received a Clear request.

Syntax

C++

```
void OnClearCallConf (Arguments & args);
```

COM

```
HRESULT OnClearCallConf ([in] IArguments * args);
```

VB

```
OnClearCallConf (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following field.

Table 42: OnClearCallConf Parameters

Keyword	Type	Description
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.

OnClearConnectionConf

The OnClearConnectionConf event is fired to the client to indicate that the CTI server received a ClearConnection request.

Syntax

C++

```
void OnClearConnectionConf (Arguments & args);
```

COM

```
HRESULT OnClearConnectionConf ([in] IArguments * args);
```

VB

```
Session_ OnClearConnectionConf (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following field.

Table 43: OnClearConnectionConf Parameters

Keyword	Type	Description
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.

OnConferenceCallConf

The OnConferenceCallConf event is fired to the client to indicate that the CTI server received a ConferenceCall or SingleStepConferenceCall request.

Syntax

C++

```
void OnConferenceCallConf (Arguments & args);
```

COM

```
HRESULT OnConferenceCallConf ([in] IArguments * args);
```

VB

```
Session_ OnConferenceCallConf (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following field.

Table 44: OnConferenceCallConf Parameters

Keyword	Type	Description
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.

OnConsultationCallConf

The OnConsultationCallConf event is fired to the client to indicate that the CTI server received a MakeConsultCall request.

Syntax

C++

```
void OnConsultationCallConf (Arguments & args);
```

COM

```
HRESULT OnConsultationCallConf ([in] IArguments * args);
```

VB

```
OnConsultationCallConf (ByVal args as CTIOSCLIENTLIB.IArguments)Parameters
```

Parameters

args

Arguments array containing the following field.

Table 45: OnConsulationCallConf Parameters

Keyword	Type	Description
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.

OnControlFailureConf

The OnControlFailureConf event is generated when a request to the peripheral (the ACD) fails.

Syntax

C++

```
void OnControlFailureConf(Arguments& args)
```

COM

```
void OnControlFailureConf (IArguments * args)
```

VB

```
session_OnControlFailureConf (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 46: OnControlFailureConf Parameters

Keyword	Type	Description
PeripheralID	INT	Peripheral ID.
FailureCode	SHORT	Code ID.
PeripheralError Code	INT	Peripheral-specific error data, if available. Zero otherwise.
AgentID	STRING	Agent ID that represents a specific client.
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.
MessageType	INT	Contains the CTI OS Command Request ID that failed to execute. The message types included in this parameter are those to used to control Call, Agent State and Supervisor actions. For more information, see CTI OS Keywords and Enumerated Types.
ErrorMessage	STRING	String text containing the description of the failure.

OnHoldCallConf

The OnHoldCallConf event is fired to the client to indicate that the CTI server received a Hold request.

Syntax

C++

```
void OnHoldCallConf (Arguments & args);
```

COM

```
HRESULT OnHoldCallConf ([in] IArguments * args);
```

VB

```
Session_ OnHoldCallConf (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following field.

Table 47: OnHoldCallConf Parameters

Keyword	Type	Description
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.

OnMakePredictiveCallConf

Not supported.

OnReconnectCallConf

The OnReconnectCallConf event is fired to the client to indicate that the CTI server received a Reconnect request.

Syntax

C++

```
void OnReconnectCallConf (Arguments & args);
```

COM

```
HRESULT OnReconnectCallConf ([in] IArguments * args);
```

VB

```
OnReconnectCallConf (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following field.

Table 48: OnMakePredictiveCallConf Parameters

Keyword	Type	Description
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.

OnReleaseCallConf

Not supported.

OnRetrieveCallConf

The OnRetrieveCallConf event is fired to the client to indicate that the CTI server received a RetrieveCall request.

Syntax

C++

```
void OnRetrieveCallConf (Arguments & args);
```

COM

```
HRESULT OnRetrieveCallConf ([in] IArguments * args);
```

VB

```
Session_ OnRetrieveCallConf (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following field.

Table 49: OnReleaseCallConf Parameters

Keyword	Type	Description
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.

OnSendDTMFConf

The OnSendDTMFConf event is fired to the client to indicate that the CTI server received a SendDTMF request.

Syntax

C++

```
void OnSendDTMFConf (Arguments & args);
```

COM

```
HRESULT OnSendDTMFConf ([in] IArguments * args);
```

VB

```
Session_ OnSendDTMFConf (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Not used; reserved for future use.

OnSetCallDataConf

The OnSetCallDataConf event is fired to the client to indicate that the CTI server received a SetCallData request.

Syntax

C++

```
void OnSetCallConf (Arguments & args);
```

COM

```
HRESULT OnClearCallConf ([in] IArguments * args);
```

VB

```
OnClearCallConf (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following field.

Table 50: OnReleaseCallConf Parameters

Keyword	Type	Description
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.

OnSnapshotCallConf

The OnSnapshotCallConf event is generated when a SnapshotCall request for a specific call is successful. It contains all the information known about the specific connection at that point in time.

Syntax

C++

```
void OnSnapshotCallConf(Arguments& args)
```

COM

```
void OnSnapshotCallConf (IArguments * args)
```

VB

```
session_OnSnapshotCallConf (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 51: OnSnapShotCallConf Parameters

Keyword	Type	Description
PeripheralID	INT	The Unified ICM PeripheralID of the ACD where the call activity occurred.
CallType	SHORT	The general classification of the call type.
UniqueObjectID	STRING	An object ID that uniquely identifies the call object.

Keyword	Type	Description
DialedNumber	STRING	The number dialed.
CallerEnteredDigits	STRING	The digits entered by the caller in response to VRU prompting.
CallWrapupData	STRING	Call-related wrap-up data.
CallVariable1 (Optional)	STRING	Call-related variable data.
...	STRING	...
CallVariable10 (Optional)	STRING	Call-related variable data.
CustomerPhone Number	STRING	The customer phone number associated with the call.
CustomerAccount Number	STRING	The customer account number associated with the call.
ECC	ARGUMENTS	Arguments array that contains all of the Expanded Call Context variables in use; for example: user.ArrayVariable[0]user.ArrayVariable[1] ...user.ArrayVariable[n]user.ScalarVariable
CTIClients (Optional)	ARGUMENTS	Arguments array that contains the information about the number of clients that are using the Call object; for example: CTIClient[1] CTIClientSignatureCTIClientTimestamp
RouterCallKeyDay	INT	Together with the RouterCallKeyCallID field forms the unique 64-bit key for locating this call's records in the Unified ICM database. Only provided for Post-routed and Translation-routed calls.
RouterCallKeyCallID	INT	The call key created by the Unified ICM. The Unified ICM resets this counter at midnight.
NumNamedVariables	SHORT	Number of Named variables.
NumNamedArrays	SHORT	Number of Named Arrays.
NumCallDevices	SHORT	Number of devices associated with the call.

Keyword	Type	Description
CalledDeviceID	STRING	The device identifier of the called device.
ConnectionCallID	UINT	The Call ID value assigned to this call by the peripheral or the Unified ICM.
CallStatus	SHORT	The current status of the call.
The following fields appear if they have information in them.		
ANI	STRING	The calling line ID of the caller.
UserToUserInfo	STRING	The ISDN user-to-user information element associated with the call.
DNIS	STRING	The DNIS provided with the call.

If the MinimizeEventArgs registry entry is set to 0, the SnapshotCallConf event contains the following additional fields.

Table 52: SnapshotCallConf Additional Fields

Keyword	Type	Description
ICMEnterpriseUnique ID	STRING	This string is a globally unique key for this contact, which corresponds to the Unified ICM 64 bit key. This parameter can be used to match this contact to a follow-on call event.
CallConnectionCallID (optional)	UINT	The CallID value assigned to the call.
CallConnectionDeviceID Type (optional)	SHORT	Indicates the type of the connection identifier supplied in the following CallConnectionDeviceID floating field. This field always immediately follows the corresponding CallConnectionCallID field.
CallConnectionDeviceID (optional)	STRING	The identifier of the call connection. This field always immediately follows the corresponding CallConnectionDeviceIDType field.
CallDeviceConnection State	SHORT	The active state of the call. This field always immediately follows the corresponding CallConnection DeviceID field.

Keyword	Type	Description
CallDeviceType	SHORT	Indicates the type of the device identifier supplied in the CallDeviceID field.

OnTransferCallConf

The OnTransferCallConf event is fired to the client to indicate that the CTI server received a TransferCall or SingleStepTransferCall request.

Syntax

C++

```
void OnTransferCallConf (Arguments & args);
```

COM

```
HRESULT OnTransferCallConf ([in] IArguments * args);
```

VB

```
Session_ OnTransferCallConf (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following field.

Table 53: OnTransferCallConf Parameters

Keyword	Type	Description
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.

IAgentEvents Interface

The Agent object fires events on the IAgentEvents interface. The following events are published to subscribers of the IAgentEvents interface.

OnAgentDeskSettingsConf

The OnAgentDeskSettingsConf event confirms successful completion of the request and provides the query response.

Syntax

C++

```
void OnAgentDeskSettings(Arguments& args)
```

COM

```
void OnAgentDeskSettings (IArguments * args)
```

VB

```
session_OnAgentDeskSettings (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 54: OnAgentDeskSettingsConf Parameters

Keyword	Type	Description
InvokeID	UINT	Set to the same value as the InvokeID from the corresponding request message.
PeripheralID	UINT	The Unified ICM PeripheralID of the ACD where the device is located.
DeskSettingsMask	UINT	A bitwise combination of the Boolean desk setting Masks listed in the table below.
WrapupData IncomingMode	UINT	Indicates whether the agent is allowed or required to enter wrap-up data after an inbound call: 0 = Required, 1 = Optional, 2 = Not allowed, 3 = Required With WrapupData.
WrapupData OutgoingMode	UINT	Indicates whether the agent is allowed or required to enter wrap-up data after an outbound call: 0 = Required, 1 = Optional, 2 = Not allowed.
LogoutNon ActivityTime	UINT	Number of seconds of non-activity at the desktop after which the Unified ICM automatically logs out the agent.
QualityRecordingRate	UINT	Indicates how frequently calls to the agent are recorded.
RingNoAnswer Time	UINT	Number of seconds a call can ring at the agent's station before being redirected.
SilentMonitor WarningMessage	UINT	Set for a warning message box to prompt on agent desktop when silent monitor starts.
SilentMonitor AudibleIndication	UINT	Set for an audio click at beginning of the silent monitor.
SupervisorAssist CallMethod	UINT	Set for PIM to create a blind conference call for supervisor assist request; otherwise creates consultative call.

Keyword	Type	Description
EmergencyCall Method	UINT	Set for PIM to create a blind conference call for emergency call request; otherwise creates a consultative call.
AutoRecordOn Emergency	UINT	Set for automatically record when emergency call request.
RecordingMode	UINT	Set for the recording request to go through Call Manager/PIM.
WorkModeTimer	UINT	Auto Wrap-up time out.
RingNoAnswer DN	UINT	The dialed number identifier for new re-route destination in the case of ring no answer.

Table 55: DeskSettingsMasks values

Mask Name	Description	Numeric Value
DESK_AVAIL_AFTER_ INCOMING_MASK	Set for automatically consider the agent available after handling an incoming call.	0x00000001
DESK_AVAIL_AFTER_OUTGOING_MASK	Set for automatically consider the agent available after handling an outbound call.	0x00000002
DESK_AUTO_ ANSWER_ENABLED_ MASK	Set when calls to the agent are automatically answered.	0x00000004
DESK_IDLE_REASON_REQUIRED_MASK	Set when the agent must enter a reason before entering the Idle state.	0x00000008
DESK_LOGOUT_ REASON_REQUIRED_MASK	Set when the agent must enter a reason before logging out.	0x00000010
DESK_SUPERVISOR_ CALLS_ALLOWED_MASK	Set when the agent can initiate supervisor assisted calls.	0x00000020
DESK_AGENT_TO_ AGENT_CALLS_ ALLOWED	Set when calls to other agents are allowed.	0x00000040
DESK_OUTBOUND_ACCESS_INTERNATIONAL_MASK	Set when the agent can initiate international calls.	0x00000080
DESK_OUTBOUND_ACCESS_PUBLIC_NET_MASK	Set when the agent can initiate calls through the public network.	0x00000100
DESK_OUTBOUND_ACCESS_PRIVATE_NET_MASK	Set when the agent can initiate calls through the private network.	0x00000200

Mask Name	Description	Numeric Value
DESK_OUTBOUND_ACCESS_OPERATOR_ASSISTED_MASK	Set when the agent can initiate operator assisted calls.	0x00000400
DESK_OUTBOUND_ACCESS_PBX_MASK	Set when the agent can initiate outbound PBX calls.	0x00000800
DESK_NON_ACD_CALLS_ALLOWED_MASK	Set when the agent can place or handle non-ACD calls.	0x00001000
DESK_AGENT_CAN_SELECT_GROUP_MASK	Set when the agent can select which groups they are logged into.	0x00002000

OnAgentGreetingControlConf

The OnAgentGreetingControlConf event confirms the successful completion of the SetAgentGreetingAction request.

Syntax

C++

```
void OnAgentGreetingControlConf(Arguments& args)
```

COM

```
void OnAgentGreetingControlConf (IArguments * args)
```

VB

```
session_OnAgentGreetingControlConf (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 56: OnAgentIGreetingControlConfEvent Parameters

Keyword	Type	Description
MessageHeader	MHDR	Standard Message Header.
InvokeID	UINT	Set to the same value as the InvokeID from the corresponding request message.

OnAgentInfoEvent

The OnAgentInfoEvent event is generated as a response to a query to the Agent Name Lookup Service and carries the agent's name. The CTI OS server generates this query when it is configured to do agent name lookup. The OnAgentInfoEvent event is sent to the client if the server obtained the information.

Syntax

C++

```
void OnAgentInfoEvent(Arguments& args)
```

COM

```
void OnAgentInfoEvent (IArguments * args)
```

VB

```
session_OnAgentInfoEvent (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 57: OnAgentInfoEvent Parameters

Keyword	Type	Description
UniqueObjectID	STRING	A unique object ID for the Agent object.
AgentLastName	STRING	Agent's last name.
AgentFirstName	STRING	Agent's first name.

OnAgentStateChange

The OnAgentStateChange event is generated when the agent state at the ACD changes. This can be as a response to a Login, Logout, or SetAgentState request.

Syntax

C++

```
void OnAgentStateChange(Arguments& args)
```

COM

```
void OnAgentStateChange (IArguments * args)
```

VB

```
session_OnAgentStateChange (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 58: OnAgentIStateChange Parameters

Keyword	Type	Description
PeripheralID	INT	The Unified ICM PeripheralID of the ACD where the agent state change occurred.
PeripheralType	SHORT	The type of the peripheral.
AgentState	SHORT	One of the values in Table 59: AgentState values, on page 68 representing the current overall state of the associated agent.
SkillGroupNumber	INT	The optional, user-defined number of the agent SkillGroup affected by the state change, as known to the peripheral. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
SkillGroupID	INT	The system-assigned identifier of the agent SkillGroup affected by the state change. May contain the special value NULL_SKILL_ GROUP when not applicable or not available.
StateDuration	INT	The number of seconds since the agent entered this state (typically 0).
SkillGroupPriority	SHORT	The priority of the skill group, or 0 when skill group priority is not applicable or not available.
EventReasonCode	SHORT	A peripheral-specific code indicating the reason for the state change.
SkillGroupState	SHORT	Values representing the current state of the associated agent with respect to the indicated Agent Skill Group.
AgentID	STRING	The agent's ACD login ID.
AgentExtension	STRING	The agent's ACD teleset extension.
CTIClientSignature (Optional)	STRING	The Client Signature of the CTI Client that is associated with this agent.

Keyword	Type	Description
Enablement Mask		Contains the bit-mask that specifies what buttons can be enabled or disabled when the agent is on this state.
UniqueObjectID	STRING	A unique object ID for the Agent object.
AgentInstrument	STRING	The agent's ACD instrument number.

The following table provides the AgentState values.

Table 59: AgentState values

enum Value	Description	Numeric Value
eLogin	The agent has logged on to the ACD. It does not necessarily indicate that the agent is ready to accept calls.	0
eLogout	The agent has logged out of the ACD and cannot accept any additional calls.	1
eNotReady	The agent is unavailable for any call work.	2
eAvailable	The agent is ready to accept a call.	3
eTalking	The agent is currently talking on a call (inbound, outbound, or inside).	4
eWorkNotReady	The agent is performing after call work, but will not be ready to receive a call when completed.	5
eWorkReady	The agent is performing after call work, and will be ready to receive a call when completed.	6
eBusyOther	The agent is busy performing a task associated with another active SkillGroup.	7
eReserved	The agent is reserved for a call that will arrive at the ACD shortly.	8
eUnknown	The agent state is currently unknown.	9

enum Value	Description	Numeric Value
eHold	The agent currently has all calls on hold.	10

Note Not all switches support all the states listed in the above table, and you should not make any assumptions about which states are supported on a particular switch without verification.

OnAgentStatistics

The OnAgentStatistics event is fired to the client to indicate that the CTI server received a request to enable agent statistics (via the EnableAgentStatistics method). The arrival of events event is determined by the configuration on the server.

The table under Parameters details all the agent statistics that can be received. To optimize bandwidth, the default configuration on the server is set to minimize the agent statistics sent. Only the statistics that the Agent Statistics grid is configured for are sent to the client. For more information about on how to configure the agent statistics grid and minimize agent statistics, see *CTI OS System Manager's Guide for Cisco Unified ICM/Contact Center Enterprise & Hosted*.

Syntax

C++

```
void OnAgentStatistics (Arguments & args);
```

COM

```
HRESULT OnAgentStatistics ([in] IArguments * args);
```

VB

```
Session_ OnAgentStatistics (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 60: OnAgentStatistics Parameters

Keyword	Description	Type
PeripheralID	The Unified ICM PeripheralID of the ACD where the agent is located.	INT
AgentExtension (required)	The agent's ACD teleset extension.	STRING
AgentID (required)	The agent's ACD login ID.	STRING
AgentInstrument (required)	The agent's ACD instrument number.	STRING

The OnAgentStatistics event contains all the agent statistics fields necessary to display the statistics configured on the CTI OS server.

OnChatMessage

The OnChatMessage event is generated when an asynchronous text message is received from another user (agent).

Syntax

C++

```
void OnChatMessage(Arguments& args)
```

COM

```
void OnChatMessage (IArguments * args)
```

VB

```
session_OnChatMessage (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 61: OnChatMessage Parameters

Keyword	Type	Description
Distribution	STRING	Currently the only supported value is “agent”.
AgentID	STRING	The AgentID of the message target.
Target	STRING	The AgentID of the message target.
Message	STRING	The text message provided by the sender.
Source	STRING	The AgentID of the message sender.

OnControlFailureConf

The OnControlFailureConf event is generated when the previously issued request, identified by the InvokeID field failed. It is sent in place of the corresponding confirmation message for that request.

Syntax

C++

```
void OnControlFailureConf(Arguments& args)
```

COM

```
void OnControlFailureConf (IArguments * args)
```

VB

```
session_OnControlFailureConf (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 62: OnControlFailureConf Parameters

Keyword	Type	Description
InvokeID	INT	InvokeID of the request that failed
FailureCode	SHORT	A value specifying the reason that the request failed. For a list of the Control Failure Code see the table below.
PeripheralError Code	INT	Peripheral-specific error data, if available. Zero otherwise.
AgentID	STRING	Agent ID that represents a specific client.
UniqueObjectID	STRING	An object ID that uniquely identifies the Call object.
MessageType	INT	Contains the CTI OS Command Request ID that failed to execute. The message types included in this parameter are those to used to control Call, Agent State and Supervisor actions. For more information, see CTI OS Keywords and Enumerated Types.
ErrorMessage	STRING	String text containing the description of the failure.

Table 63: Control Failure Codes

Status Code	Description	Value
E_CTI_NO_ERROR	No error occurred.	0
E_CTI_INVALID_ VERSION	The CTI Server does not support the protocol version number requested by the CTI client.	1
E_CTI_INVALID_ MESSAGE_ TYPE	A message with an invalid message type field was received.	2

Status Code	Description	Value
E_CTI_INVALID_ FIELD_TAG	A message with an invalid floating field tag was received.	3
E_CTI_SESSION_ NOT_OPEN	No session is currently open on the connection.	4
E_CTI_SESSION_ ALREADY_ OPEN	A session is already open on the connection.	5
E_CTI_REQUIRED_ DATA_ MISSING	The request did not include one or more floating items that are required.	6
E_CTI_INVALID_ PERIPHERAL_ID	A message with an invalid PeripheralID value was received.	7
E_CTI_INVALID_ AGENT_ DATA	The provided agent data items are invalid.	8
E_CTI_AGENT_NOT_ LOGGED_ON	The indicated agent is not currently logged in.	9
E_CTI_DEVICE_IN_ USE	The indicated agent teleset is already associated with a different CTI client.	10
E_CTI_NEW_ SESSION_ OPENED	This session is being terminated due to a new session open request from the client.	11
E_CTI_FUNCTION_ NOT_ AVAILABLE	A request message was received for a function or service that was not granted to the client.	12
E_CTI_INVALID_ CALLID	A request message was received with an invalid CallID value.	13
E_CTI_PROTECTED_ VARIABLE	The CTI client cannot update the requested variable.	14
E_CTI_CTI_SERVER_ OFFLINE	The CTI Server cannot function normally. The CTI client closes the session upon receipt of this error.	15
E_CTI_TIMEOUT	The CTI Server failed to respond to a request message within the time-out period, or no messages were received from the CTI client within the IdleTimeout period.	16
E_CTI_UNSPECIFIED_FAILURE	An unspecified error occurred.	17
E_CTI_INVALID_ TIMEOUT	The IdleTimeout field contains a value that is less than 20 seconds (4 times the minimum heartbeat interval of 5 seconds).	18
E_CTI_INVALID_ SERVICE_MASK	The ServicesRequested field has unused bits set. All unused bit positions must be zero.	19
E_CTI_INVALID_ CALL_MSG_MASK	The CallMsgMask field has unused bits set. All unused bit positions must be zero.	20
E_CTI_INVALID_ AGENT_ STATE_ MASK	The AgentStateMask field has unused bits set. All unused bit positions must be zero.	21

Status Code	Description	Value
E_CTI_INVALID_ RESERVED_ FIELD	A Reserved field has a non-zero value.	22
E_CTI_INVALID_ FIELD_ LENGTH	A floating field exceeds the allowable length for that field type.	23
E_CTI_INVALID_ DIGITS	A STRING field contains characters that are not digits (“0” through “9”).	24
E_CTI_BAD_ MESSAGE_ FORMAT	The message is improperly constructed. This can be caused by omitted or incorrectly sized fixed message fields.	25
E_CTI_INVALID_ TAG_FOR_MSG_ TYPE	A floating field tag is present that specifies a field that does not belong in this message type.	26
E_CTI_INVALID_ DEVICE_ID_ TYPE	A DeviceIDType field contains an invalid value.	27
E_CTI_INVALID_ LCL_CONN_ STATE	A LocalConnectionState field contains an invalid value.	28
E_CTI_INVALID_ EVENT_ CAUSE	An EventCause field contains an invalid value.	29
E_CTI_INVALID_ NUM_ PARTIES	The NumParties field contains a value that exceeds the maximum (16).	30
E_CTI_INVALID_ SYS_ EVENT_ID	The SystemEventID field contains an invalid value.	31
E_CTI_ INCONSISTENT_ AGENT_DATA	The provided agent extension, agent ID, and/or agent instrument values are inconsistent with each other.	32
E_CTI_INVALID_ CONNECTION_ID_ TYPE	A ConnectionDeviceIDType field contains an invalid value.	33
E_CTI_INVALID_ CALL_TYPE	The CallType field contains an invalid value.	34
E_CTI_NOT_CALL_ PARTY	A CallDataUpdate or Release Call request specified a call that the client is not a party to.	35
E_CTI_INVALID_ PASSWORD	The ClientID and Client Password provided in an OPEN_REQ message is incorrect.	36
E_CTI_CLIENT_ DISCONNECTED	The client TCP/IP connection was disconnected without a CLOSE_REQ.	37
E_CTI_INVALID_ OBJECT_ STATE	An invalid object state value was provided.	38
E_CTI_INVALID_ NUM_ SKILL_GROUPS	An invalid NumSkillGroups value was provided.	39
E_CTI_INVALID_ NUM_LINES	An invalid NumLines value was provided.	40

Status Code	Description	Value
E_CTI_INVALID_ LINE_TYPE	An invalid LineType value was provided.	41
E_CTI_INVALID_ ALLOCATION_STATE	An invalid AllocationState value was provided.	42
E_CTI_INVALID_ ANSWERING_ MACHINE	An invalid AnsweringMachine value was provided.	43
E_CTI_INVALID_ CALL_MANNER_ TYPE	An invalid CallMannerType value was provided.	44
E_CTI_INVALID_ CALL_PLACEMENT_ TYPE	An invalid CallPlacementType value was provided.	45
E_CTI_INVALID_ CONSULT_ TYPE	An invalid ConsultType value was provided.	46
E_CTI_INVALID_ FACILITY_ TYPE	An invalid FacilityType value was provided.	47
E_CTI_INVALID_ MSG_TYPE_ FOR_ VERSION	The provided MessageType is invalid for the opened protocol version.	48
E_CTI_INVALID_ TAG_FOR_ VERSION	A floating field tag value is invalid for the opened protocol version.	49
E_CTI_INVALID_ AGENT_WORK_ MODE	An invalid AgentWorkMode value was provided.	50
E_CTI_INVALID_ CALL_OPTION	An invalid call option value was provided.	51
E_CTI_INVALID_ DESTINATION_ COUNTRY	An invalid destination country value was provided.	52
E_CTI_INVALID_ ANSWER_DETECT_ MODE	An invalid answer detect mode value was provided.	53
E_CTI_MUTUALLY_ EXCLUS_DEVICEID_ TYPES	A peripheral monitor request cannot specify both a call and a device.	54
E_CTI_INVALID_ MONITORID	An invalid monitorID value was provided.	55
E_CTI_SESSION_ MONITOR_ ALREADY_EXISTS	A requested session monitor was already created.	56
E_CTI_SESSION_ MONITOR_IS_ CLIENTS	A client may not monitor its own session.	57
E_CTI_INVALID_ CALL_CONTROL_ MASK	An invalid call control mask value was provided.	58
E_CTI_INVALID_ FEATURE_MASK	An invalid feature mask value was provided.	59
E_CTI_INVALID_ TRANSFER_ CONFERENCE_ SETUP_MASK	An invalid transfer conference setup mask value was provided.	60

Status Code	Description	Value
E_CTI_INVALID_ ARRAY_INDEX	An invalid named array index value was provided.	61
E_CTI_INVALID_ CHARACTER	An invalid character value was provided.	62
E_CTI_CLIENT_NOT_FOUND	There is no open session with a matching ClientID.	63
E_CTI_SUPERVISOR_NOT_FOUND	The agent's supervisor is unknown or does not have an open CTI session.	64
E_CTI_TEAM_NOT_ FOUND	The agent is not a member of an agent team.	65
E_CTI_NO_CALL_ ACTIVE	The specified agent does not have an active call.	66
E_CTI_NAMED_ VARIABLE_NOT_ CONFIGURED	The specified named variable is not configured in the Unified ICM database.	67
E_CTI_NAMED_ ARRAY_NOT_ CONFIGURED	The specified named array is not configured in the Unified ICM database.	68
E_CTI_INVALID_ CALL_VARIABLE_ MASK	The specified call variable mask in not valid.	69
E_CTI_ELEMENT_ NOT_FOUND	An internal error occurred manipulating a named variable or named array element.	70
E_CTI_INVALID_ DISTRIBUTION_TYPE	The specified distribution type is invalid.	71
E_CTI_INVALID_ SKILL_GROUP	The specified skill group is invalid.	72
E_CTI_TOO_MUCH_ DATA	The total combined size of named variables and named arrays cannot exceed the limit of 2000 bytes.	73
E_CTI_VALUE_TOO_LONG	The value of the specified named variable or named array element exceeds the maximum permissible length.	74
E_CTI_SCALAR_ FUNCTION_ON_ ARRAY	A NamedArray was specified with a NamedVariable tag.	75
E_CTI_ARRAY_ FUNCTION_ON_ SCALAR	A NamedVariable was specified with a NamedArray tag.	76
E_CTI_INVALID_ NUM_NAMED_ VARIABLES	The value in the NumNamedVariables field is different than the number of NamedVariable floating fields in the message.	77
E_CTI_INVALID_ NUM_NAMED_ ARRAYS	The value in the NumNamedArrays field is different than the number of NamedArray floating fields in the message.	78

OnEmergencyCall

The OnEmergencyCall event indicates that a CTI client (with Supervisory capabilities) is handling the indicated call as an emergency call. This event only applies to ACDs with Supervisor capabilities.

Syntax

C++

```
void OnEmergencyCall(Arguments& args)
```

COM

```
void OnEmergencyCall (IArguments * args)
```

VB

```
session_OnEmergencyCall (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 64: OnEmergencyCall Parameters

Keyword	Type	Description
PeripheralID	INT	The Unified ICM PeripheralID of the ACD where the call is located.
Connection CallID	INT	The Call ID value assigned to the call by the peripheral or the Unified ICM.
ConnectionDevice IDType	SHORT	Indicates the type of the connection identifier supplied in the ConnectionDeviceID floating field.
SessionID	INT	The CTI client SessionID of the CTI client making the notification.
Connection DeviceID	INT	The identifier of the connection between the call and the agent's device.
ClientID (required)	STRING	The ClientID of the client making the notification.
ClientAddress (Required)	STRING	The IP address of the client making the notification.
AgentExtension (Required)	STRING	The agent's teleset extension.
AgentID (required)	STRING	The agent's ACD login ID.

Keyword	Type	Description
AgentInstrument (required)	STRING	The agent's ACD instrument number.

Remarks

Supported for use with Unified CCE only.

OnLogoutFailed

The OnLogoutFailed is always generated before (or with) an OnControlFailureConf event and is identical to it but is generated only when a Logout request fails.

Syntax

```
C++:  void OnLogoutFailed (Arguments& args)
COM:  void OnLogoutFailed (IArguments * args)
VB:   session_OnLogoutFailed (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 65: OnLogoutFailed Parameters

Keyword	Type	Description
InvokeID	INT	InvokeID of the request that failed.
FailureCode	SHORT	A value specifying the reason that the request failed. For a list of the Control Failure Codes see Table 62: OnControlFailureConf Parameters, on page 71.
Peripheral ErrorCode	INT	Peripheral-specific error data, if available. Zero otherwise.

OnMakeCallConf

The OnMakeCallConf event confirms the successful completion of the MakeCall request. It conveys the information detailed in the table under Parameters.

Syntax

C++

```
C++: int OnMakeCallConf (Arguments & args);
```

COM

```
HRESULT OnMakeCallConf ([in] IArguments * args);
```

VB

```
Session_ OnMakeCallConf (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Java

```
void OnMakeCallConf (Arguments args);
```

Parameters

args

Arguments array containing the following fields.

Table 66: OnMakeCallConf Parameters

Keyword	Description	Type
NewConnectionCallID	The Call ID value assigned to the call by the peripheral or the Unified ICM.	UINT
NewConnectionDevice IDType	Indicates the type of the connection identifier supplied in the New ConnectionDeviceID floating field.	SHORT
LineHandle	Identifies the teleset line used, if known. Otherwise this field is set to 0xffff.	SHORT
LineType	Indicates the type of the teleset line given in the LineHandle field.	SHORT
NewConnectionDeviceID (required)	The identifier of the connection between the call and the device.	STRING

OnNewAgentTeamMember

The OnNewAgentTeamMember event informs the supervisor about a new agent team member. The event is typically received in response to a RequestAgentTeamList request from the supervisor object. It is also received when CTI OS Server receives an AGENT_TEAM_CONFIG_EVENT indicating a change in agent team configuration (add/remove).

Syntax

C++

```
void OnNewAgentTeamMember (Arguments& args)
```

COM

```
void OnNewAgentTeamMember (IArguments * args)
```

VB

```
session_OnNewAgentTeamMember (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array that can contain the following fields. Not all fields are always returned. Skillgroup and AgentInstrument are not returned if the agent is not logged in.

Table 67: OnNewAgentTeamMember Parameters

Keyword	Type	Description
PeripheralID	STRING	The Unified ICM PeripheralID of the agent's ACD.
UniqueObjectID	STRING	Unique object ID of the Agent object for this agent.
AgentState	SHORT	One of the values in Table 59: AgentState values, on page 68 representing the current state of the associated agent.
NumSkillGroups	INT	The number of skill groups that the agent is currently associated with, up to a maximum of 99.
AgentID	STRING	Agent's ACD login.
AgentExtension	STRING	Agent's ACD teleset extension.
AgentInstrument	STRING	Agent's ACD instrument number.
AgentLastName	STRING	Agent's last name.
AgentFirstName	STRING	Agent's first name.
AgentName	STRING	Agent's full name.
AgentAvailability Status	SHORT	The current status of the agent's availability to take a call.
EventReasonCode	SHORT	A peripheral-specific code indicating the reason for the change in agent state to NotReady.
EnablementMask	INT	Contains the bit-mask that specifies what buttons can be enabled or disabled when the agent is on the state specified in the AgentState field.
SupervisorID	STRING	The ID of the agent's supervisor.

Keyword	Type	Description
AgentFlags	INT	Used to describe the agent carried in this event. The possible values for this field as well as their meanings are as follows: • TeamMemberFlags.AGENT_FLAG_REGULAR_AGENT - Value is 0. The agent is a regular agent. • TeamMemberFlags.AGENT_FLAG_PRIMARY_SUPERVISOR - Value is 1. The agent is a primary supervisor. • TeamMemberFlags.AGENT_FLAG_TEMPORARY_AGENT - Value is 2. The agent is a temporary agent. • TeamMemberFlags.AGENT_FLAG_SUPERVISOR - Value is 4. The agent is a supervisor.
Skillgroup[1}	ARGUMENTS	Arguments array containing information about the agent's first skillgroup. The array contains the following arguments: • SkillGroupNumber • SkillGroupID • StateDuration • SkillGroupPriority
Skillgroup[n]	ARGUMENTS	Arguments array containing information about the agent's nth skillgroup.
ConfigOperation	USHORT	Used to describe a change to the team. The possible values for this field as well as their meanings are as follows: • TeamMemberFlags.CONFIG_OPERATION_ADD_AGENT - Value is 1 - The agent belongs to the team. • TeamMemberFlags.CONFIG_OPERATION_REMOVE_AGENT - Value is 2 - The agent no longer belongs to the team.

OnPostLogout

The OnPostLogout event is generated after the agent has logged out. Arrival of this event guarantees that the agent state event signalling the agent's transition to logout state was received and handled by all interested event listeners.

Syntax

C++

```
void OnPostLogout(Arguments& args)
```

COM

```
void OnPostLogout (IArguments * args)
```

VB

```
session_OnPostLogout (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 68: OnPostLogout Parameters

Keyword	Type	Description
PeripheralID	INT	The Unified ICM PeripheralID of the ACD where the agent state change occurred.
PeripheralType	SHORT	The type of the peripheral.
AgentState	SHORT	One of the values in Table 59: AgentState values, on page 68 representing the current overall state of the associated agent.
SkillGroupNumber	INT	The optional, user-defined number of the agent SkillGroup affected by the state change, as known to the peripheral. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
SkillGroupID	INT	The system-assigned identifier of the agent SkillGroup affected by the state change. May contain the special value NULL_SKILL_ GROUP when not applicable or not available.
StateDuration	INT	The number of seconds since the agent entered this state (typically 0).
SkillGroupPriority	SHORT	The priority of the skill group, or 0 when skill group priority is not applicable or not available.
EventReasonCode	SHORT	A peripheral-specific code indicating the reason for the state change.
SkillGroupState	SHORT	Values representing the current state of the associated agent with respect to the indicated Agent Skill Group.
AgentID	STRING	The agent's ACD login ID.

Keyword	Type	Description
AgentExtension	STRING	The agent's ACD teleset extension.
CTIClientSignature (Optional)	STRING	The Client Signature of the CTI Client that is associated with this agent.
EnablementMask	INT	Contains the bit-mask that specifies what buttons can be enabled or disabled when the agent is on this state.
UniqueObjectID	STRING	A unique object ID for the Agent object.
AgentInstrument	STRING	The agent's ACD instrument number.

Remarks

When PG failover occurs, the client application can receive an OnPostLogout event with an EventReasonCode of CTIOS_FORCED_LOGOUT_REASON_CODE. For example, this can happen on an Unified CCE system after reconnecting to a different server during a failover, because there is a race condition of the PG logging the agent out and the client reconnecting to the other server before it happens. If this happens, the client application should not disconnect from CTI OS Server.

OnPreLogout

The OnPreLogout event just before the agent is logged out. It allows for any cleanup or logic that needs to be done before logout is completed.

Syntax

C++

```
void OnPreLogout(Arguments& args)
```

COM

```
void OnPreLogout (IArguments * args)
```

VB

```
session_OnPreLogout (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 69: OnPreLogout Parameters

Keyword	Type	Description
PeripheralID	INT	The Unified ICM PeripheralID of the ACD where the agent state change occurred.
PeripheralType	SHORT	The type of the peripheral.
AgentState	SHORT	One of the values in Table 59: AgentState values, on page 68 representing the current overall state of the associated agent.
SkillGroupNumber	INT	The optional, user-defined number of the agent SkillGroup affected by the state change, as known to the peripheral. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
SkillGroupID	INT	The system-assigned identifier of the agent SkillGroup affected by the state change. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
StateDuration	INT	The number of seconds since the agent entered this state (typically 0).
SkillGroupPriority	SHORT	The priority of the skill group, or 0 when skill group priority is not applicable or not available.
EventReasonCode	SHORT	A peripheral-specific code indicating the reason for the state change.
SkillGroupState	SHORT	Values representing the current state of the associated agent with respect to the indicated Agent Skill Group.
AgentID	STRING	The agent's ACD login ID.
AgentExtension	STRING	The agent's ACD teleset extension.
CTIClientSignature (Optional)	STRING	The Client Signature of the CTI Client that is associated with this agent.

Keyword	Type	Description
Enablement Mask		Contains the bit-mask that specifies what buttons can be enabled or disabled when the agent is on this state.
UniqueObjectID	STRING	A unique object ID for the Agent object.
AgentInstrument	STRING	The agent's ACD instrument number.

OnQueryAgentStateConf

The OnQueryAgentStateConf event is generated and returned by the server at login as a response to the QueryAgentState() request. A user cannot issue this request.

Syntax

C++

```
void OnQueryAgentStateConf(Arguments& args)
```

COM

```
void OnQueryAgentStateConf (IArguments * args)
```

VB

```
session_OnQueryAgentStateConf (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 70: OnQueryAgentStateConf Parameters

Keyword	Type	Description
AgentID	STRING	Agent's ACD login.
AgentExtension	STRING	Agent's ACD teleset extension.
AgentInstrument	STRING	Agent's ACD instrument number.
AgentState	SHORT	One of the values in Table 59: AgentState values, on page 68 representing the current state of the associated agent.
NumSkillGroups	INT	The number of skill groups that the agent is currently associated with, up to a maximum of 20.

Keyword	Type	Description
SkillGroup[j]	ARGUMENTS	Argument array that contains Skill Group information for the j-th element less than NumSkillGroups. The message contains NumSkillGroups elements of this type.
MRDID	INT	Media Routing Domain ID as configured in Unified ICM and the ARM client.
NumTasks	INT	The number of tasks currently assigned to the agent—this is the number that Unified ICM compares to the MaxTaskLimit to decide if the agent is available to be assigned additional tasks. This includes active tasks as well as those that are offered, paused, and in wrapup.
AgentMode	SHORT	The mode that the agent is not in when the login completes. ROUTABLE = 0, NOT ROUTABLE = 1
MaxTaskLimit	INT	The maximum number of tasks that the agent can simultaneously work on.
ICMAgentID	INT	The Unified ICM Skill Target ID, a unique agent identifier for Unified ICM.

Keyword	Type	Description
Agent Availability Status	INT	An agent is available to work on a task in this Media Routing Domain if the agent meets all of these conditions: • The agent is routable for this Media Routing Domain. • The agent is not in Not Ready state for skill groups in other Media Routing Domain. • The agent is temp routable, meaning that the agent is not in Reserved, Active, Work-Ready, or Work-Not Ready state on a non-interruptible task in another Media Routing Domain. • The agent has not reached the maximum task limit for this Media Routing Domain. An available agent is eligible to be assigned a task. Who can assign a task to the agent is determined by whether or not the agent is Routable. An agent is *ICMAvailable* in MRD X if he is available in X and Routable with respect to X. An agent is *ApplicationAvailable* in MRD X if he is available in X and not Routable with respect to X. Otherwise an agent is NotAvailable in MRD X. NOT AVAILABLE = 0, ICM AVAILABLE = 1, APPLICATION AVAILABLE=2

Each SkillGroup*[j]* field in the message contains the following information.

Table 71: SkillGroup Parameters

Keyword	Type	Description
SkillGroupNumber	INT	The optional, user-defined number of an agent SkillGroup queue that the call was added to, as known to the peripheral. May contain the special value NULL_SKILL_GROUP when not applicable or not available.
SkillGroupID	INT	The system-assigned identifier of the agent SkillGroup the call is attributed to. May contain the special value NULL_SKILL_GROUP when not applicable or available.
SkillGroupPriority	SHORT	The priority of the skill group, or 0 when the skill group priority is not applicable or not available.
SkillGroupState	SHORT	One of the values representing the current state associated agent with respect to the skill group.

OnSetAgentModeEvent

The OnSetAgentModeEvent event indicates that the client made a successful AgentMode connection.

Syntax

C++

```
void OnSetAgentModeEvent (Arguments& args)
```

COM

```
void OnSetAgentModeEvent (IArguments * args)
```

VB

```
Session_OnSetAgentModeEvent (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 72: OnSetAgentModeEven Parameters

Keyword	Type	Description
PeripheralID	STRING	ID of the Unified ICM Peripheral ACD associated with the agent.

Keyword	Type	Description
AgentID	STRING	The agent's ID.
UniqueObject ID	STRING	The new unique object ID for the Agent object.
ClientAgent TemporaryID	STRING	Temporary ID used before server passes the new unique object ID.
CIL ConnectionID	STRING	ID of the client's connection on the server.
StatusSystem	ARGUMENTS	Arguments array containing the following elements: • StatusCTIServer • StatusCtiServerDriver • StatusCentralController • StatusPeripherals (Arguments array with a peripheral ID for each key and a boolean true/false value indicating if that peripheral is online.)

OnSetAgentStateConf

The OnSetAgentStateConf confirmation message is fired to the client to indicate that the CTI server received the SetAgentState request. This confirmation message does not indicate that the agent has changed to the desired state; rather, the programmer should expect one or more OnAgentStateChange events to indicate the change of state.

Syntax

C++

```
int OnSetAgentStateConf (Arguments & args);
```

COM

```
HRESULT OnSetAgentStateConf ([out] IArguments * args);
```

VB

```
Session_ OnSetAgentStateConf (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Java

```
void OnSetAgentStateConf (Arguments args);
```

Parameters

args

Not used; reserved for future use.

OnStartMonitoringAgent

The OnStartMonitoringAgent event is generated when a new agent is selected to be monitored in response to a StartMonitoringAgent() request.

Syntax

C++

```
void OnStartMonitoringAgent (Arguments& args)
```

COM

```
void OnStartMonitoringAgent (IArguments * args)
```

VB

```
session_OnStartMonitoringAgent (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 73: OnStartMonitoringAgent Parameters

Keyword	Type	Description
UniqueObjectID	STRING	Unique object ID for the supervisor object.
AgentReference	STRING	String containing the Agent ID for the agent to be monitored.
SupervisorID	STRING	String containing the supervisor's AgentID.
SupervisorKey	STRING	Supervisor's unique object ID.
BargedInCallID	STRING	If the supervisor has barged in on the agent's call, the unique object ID of that call.
Supervisor AgentState	STRING	The supervisor's agent state.

Remarks

This is a Supervisor specific event. It is supported for use with Unified CCE only.

OnStopMonitoringAgent

The OnStopMonitoringAgent event is generated when monitoring of an agent is dropped in response to a StopMonitoringAgent() request.

Syntax

C++

```
void OnStopMonitoringAgent (Arguments& args)
```

COM

```
void OnStopMonitoringAgent (IArguments * args)
```

VB

```
session_OnStopMonitoringAgent (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 74: OnStopMonitoringAgent Parameters

Keyword	Type	Description
UniqueObjectID	STRING	Unique object ID for the supervisor object.
AgentReference	STRING	String containing the Agent ID for the agent to be monitored.
SupervisorID	STRING	String containing the supervisor's AgentID.
SupervisorKey	STRING	Supervisor's unique object ID.
BargedInCallID	STRING	If the supervisor has barged in on the agent's call, the unique object ID of that call.
Supervisor AgentState	STRING	The supervisor's agent state.

Remarks

This is a Supervisor specific event. It is supported for use with Unified CCE only.

OnUserMessageConf

Not supported.

ISkillGroupEvents Interface

The SkillGroup object fires events on the ISkillGroupEvents interface. The following events are published to subscribers of the ISkillGroupEvents interface.

OnSkillGroupStatisticsUpdated

The OnSkillGroupStatisticsUpdated event is generated when skill group statistics are reported. You can connfigure the update frequency of OnSkillGroupStatisticsUpdated on the CTI OS server (for more information, see *CTI OS System Manager's Guide for Cisco Unified ICM/Contact Center Enterprise & Hosted*).

Syntax

C++

```
void OnSkillGroupStatisticsUpdated (Arguments& args)
```

COM

```
void OnSkillGroupStatisticsUpdated (IArguments * args)
```

VB

```
skillgroup_ OnSkillGroupStatisticsUpdated (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 75: OnSkillGroupStatisticsUpdated Parameters

Keyword	Type	Description
PeripheralID	INT	The Unified ICM PeripheralID of the ACD on which the agent resides.
SkillGroupNumber	INT	The optional, user-defined number of the agent skill group as known to the peripheral. May contain the special value NULL_SKILL_GROUP when not available.
SkillGroupID	INT	The system-assigned identifier of the skill group. May contain the special value NULL_SKILL_GROUP when not available.

The statistics event also contains all the statistics fields listed in Table 1 in a nested Arguments array named STATISTICS.

OnSkillInfoEvent

Provides information about a particular skill group. This event is sent to any client that has enabled skill group statistics.

Syntax

C++

```
void OnSkillInfoEvent(Arguments& args)
```

COM

```
void OnSkillInfoEvent(IArguments * args)
```

VB

```
skillgroup_OnSkillInfoEvent(ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 76: OnSkillInfoEVent Parameters

Keyword	Type	Description
SkillGroupNumber	INT	Skill group number.
SkillGroupName	STRING	Skill group name associated with the skill group number above.

IButtonEnablementEvents

This interface allows a client application to receive events that indicate what buttons you can enable on the user interface, given the current agent and current call states.

OnButtonEnablementChange

The OnButtonEnablementChange event is received by a client in agent mode whenever CIL receives an agent or call event that carries the EnablementMask field in its parameters. This event allows the client application to enable or disable elements on the user interface. The fields in the event are the same as in OnButtonEnablementChange.

C++

```
void OnButtonEnablementChange (Arguments& args)
```

COM

```
void OnButtonEnablementChange (IArguments * args)
```

VB

```
session_ OnButtonEnablementChange (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 77: OnButtonEnablementChange Parameters

Keyword	Type	Description
EnablementMask	INT	Contains the bit-mask that specifies what buttons can be enabled or disabled when this call is the current call. For more information, see the table below.
UniqueObjectID	STRING	ID of the object (for example, agent, call) that the event is meant for.
MessageID	INT	The event that triggered the button enablement change.

Note The following table represents the C++/COM/VB enumerations. Enumerations for Java are in the description of CtiOs_Enums.ButtonEnablement in the Javadoc. Reference bits by the enumeration rather than the actual number in the bit mask.

Table 78: Table of Enablement Bits

Button	Bit Mask
DISABLE_ALL	0x00400000
ENABLE_ANSWER	0X00000001
ENABLE_RELEASE	0X00000002
ENABLE_HOLD	0X00000004
ENABLE_RETRIEVE	0X00000008
ENABLE_MAKECALL	0X00000010
ENABLE_TRANSFER_INIT	0X00000020
ENABLE_TRANSFER_COMPLETE	0X00000040
ENABLE_SINGLE_STEP_TRANSFER	0X00000080
ENABLE_CONFERENCE_INIT	0X00000100
ENABLE_CONFERENCE_COMPLETE	0X00000200
ENABLE_SINGLE_STEP_ CONFERENCE	0X00000400
ENABLE_ALTERNATE	0X00000800
ENABLE_RECONNECT	0X00001000

Button	Bit Mask
ENABLE_WRAPUP	0X00002000
ENABLE_INSIDE_MAKECALL	0X00004000
ENABLE_OUTSIDE_MAKECALL	0X00008000
ENABLE_SUPERVISOR_ASSIST	0X00010000
ENABLE_EMERGENCY_CALL	0X00020000
ENABLE_BAD_LINE_CALL	0X00040000
ENABLE_STATISTICS	0X00080000
ENABLE_CHAT	0X00100000
ENABLE_RECORD	0X00200000
ENABLE_LOGIN	0X01000000
ENABLE_LOGOUT	0X02000000
ENABLE_LOGOUT_WITH_REASON	0x04000000
ENABLE_READY	0X08000000
ENABLE_NOTREADY	0X10000000
ENABLE_NOTREADY_WITH_ REASON	0X20000000
ENABLE_WORKREADY	0X40000000
ENABLE_WORKNOTREADY	0x80000000
DISABLE_READY	0xF7FFFFFF
DISABLE_NOTREADY	0xCFFFFFFF
DISABLE_WORKREADY	0xBFFFFFFF
Supervisor Button Enablement Masks	
ENABLE_SET_AGENT_LOGOUT	0x00000001
ENABLE_SET_AGENT_READY	0x00000002
ENABLE_SILENTMONITOR	0x00000004
ENABLE_BARGE_IN	0x00000004
ENABLE_INTERCEPT	0x00000008
ENABLE_CLEAR	0x00000010
ENABLE_START_SILENTMONITOR	0x00000020

Button	Bit Mask
ENABLE_STOP_SILENTMONITOR	0x00000040
DISABLE_SET_AGENT_LOGOUT	0xFFFFFFFE
DISABLE_SET_AGENT_READY	0xFFFFFFFD
DISABLE_SILENTMONITOR	0xFFFFFFFB
DISABLE_BARGE_IN	0xFFFFFFFB
DISABLE_INTERCEPT	0xFFFFFFF7
DISABLE_CLEAR	0xFFFFFFEF
DISABLE_START_SILENTMONITOR	0xFFFFFFDF
DISABLE_STOP_SILENTMONITOR	0xFFFFFFBF
DISABLE_SUPERVISE_CALL	DISABLE_BARGE_IN & DISABLE_INTERCEPT & DISABLE_CLEAR & DISABLE_SILENTMONITOR & DISABLE_START_SILENTMONITOR & DISABLE_STOP_SILENTMONITOR
DISABLE_SET_AGENT_STATE	DISABLE_SET_AGENT_ LOGOUT, DISABLE_SET_ AGENT_READY
DISABLE_ALL_AGENT_SELECT	DISABLE_BARGE_IN & DISABLE_INTERCEPT & DISABLE_CLEAR & DISABLE_SILENTMONITOR & DISABLE_START_SILENTMONITOR & DISABLE_STOP_SILENTMONITOR

OnSupervisorButtonChange

A client in agent mode working as supervisor receives the OnSupervisorButtonChange event whenever CIL receives a Monitored Agent, Monitored call event that carries the SupervisorBtnEnablementMask field in its parameters. This event allows the client application to enable or disable elements on the user interface. The fields in the event are the same as in OnButtonEnablementChange.

C++

```
void OnSupervisorButtonChange (Arguments& args)
```

COM

```
void OnSupervisorButtonChange (IArguments * args)
```

VB

```
session_ OnSupervisorButtonChange (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 79: OnSupervisorButtonChange Parameters

Keyword	Type	Description
SupervisorBtn EnablementMask	INT	Contains the bit-mask that specifies what buttons can be enabled or disabled when this call is the current call. For more information, see Table 78: Table of Enablement Bits, on page 93.

Remarks

Supported for use with Unified CCE only.

IMonitoredAgentEvents Interface

Note The events in this section are supported for use with Unified CCE only.

This interface fires Agent events to a supervisor for his team members. IMonitoredAgentEvents are triggered by the supervisor sending a StartMonitoringAllAgentTeams request (for more information, see Agent Object). For more information about the event parameters, see the IAgentEvents interface.

The most common event handled is the OnMonitoredAgentStateChange event, which informs a supervisor of agent state changes of agents in the supervisor's team. All the parameters are the same as the regular OnAgentStateChange events, except for an additional keyword called CTIOS_MONITORED, which indicates that this event is for a monitored agent.

List of Monitored Agent events:

OnMonitoredAgentStateChange([in] IArguments *pIArguments);

OnMonitoredAgentInfoEvent([in] IArguments *pIArguments);

IMonitoredCallEvents Interface

Note The events in this section are supported with Unified CCE only.

This interface fires Call events to a supervisor for one of his agent team members. When the supervisor sends a StartMonitoringAgent request (for more information, see Agent Object), the supervisor starts receiving

MonitoredCallEvents for this “currently” monitored agent. Monitored call events are received until the supervisor sends a StopMonitoringAgent request for this agent.

The IMonitoredCallEvents interface includes OnMonitoredCallBegin, OnMonitoredCallEnd, and OnMonitoredCallDataUpdate as well as other call events (see list below). These events are described in detail for the ICallEventsInterface. The only difference is that the Arguments array contains an additional keyword call CTIOS_MONITORED, indicating that this event is for a monitored call.

List of Monitored Call events:

OnMonitoredCallBegin([in] IArguments *pIArguments);

OnMonitoredCallEnd([in] IArguments *pIArguments);

OnMonitoredCallDataUpdate([in] IArguments *pIArguments);

OnMonitoredCallDelivered([in] IArguments *pIArguments);

OnMonitoredCallEstablished([in] IArguments *pIArguments);

OnMonitoredCallHeld([in] IArguments *pIArguments);

OnMonitoredCallRetrieved([in] IArguments *pIArguments);

OnMonitoredCallCleared([in] IArguments *pIArguments);

OnMonitoredCallConnectionCleared([in] IArguments *pIArguments);

MonitoredCallReachedNetworkEvent([in] IArguments *pIArguments);

OnMonitoredCallOriginated([in] IArguments *pIArguments);

OnMonitoredCallFailed([in] IArguments *pIArguments);

OnMonitoredCallTransferred([in] IArguments *pIArguments);

OnMonitoredCallConferenced([in] IArguments *pIArguments);

OnMonitoredCallDiverted([in] IArguments *pIArguments);

OnMonitoredTranslationRoute([in] IArguments *pIArguments);

OnMonitoredCallAgentPrecallEvent([in] IArguments *pIArguments);

OnMonitoredCallAgentPrecallAbortEvent([in] IArguments *pIArguments);

MonitoredCallServiceInitiatedEvent([in] IArguments *pIArguments);

MonitoredCallQueuedEvent([in] IArguments *pIArguments);

MonitoredCallDequeuedEvent([in] IArguments *pIArguments);

ISilentMonitorEvents

The silent monitor manager object fires events on the ISilentMonitorEvents interface. The following events are published to subscribers of the ISilentMonitorEvents interface.

Note The events in this section are supported with Unified CCE only.

Note The following events apply only to CTI OS based silent monitor unless noted otherwise.

OnCallRTPStarted

The OnCallRTPStarted event indicates that an RTP media stream has started. This event accompanies the Call object in an Unified CCE environment.

Syntax

C++

```
void OnCallRTPStarted(Arguments& args)
```

COM

```
void OnCallRTPStarted (IArguments * args)
```

VB

```
session_OnCallRTPStarted (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 80: OnCallRTPStarted Parameters

Keyword	Type	Description
MonitorID	UINT	The Monitor ID of the device or call monitor that caused this message to be sent to the client, or zero if there is no monitor associated with the event (All Events Service).
PeripheralID	UINT	The Unified ICM PeripheralID of the ACD where the device is located.
ClientPort	UINT	The TCP/IP port number of the CTI Client connection.
Direction	USHORT	The direction of the event. One of the following values: 0: Input; 1: Output; 2: Bi-directional.

Keyword	Type	Description
RTPType	USHORT	The type of the event. One of the following values: 0: Audio; 1: Video; 2: Data.
BitRate	UINT	The media bit rate, used for g.723 payload only.
EchoCancellation	USHORT	on/off.
PacketSize	UINT	In milliseconds.
PayloadType	USHORT	The audio codec type.
ConnectionDevice IDType	USHORT	Indicates the type of the connection identifier supplied in the ConnectionDeviceID floating field.
ConnectionCallID	UINT	The Call ID value assigned to this call by the peripheral or Unified ICM.
Connection DeviceID	STRING	The identifier of the connection between the call and the device.
ClientAddress	STRING	The IP address of the phone.
AgentID (optional)	STRING	The agent's ACD login ID.
AgentExtension (optional)	STRING	The agent's ACD teleset extension.
AgentInstrument (optional)	STRING	The agent's ACD instrument number.

OnCallRTPStopped

The OnCallRTPStopped event indicates that an RTP media has stopped. This event accompanies the Call object in an Unified CCE environment.

Syntax

C++

```
void OnCallRTPStopped(Arguments& args)
```

COM

```
void OnCallRTPStopped (IArguments * args)
```

VB

```
session_OnCallRTPStopped (ByVal args As CtiosCLIENTLib.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 81: OnCallRTPStopped Parameters

Keyword	Type	Description
MonitorID	UINT	The Monitor ID of the device or call monitor that caused this message to be sent to the client, or zero if there is no monitor associated with the event (All Events Service).
PeripheralID	UINT	The Unified ICM PeripheralID of the ACD where the device is located.
ClientPort	UINT	The TCP/IP port number of the CTI Client connection.
Direction	USHORT	The direction of the event. One of the following values: 0: Input; 1: Output; 2: Bi-directional.
RTPType	USHORT	The type of the event. One of the following values: 0: Audio; 1: Video; 2: Data.
BitRate	UINT	The media bit rate, used for g.723 payload only.
EchoCancellation	USHORT	on/off.
PacketSize	UINT	In milliseconds.
PayloadType	USHORT	The audio codec type.
ConnectionDevice IDType	USHORT	Indicates the type of the connection identifier supplied in the ConnectionDeviceID floating field.
ConnectionCallID	UINT	The Call ID value assigned to this call by the peripheral or Unified ICM.

Keyword	Type	Description
Connection DeviceID	STRING	The identifier of the connection between the call and the device.
ClientAddress	STRING	The IP address of the phone.
AgentID (optional)	STRING	The agent's ACD login ID.
AgentExtension (optional)	STRING	The agent's ACD teleset extension.
AgentInstrument (optional)	STRING	The agent's ACD instrument number.

OnStartSilentMonitorConf

The OnStartSilentMonitorConf event is sent to the monitoring application to indicate that the CTI OS server has processed a StartSilentMonitorRequest.

Syntax

C++

```
void OnStartSilentMonitorConf (Arguments & args);
```

COM

```
HRESULT OnStartSilentMonitorConf ([in] Arguments* args);
```

VB

```
Session_ OnStartSilentMonitorConf (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 82: OnStartSilentMonitorConf Parameters

Keyword	Type	Description
MonitoredUniqueObjectID	STRING	Unique Object ID of the object being monitored.
AgentID	STRING	Agent ID of the agent to be monitored. This message contains either AgentID or DeviceID, but not both.
DeviceID	STRING	Device ID of the agent to be monitored. This message contains either AgentID or DeviceID, but not both.

Keyword	Type	Description
PeripheralID	INT	The Unified ICM PeripheralID of the ACD where the silent monitor start was requested.
MonitoringIPAddress	STRING	TCP/IP address of the monitoring application.
MonitoringIPPort	INT	TCP/IP port of the monitoring application.
SMSessionKey	UNSIGNED SHORT	Unique identifier for the Silent Monitor Session.
HeartbeatInterval	INT	Heartbeat interval for the silent monitor session.
HeartbeatTimeout	INT	Timeout for no activity.
OriginatingServerID	STRING	TCP/IP Address:Port of the CTI OS server from which the request originated.
OriginatingClientID	STRING	Client Identification of the monitoring application.

OnSilentMonitorStartedEvent

For CTI OS Based Silent Monitor

The OnSilentMonitorStartedEvent event is fired to the subscriber to indicate that a silent monitor session has started on its behalf and that audio transmission to the monitoring client has started.

Syntax

C++

```
void OnSilentMonitorStartedEvent(Arguments & args);
```

COM

```
HRESULT OnSilentMonitorStartedEvent([in] Arguments* args);
```

VB

```
Session_ OnSilentMonitorStartedEvent(ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 83: OnSilentMonitorStartedEvent Parameters

Keyword	Type	Description
MonitoredUniqueObjectID	STRING	Unique Object ID of the object being monitored.
AgentID	STRING	Agent ID of the agent to be monitored. This message contains either AgentID or DeviceID, but not both.
DeviceID	STRING	Device ID of the agent to be monitored. This message contains either AgentID or DeviceID, but not both.
PeripheralID	INT	The Unified ICM PeripheralID of the ACD where silent monitoring started.
MonitoringIPAddress	STRING	TCP/IP address of the monitoring application.
MonitoringIPPort	INT	TCP/IP port of the monitoring application.
SMSessionKey	UNSIGNED SHORT	Unique identifier for the Silent Monitor Session.
HeartbeatInterval	INT	Heartbeat interval for the silent monitor session.
HeartbeatTimeout	INT	Timeout for no activity.
OriginatingServerID	STRING	TCP/IP Address: Port of the CTI OS server from which the request originated.
OriginatingClientID	STRING	Client Identification of the monitoring application.

For CCM-Based Silent Monitor

When you configure CCM based silent monitor, this event tells the monitored application, for example an agent desktop, that it is being monitored. This event, in addition to call events for the silent monitor call, tells the monitoring application, for example a supervisor desktop, that silent monitor of the agent has begun.

Note At failover, the desktop can receive multiple OnSilentMonitorStartedEvents.

Syntax

C++

```
void OnSilentMonitorStartedEvent(Arguments & args);
```

COM

```
HRESULT OnSilentMonitorStartedEvent([in] Arguments* args);
```

VB

```
Session_ OnSilentMonitorStartedEvent(ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 84: OnSilentMonitorStartedEvent

Keyword	Type	Description
SilentMonitorInitiatingAgentUID	STRING	Unique object ID of the agent that initiated silent monitor.
SilentMonitorInitiatingDeviceID	STRING	ID of the device that initiated silent monitor.
SilentMonitorTargetAgentUID	STRING	Unique object ID of the silently monitored agent.
SilentMonitorTargetDeviceID	STRING	ID of the silently monitored device.
SilentMonitorCallUID	STRING	Unique object ID of the silent monitor call.

OnSilentMonitorStartRequestedEvent

The OnSilentMonitorStartRequestedEvent event is fired to the subscriber to indicate that a silent monitor session request has arrived and that it will be established on its behalf if the DoDefaultMessageHandling parameter is set to True. The default behavior is to start sending audio and establish the session automatically. If the subscriber wishes to process the event by itself, they must set the DoDefaultMessageHandling parameter to False and invoke AcceptSilentMonitoring when it is ready to start the session and call ReportSMSessionStatus to the monitoring client.

CTI OS server generates this event whenever a remote application calls the StartSilentMonitorRequest method.

Syntax

C++

```
void OnSilentMonitorStartRequestedEvent(Arguments & args);
```

COM

```
HRESULT OnSilentMonitorStartRequestedEvent([in] Arguments* args);
```

VB

```
Session_ OnSilentMonitorStartRequestedEvent(ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 85: OnSilentMonitorStartRequestedEvent Parameters

Keyword	Type	Description
MonitoredUniqueObjectID	STRING	Unique Object ID of the object being monitored.
AgentID	STRING	Agent ID of the agent to be monitored. This message contains either AgentID or DeviceID, but not both.
DeviceID	STRING	Device ID of the agent to be monitored. This message contains either AgentID or DeviceID, but not both.
PeripheralID	INT	The Unified ICM PeripheralID of the ACD where the silent monitor start was requested.
MonitoringIPAddress	STRING	TCP/IP address of the monitoring application.
MonitoringIPPort	INT	TCP/IP port of the monitoring application.
SMSessionKey	UNSIGNED SHORT	Unique identifier for the Silent Monitor Session.
HeartbeatInterval	INT	Heartbeat interval for the silent monitor session.
HeartbeatTimeout	INT	Timeout for no activity.
OriginatingServerID	STRING	TCP/IP Address: Port of the CTI OS server from which the request originated.
OriginatingClientID	STRING	Client Identification of the monitoring application.

Keyword	Type	Description
DoDefaultMessage Handling	BOOLEAN	When this parameter is set to True, it instructs the SilentMonitorManager to immediately start sending audio and establish the silent monitor session. If this parameter is set to False, it instructs the SilentMonitorManager to not send voice and to not establish the silent monitor session. In this case, it is the responsibility of the subscriber to report this status accordingly.

OnSilentMonitorSessionDisconnected

The OnSilentMonitorSessionDisconnected event is sent to the application to report errors if the connection fails between the monitoring and monitored clients.

Syntax

C++

```
void OnSilentMonitorSessionDisconnected (Arguments & args);
```

COM

```
HRESULT OnSilentMonitorSessionDisconnected ([in] Arguments* args);
```

VB

```
Session_ OnSilentMonitorSessionDisconnected (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 86: OnSilentMonitorSessionDisconnected Parameters

Keyword	Type	Description
MonitoredUniqueObjectID	STRING	Unique Object ID of the object being monitored.
SMSessionKey	UNSIGNED SHORT	Unique identifier for the Silent Monitor Session.
StatusCode	SHORT	One of the ISilentMonitorEvent status codes in Table 90: ISilentMonitorEvent Status Codes, on page 110.

OnSilentMonitorStopRequestedEvent

For CTI OS Based Silent Monitor

The OnSilentMonitorStopRequestedEvent event is fired to the subscriber to indicate that a silent monitor session was stopped on their behalf. CTI OS server generates this event whenever a remote application calls the StopSilentMonitorRequest method.

Syntax

C++

```
void OnSilentMonitorStopRequestedEvent(Arguments & args);
```

COM

```
HRESULT OnSilentMonitorStopRequestedEvent([in] Arguments* args);
```

VB

```
Session_ OnSilentMonitorStopRequestedEvent(ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 87: OnSilentMonitorStopRequestedEvent Parameters

Keyword	Type	Description
MonitoredUniqueObjectID	STRING	Unique Object ID of the object being monitored.
AgentID	STRING	Agent ID of the agent to be monitored. This message contains either AgentID or DeviceID, but not both.
DeviceID	STRING	Device ID of the agent to be monitored. This message contains either AgentID or DeviceID, but not both.
PeripheralID	INT	The Unified ICM PeripheralID of the ACD where silent monitoring has stopped.
MonitoringIPAddress	STRING	TCP/IP address of the monitoring application.
SMSessionKey	UNSIGNED SHORT	Unique identifier for the Silent Monitor Session.

Keyword	Type	Description
OriginatingServerID	STRING	TCP/IP Address:Port of the CTI OS server from which the request originated.
OriginatingClientID	STRING	Client Identification of the monitoring application.

For CCM-Based Silent Monitor

When CCM based silent monitor is configured this event tells the monitored application, for example an agent desktop, that it is no longer being monitored. This event in addition to call events for the silent monitor call tells the monitoring application, for example a supervisor desktop, that silent monitor of the agent has ended.

If an error occurs, the Disposition field is set to the error returned in OnControlFailure.

Syntax

C++

```
void OnSilentMonitorStopRequestedEvent(Arguments & args);
```

COM

```
HRESULT OnSilentMonitorStopRequestedEvent([in] Arguments* args);
```

VB

```
Session_ OnSilentMonitorStopRequestedEvent(ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 88: OnSilentMonitorStopRequestedEvent Parameters

Keyword	Type	Description
SilentMonitorInitiatingAgentUID	STRING	Unique object ID of the agent that initiated silent monitor.
SilentMonitorInitiatingDeviceID	STRING	ID of the device that initiated silent monitor.
SilentMonitorTargetAgentUID	STRING	Unique object ID of the silently monitored agent.
SilentMonitorTargetDeviceID	STRING	ID of the silently monitored device.
SilentMonitorCallUID	STRING	Unique object ID of the silent monitor call.

Keyword	Type	Description
SilentMonitorCallDisposition	unsigned int	If the silent monitor session failed, the event cause carried by the call failed event is stored here. If the silent monitor session was either terminated by the supervisor or the agent's call ended, this field is set to 0.

OnSilentMonitorStatusReportEvent

The OnSilentMonitorStatusReportEvent event indicates a change in status of a silent monitor session. This event is sent only to the monitoring application.

Syntax

C++

```
void OnSilentMonitorStatusReportEvent (Arguments & args);
```

COM

```
HRESULT OnSilentMonitorStatusReportEvent ([in] Arguments* args);
```

VB

```
Session_ OnSilentMonitorStatusReportEvent (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 89: OnSilentMonitorStatusReportEvent Parameters

Keyword	Type	Description
MonitoredUniqueObjectID	STRING	Unique Object ID of the object being monitored.
SMSessionKey	UNSIGNED SHORT	Unique identifier for the Silent Monitor Session.
StatusCode	SHORT	One of the ISilentMonitorEvent status codes in Table 90: ISilentMonitorEvent Status Codes, on page 110.
OriginatingServerID	STRING	TCP/IP Address:Port of the CTI OS server from which the request originated.
OriginatingClientID	STRING	Client Identification of the monitoring application.

Keyword	Type	Description
TargetCILClientID	STRING	CIL Client ID of the monitoring application.

Table 90: ISilentMonitorEvent Status Codes

enum Value	Numeric Value (Hex)
General Codes	
eSMStatusUnknown	-1
eSMStatusOK	0
eSMStatusFailed	0x00000001
eSMStatusComError	0x00000002
eSMStatusMonitorStarted	0x00000003
eSMStatusMonitorStopped	0x00000004
eSMStatusHeartbeatTimeout	0x00000005
eSMStatusOutOfMemory	0x00000006
eSMStatusPortUnavailable	0x00000007
eSMStatusIncorrectStateForThisAction	0x00000008
eSMStatusResourceError	0x00000009
eSMStatusRejectedBadParameter	0x0000000A
eSMStatusWinsockError	0x0000000B
eSMStatusMediaTerminationNotPresent	0x0000000C
eSMStatusIPPhoneInformatioNotAvailable	0x0000000D
eSMStatusMissingParameter	0x0000000E
eSMStatusSessionNotFound	0x0000000F
eSMStatusSessionAlreadyExists	0x00000010
eSMStatusDisconnected	0x00000011
eSMStatusInvalidStateForAction	0x00000012
eSMStatusInProgress	0x00000013
eSMStatusMaxSessionsExceeded	0x00000014
eSMStatusCCMSilentMonitor	0x00000015

enum Value	Numeric Value (Hex)
Silent Monitor Session Codes	
eSMStatusSessionTerminatedAbnormally	0x10000000
eSMStatusRejectedAlreadyInSession	0x10000001
eSMStatusRejectedWinPcapNotPresent	0x10000002
eSMStatusWinPcapError	0x10000003
eSMStatusMediaUnknownCodec	0x10000004
eSMStatusIncorrectSessionMode	0x10000005
eSMStatusPeerSilentMonitorNotEnabled	0x10000006
eSMStatusSilentMonitorNotEnabled	0x10000007
eSMStatusNoResponseFromPeer	0x10000008
eSMStatusPeerLoggedOut	0x10000009
eSMStatusSessionTerminatedByMonitoredClient	0x1000000A
eSMStatusSessionTerminatedByMonitoringClient	0x1000000B
eSMStatusNoRTPPacketsReceivedFormIPPhone	0x1000000C
eSMStatusSessionConnectionToDelegateLost	0x1000000D
eSMStatusMTError	0x20000000
Voice Capture-Specific Codes	
eSMStatusWPNoPacketsReceived	0x30000000
eSMStatusWPFailedToOpenDevice	0x30000001
eSMStatusWPFailedToSetFilterExp	0x30000002
eSMStatusWPErrorInFilterExp	0x30000003

OnStopSilentMonitorConf

This OnStopSilentMonitorConf event is sent to the monitoring application to indicate that the CTI OS server has processed a StopSilentMonitorRequest.

Syntax

C++

```
void OnStopSilentMonitorConf (Arguments & args);
```

COM

```
HRESULT OnStopSilentMonitorConf ([in] Arguments* args);
```

VB

```
Session_ OnStopSilentMonitorConf (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 91: OnStopSilentMonitorConf Parameters

Keyword	Type	Description
MonitoredUniqueObjectID	STRING	Unique Object ID of the object being monitored.
AgentID	STRING	Agent ID of the agent to be monitored. This message contains either AgentID or DeviceID, but not both.
DeviceID	STRING	Device ID of the agent to be monitored. This message contains either AgentID or DeviceID, but not both.
PeripheralID	INT	The Unified ICM PeripheralID of the ACD where the silent monitor start was requested.
MonitoringIPAddress	STRING	TCP/IP address of the monitoring application.
MonitoringIPPort	INT	TCP/IP port of the monitoring application.
SMSessionKey	UNSIGNED SHORT	Unique identifier for the Silent Monitor Session.
HeartbeatInterval	INT	Heartbeat interval for the silent monitor session.
HeartbeatTimeout	INT	Timeout for no activity.
OriginatingServerID	STRING	TCP/IP Address:Port of the CTI OS server from which the request originated.
OriginatingClientID	STRING	Client Identification of the monitoring application.

Keyword	Type	Description

MonitoredUniqueObjectID	STRING	Unique Object ID of the object being monitored.
AgentID	STRING	Agent ID of the agent who was monitored. This message contains either AgentID or DeviceID, but not both.
DeviceID	STRING	Device ID of the agent who was monitored. This message contains either AgentID or DeviceID, but not both.
PeripheralID	INT	The Unified ICM PeripheralID of the ACD where silent monitoring has stopped.
MonitoringIPAddress	STRING	TCP/IP address of the monitoring application.
SMSessionKey	UNSIGNED SHORT	Unique identifier for the Silent Monitor Session.
OriginatingServerID	STRING	TCP/IP Address:Port of the CTI OS server from which the request originated.
OriginatingClientID	STRING	Client Identification of the monitoring application.

OnRTPStreamTimedoutEvent

The OnRTPStreamTimedoutEvent event is sent to the monitored application to report that no RTP voice packets were received from the monitored IP Phone.

Syntax

C++

```
void OnRTPStreamTimedoutEvent (Arguments & args);
```

COM

```
HRESULT OnRTPStreamTimedoutEvent ([in] Arguments* args);
```

VB

```
Session_ OnRTPStreamTimedoutEvent (ByVal args as CTIOSCLIENTLIB.IArguments)
```

Parameters

args

Arguments array containing the following fields.

Table 92: OnRTPStreamlinedEvent Parameters

Keyword	Type	Description
MonitoredUniqueObjectID	STRING	Unique Object ID of the object being monitored.
SMSessionKey	UNSIGNED SHORT	Unique identifier for the Silent Monitor Session.

Keyword	Type	Description
StatusCode	SHORT	One of the ISilentMonitorEvent status codes in Table 90: ISilentMonitorEvent Status Codes, on page 110.

IGenericEvents Interface

The IGenericEvents interface receives Generic events. Unlike other interfaces that have a callback method for each event, the IGenericEvents interface has one method that passes the CtiOs_Enums.EventID code and the Arguments for the event.

OnEvent

Passes the eventID code and arguments for generic events received by the IGenericEvents interface.

Syntax

Java

```
void OnEvent(int iEventID, Arguments rArgs
```

.NET

```
void OnEvent(int iEventID, Cisco.CtiOs.Cil.EventPublisher.EventPublisherEventArgs args)
```

Java Adapter Classes

The CTI OS Java CIL contains the same adapter classes as the C++ CIL plus the LogEventsAdapter class. This class provides the default implementation for the message handlers in ILogEvents.

This section lists the methods available in the CTI OS Java CIL for event subscription and unsubscription.

IAllInOne

The following methods subscribe and unsubscribe the CTI OS Session Object for the IAllInOne interface:

Methods

- `int addAllInOneEventListener(IAllInOne allInOneEvents)`
- `int removeAllInOneEventListener(IAllInOne allInOneEvents)`

IAgentEvents

The following methods subscribe and unsubscribe the CTI OS Session Object for the IAgentEventsinterface:

Methods

- `int addAgentEventListener(IAgentEvents agentEvents)`

- `int removeAgentEventListener(IAgentEvents agentEvents)`

IButtonEnablementEvents

The following methods subscribe and unsubscribe the CTI OS Session Object for the IButtonEnablementEvents interface:

Methods

- `int addButtonEnablementEventListener(IButtonEnablementEvents buttonEvents)`
- `int removeButtonEnablementEventListener(IButtonEnablementEvents buttonEvents)`

ICallEvents

The following methods subscribe and unsubscribe the CTI OS Session Object for the ICallEvents interface:

Methods

- `int addCallEventListener (ICallEvents callEvents)`
- `int removeCallEventListener (ICallEvents callEvents)`

ISkillGroupEvents

The following methods subscribe and unsubscribe the CTI OS Session Object for the ISkillGrouEvents interface:

Methods

- `int addSkillGroupEventListener (ISkillGroupEvents skillGroupEvents)`
- `int removeSkillGroupEventListener (ISkillGroupEvents skillGroupEvents)`

Events in Java CIL

To subscribe for events in the Java CIL, use the AddEventListener method. This method has the following syntax:

```
int AddEventListener(IGenericEvents Listener, int iListID)
```

where Listener is the IGenericEvents object that subscribes for events and iListID is the ID of the subscriber list to add this listener to. Java subscriber list IDs are part of the CtiOs_Enums.SubscriberList interface; each C++/COM/VB event interface has a corresponding Java subscriber list (for example, C++/COM/VB ISessionEvents corresponds to Java eSessionList). For more information about the CtiOs_Enums.SubscriberList interface, see the Javadoc file.

The IGenericEvents interface, though it contains the C++/COM/VB events documented in this chapter, does not have a callback method for each event. Instead, the OnEvent method passes the event ID code and arguments for each event. The OnEvent method has the following syntax:

```
void OnEvent(int iEventID, Arguments rArgs)
```

where iEventID is the event ID code for the event and rArgs is the arguments for the event. The arguments for each Java event are the same as for the corresponding C++/COM/VB event. For more information about the IGenericEvents interface, see the Javadoc file.

To unsubscribe for events in the Java CIL, use the RemoveEventListener method. This method has the following syntax:

```
int RemoveEventListener(IGenericEvents Listener, int iListID)
```

where Listener is the IGenericEvents object that is unsubscribing for events and iListID is the ID of the subscriber list to remove this listener from.

Events in .NET CIL

To subscribe for events in the .NET CIL, use the AddEventListener method. This method has the following syntax:

```
CilError AddEventListener(IGenericEvents Listener, int iListID)
```

where Listener is the IGenericEvents object that subscribes for events and iListID is the ID of the subscriber list to add this listener to. Subscriber list IDs for .NET are part of the CtiOs_Enums.SubscriberList interface; each C++/COM/VB event interface has a corresponding .NET subscriber list (for example, C++/COM/VB ISessionEvents corresponds to .NET eSessionList).

The IGenericEvents interface, though it contains the C++/COM/VB events documented in this chapter, does not have a callback method for each event. Instead, the OnEvent method passes the event ID code and arguments for each event. The OnEvent method has the following syntax:

```
void OnEvent(Object sender, Cisco.CtiOs.Cil.EventPublisher.EventPublisherEventArgs
 eventArgs)
```

where, sender is a null object and eventArgs contains the eventID and arguments for the event. The arguments for each .NET event are the same as for the corresponding C++/COM/VB event.

The EventPublisherEventArgs class is a data type that defines the information passed to receivers of the event. This information includes the event ID and an Arguments array containing the arguments for the event. Therefore, event handling code must extract the event arguments from the EventPublisherEventArgs object as shown in the following sample code snippet, which uses the .NET CIL:

```
Arguments args = eventArgs.rArgs;EventID receivedEvent = (EventID)
eventArgs.iEventID;
switch(receivedEvent)
{
case EventID.eQueryAgentStatisticsConf:
ProcessQueryConf(args);
break;
...
}
```

To unsubscribe for events in the .NET CIL, use the RemoveEventListener method.

This method has the following syntax:

```
CilError RemoveEventListener(IGenericEvents Listener, int iListID)
```

where Listener is the IGenericEvents object that is unsubscribing for events and iListID is the ID of the subscriber list from which to remove this listener.

Event Parameters

Amount of Nonessential Call Object Parameters

The MinimizeEventArgs registry value controls the amount of nonessential Call object parameters that are sent to the client. When MinimizeEventArgs is set to 1, a minimal set of nonessential Call object parameters are sent to the CTI OS Client. When the MinimizeEventArgs registry value is set to 0, the CTI OS server sends to CTI OS Clients the event parameters listed in Table 6-90.

The MinimizeEventArgs value is located under the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems,
Inc.\Ctios\<Customer-Instancename>\CTIOS1\Server\CallObject
```

Table 93: MinimizeEventArgs Event Parameters

Event Name	Parameters
eCallRetrievedEvent	CTIOS_RETRIEVINGDEVICEID CTIOS_RETRIEVINGDEVICEIDFULL CTIOS_ENABLEMENTMASK CTIOS_ICMENTERPRISEUNIQUEID CTIOS_UNIQUEOBJECTID CTIOS_DEVICEUNIQUEOBJECTID CTIOS_CALLSTATUS* CTIOS_FILTERTARGET**
eCallHeldEvent	CTIOS_HOLDINGDEVICEID CTIOS_HOLDINGDEVICEIDFULL CTIOS_ENABLEMENTMASK CTIOS_ICMENTERPRISEUNIQUEID CTIOS_UNIQUEOBJECTID CTIOS_DEVICEUNIQUEOBJECTID CTIOS_FILTERTARGET** CTIOS_CALLSTATUS*

Event Name	Parameters
eCallConnectionClearedEvent	CTIOS_RELEASINGDEVICEID CTIOS_RELEASINGDEVICEIDFULL CTIOS_ENABLEMENTMASK CTIOS_ICMENTERPRISEUNIQUEID CTIOS_UNIQUEOBJECTID CTIOS_DEVICEUNIQUEOBJECTID CTIOS_FILTERTARGET** CTIOS_CALLSTATUS*
eCallTransferredEvent	CTIOS_PRIMARYCALLID CTIOS_SECONDARYCALLID CTIOS_TRANSFERRINGDEVICEID CTIOS_TRANSFERRINGDEVICEIDFULL CTIOS_TRANSFERREDDEVICEID CTIOS_TRANSFERREDDEVICEIDFULL CTIOS_NUMPARTIES ConnectedParty[PartyNumber] CTIOS_ISTRANSFERCONTROLLER GenerateCallDataUpdateArgs()*** CTIOS_ENABLEMENTMASK CTIOS_ICMENTERPRISEUNIQUEID CTIOS_UNIQUEOBJECTID CTIOS_DEVICEUNIQUEOBJECTID CTIOS_FILTERTARGET** CTIOS_CALLSTATUS*

Event Name	Parameters
eCallConferencedEvent	CTIOS_PRIMARYCALLID CTIOS_SECONDARYCALLID CTIOS_CONTROLLERDEVICEID CTIOS_CONTROLLERDEVICEIDFULL CTIOS_ADDEDPARTYDEVICEID CTIOS_ADDEDPARTYDEVICEIDFULL CTIOS_PRIMARYDEVICEID CTIOS_PRIMARYDEVICEIDFULL CTIOS_SECONDARYDEVICEID CTIOS_SECONDARYDEVICEIDFULL CTIOS_NUMPARTIES ConnectedParty[PartyNumber] GenerateCallDataUpdateArgs()*** CTIOS_ENABLEMENTMASK CTIOS_ICMENTERPRISEUNIQUEID CTIOS_UNIQUEOBJECTID CTIOS_DEVICEUNIQUEOBJECTID CTIOS_FILTERTARGET** CTIOS_CALLSTATUS*
eCallBeginEvent, eCallDataUpdateEvent	GenerateCallDataUpdateArgs()*** CTIOS_DEVICEID CTIOS_DIVERTINGDEVICEID CTIOS_DIVERTINGDEVICEIDFULL CTIOS_ENABLEMENTMASK CTIOS_ICMENTERPRISEUNIQUEID CTIOS_UNIQUEOBJECTID CTIOS_DEVICEUNIQUEOBJECTID CTIOS_FILTERTARGET** CTIOS_CALLSTATUS*

Event Name	Parameters
eCallDivertedEvent	GenerateCallDataUpdateArgs()*** CTIOS_DIVERTINGDEVICEID CTIOS_DIVERTINGDEVICEIDFULL CTIOS_ENABLEMENTMASK CTIOS_ICMENTERPRISEUNIQUEID CTIOS_UNIQUEOBJECTID CTIOS_DEVICEUNIQUEOBJECTID CTIOS_FILTERTARGET** CTIOS_CALLSTATUS*
eSnapshotCallConf	Includes all the parameters except for: CTIOS_ICMENTERPRISEUNIQUEID CTIOS_CALLCONNECTIONCALLID CTIOS_CALLCONNECTIONDEVICEIDTYPE CTIOS_CALLCONNECTIONDEVICEID CTIOS_CALLDEVICECONNECTIONSTATE CTIOS_CALLDEVICETYPE

Event Name	Parameters
eCallEstablishedEvent	CTIOS_ANSWERINGDEVICEID CTIOS_ANSWERINGDEVICEIDFULL CTIOS_CALLINGDEVICEID CTIOS_CALLINGDEVICEIDFULL CTIOS_CALLEDDEVICEID CTIOS_CALLEDDEVICEIDFULL CTIOS_SKILLGROUPID CTIOS_SKILLGROUPNUMBER CTIOS_SKILLGROUPPRIORITY CTIOS_SERVICEID CTIOS_SERVICENUMBER CTIOS_LINETYPE CTIOS_MEASUREDCALLQTIME CTIOS_CAMPAIGNID CTIOS_QUERYRULEID CTIOS_ENABLEMENTMASK CTIOS_ICMENTERPRISEUNIQUEID CTIOS_UNIQUEOBJECTID CTIOS_DEVICEUNIQUEOBJECTID CTIOS_FILTERTARGET** CTIOS_CALLSTATUS*

Event Name	Parameters
eCallDeliveredEvent	CTIOS_ALERTINGDEVICEID CTIOS_ALERTINGDEVICEIDFULL CTIOS_CALLINGDEVICEID CTIOS_CALLEDDEVICEID CTIOS_CALLINGDEVICEIDFULL CTIOS_CALLEDDEVICEIDFULL CTIOS_SKILLGROUPID CTIOS_SKILLGROUPNUMBER CTIOS_SKILLGROUPPRIORITY CTIOS_SERVICEID CTIOS_SERVICENUMBER CTIOS_LINETYPE CTIOS_MEASUREDCALLQTIME CTIOS_CAMPAIGNID CTIOS_QUERYRULEID CTIOS_ENABLEMENTMASK CTIOS_ICMENTERPRISEUNIQUEID CTIOS_UNIQUEOBJECTID CTIOS_DEVICEUNIQUEOBJECTID CTIOS_FILTERTARGET** CTIOS_CALLSTATUS*

Event Name	Parameters
eCallServiceInitiatedEvent, eCallOriginatedEvent, eCallQueuedEvent, eCallDequeuedEvent	CTIOS_CALLINGDEVICEIDFULL CTIOS_CALLEDDEVICEIDFULL CTIOS_CALLINGDEVICEID CTIOS_CALLEDDEVICEID CTIOS_SKILLGROUPID CTIOS_SKILLGROUPNUMBER CTIOS_SKILLGROUPPRIORITY CTIOS_SERVICEID CTIOS_SERVICENUMBER CTIOS_LINETYPE CTIOS_MEASUREDCALLQTIME CTIOS_CAMPAIGNID CTIOS_QUERYRULEID CTIOS_ENABLEMENTMASK CTIOS_ICMENTERPRISEUNIQUEID CTIOS_UNIQUEOBJECTID CTIOS_DEVICEUNIQUEOBJECTID CTIOS_FILTERTARGET** CTIOS_CALLSTATUS*
eControlFailureConf	CTIOS_PERIPHERALERRORCODE CTIOS_ERRORMESSAGE CTIOS_FAILURECODE CTIOS_ENABLEMENTMASK CTIOS_ICMENTERPRISEUNIQUEID CTIOS_UNIQUEOBJECTID CTIOS_DEVICEUNIQUEOBJECTID CTIOS_FILTERTARGET** CTIOS_CALLSTATUS*

Event Name	Parameters
eFailureConf, eFailureEvent, eCallFailedEvent	CTIOS_ERRORMESSAGE CTIOS_FAILURECODE CTIOS_ENABLEMENTMASK CTIOS_ICMENTERPRISEUNIQUEID CTIOS_UNIQUEOBJECTID CTIOS_DEVICEUNIQUEOBJECTID CTIOS_FILTERTARGET** CTIOS_CALLSTATUS*
eCallEndEvent	CTIOS_DEVICEID CTIOS_ENABLEMENTMASK CTIOS_ICMENTERPRISEUNIQUEID CTIOS_UNIQUEOBJECTID CTIOS_DEVICEUNIQUEOBJECTID CTIOS_FILTERTARGET** CTIOS_CALLSTATUS*

* If the eCallFailedEvent notification is received, the CTIOS_CALLSTATUS parameter is not added to any more events for the call ID specified in the eCallFailedEvent.

** If there is an agent on the device, then CTIOS_FILTERTARGET is added to all events listed in table 6-90.

*** The GenerateCallDataUpdateArgs() method adds the following parameters to the event:

CTIOS_PERIPHERALID, CTIOS_PERIPHERALTYPE, CTIOS_CALLTYPE, CTIOS_UNIQUEOBJECTID, CTIOS_ROUTERCALLKEYDAY, CTIOS_ROUTERCALLKEYCALLID, CTIOS_CONNECTIONCALLID, CTIOS_ANI, CTIOS_USERTOUSERINFO, CTIOS_DNIS, CTIOS_DIALEDNUMBER, CTIOS_CALLERENTEREDDIGITS, CTIOS_SERVICENUMBER, CTIOS_SERVICEID, CTIOS_SKILLGROUPNUMBER, CTIOS_SKILLGROUPPRIORITY, CTIOS_CALLWRAPUPDATA, CTIOS_CAMPAIGNID, CTIOS_QUERYRULEID, CTIOS_CALLVARIABLE1, CTIOS_CALLVARIABLE2, CTIOS_CALLVARIABLE3, CTIOS_CALLVARIABLE4, CTIOS_CALLVARIABLE5, CTIOS_CALLVARIABLE6, CTIOS_CALLVARIABLE7, CTIOS_CALLVARIABLE8, CTIOS_CALLVARIABLE9, CTIOS_CALLVARIABLE10, CTIOS_CUSTOMERPHONENUMBER, CTIOS_CUSTOMERACCOUNTNUMBER, CTIOS_NUMNAMEDVARIABLES, CTIOS_NUMNAMEDARRAYS, CTIOS_ECC, CTIOS_CTICLIENTS