Agent Object

- Agent Object, on page 1
- Agent Object Properties, on page 1
- Agent Statistics, on page 3
- Methods, on page 10

Agent Object

The Agent object provides developers using the CTI OS Client Interface Library with an interface to agent behavior. The Agent object exposes methods to perform all agent behaviors, such as logging in and setting the agent state.

The object stores specific agent information as properties, including the AgentID, AgentPassword, AgentInstrument, AgentExtension, and SkillGroups. When the agent is logged in to an ACD, the Agent object receives updates through AgentStateEvents and Agent Statistics updates.

You can use the Agent object in two different modes:

- In Agent Mode, the application creates an Agent object and informs the Session about the agent using Session.SetAgent().
- In Monitor Mode, the client application sets a message filter, and if the event stream involves events for Agent objects, those objects are dynamically created at the CIL as needed.

Agent Object Properties

The following table lists the agent object properties.

Note The data type listed for each keyword is the standardized data type discussed in CTI OS CIL data types in Chapter Three. For more information about the appropriate language specific types for these keywords Table 1.

Table 1: Agent Object Properties

Keyword	Type	Description
AgentAvailability Status	INT	One of the following values: UNKNOWN (-1), NOT AVAILABLE (0), ICM AVAILABLE (1), or APPLICATION AVAILABLE (2).
Agent CallMode	INT	A value that indicates the agent's call mode. Valid values are call-by-call (3) and nailed-up (4).
AgentExtension	STRING*	Extension associated by ACD to agent.
AgentID	STRING*	Can be set prior to login or after logout.
AgentInstrument	STRING*	Instrument associated by ACD to agent.
AgentRemote Number	STRING	The phone number that the agent uses for remote login.
AgentState	SHORT	One of the values in Table 2 representing the current state of the associated agent.
ClassIdentifier	INT	Identifies the type of this object.
SilentMonitorCallUID	STRING	The unique object ID of the silent monitor call. This is the call that results from calling SuperviseCall() with the SupervisorAction set to eSupervisorMonitor. **Note** Only applies to Cisco Unified Communications Manager based silent monitor.
SilentMonitorTargetAgentUID	STRING	This property contains the unique object ID of the agent who the supervisor is currently silent monitoring. **Note** Only applies to Cisco Unified Communications Manager based silent monitor.

Keyword	Type	Description
Extension		Extension associated by ACD to agent.
CurrentConnection Profile	STRING	The last selected agent connection profile.
IsSupervisor	INT	Indicates whether this agent is a supervisor.
LastError	INT	Last error code, if any. Otherwise this value is 0.
PeripheralID	INT	ID of peripheral.
PeripheralType	INT	The type of the peripheral.
Statistics	ARGUMENTS	An Arguments array containing the statistics listed in Table 2: Agent Statistics, on page 3.

*The CTI OS server imposes no restriction on the maximum length of this string. However, such restrictions are generally imposed by your switch/ACD and Cisco CTI Server. For more information about length restrictions for this string, see the documentation for the switch/ACD or CTI Server.

Agent Statistics

You can access statistics by first using GetValueArray on the Agent object to obtain the “Statistics” Arguments array and then using GetValueInt on the “Statistics” arguments array to obtain the desired value:

```
' First get the statistics argumentsDim args As Arguments
args = agent.GetValueArray ("Statistics")

' Then get the desired statistics
Dim availTimeSession As Integer
Dim loggedOnTimeSession As Integer
availTimeSession = args.GetValueInt("AvailTimeSession")
bargeInCallsToday = args.GetValueInt("BargeInCallsToday")
```

Note Not all the statistics values listed in the following table are present in every system configuration. Whether or not a particular statistic value is available depends both on the protocol version of CTI Server with which CTI OS connects and on the peripheral on which the agent resides.

Table 2: Agent Statistics

Statistic	Definition
AvailTime Session	Total time, in seconds, the agent was in the Available state for any skill group.

Statistic	Definition
LoggedOnTime Session	Total time, in seconds, the agent has been logged in.
NotReadyTime Session	Total time, in seconds, the agent was in the Not Ready state for all skill groups.
ICMAvailable TimeSession	Total time, in seconds, the agent was in the Unified ICM Available state.
RoutableTime Session	Total time, in seconds, the agent was in the Routable state for all skill groups.
AgentOutCalls Session	Total number of completed outbound ACD calls made by agent.
AgentOutCalls TalkTimeSession	Total talk time, in seconds, for completed outbound ACD calls handled by the agent. The value includes the time spent from the call being initiated by the agent to the time the agent begins after call work for the call. The time includes hold time associated with the call.
AgentOutCalls Time Session	Total handle time, in seconds, for completed outbound ACD calls handled by the agent. The value includes the time spent from the call being initiated by the agent to the time the agent completes after call work time for the call. The time includes hold time associated with the call.
AgentOutCalls Held Session	The total number of completed outbound ACD calls the agent has placed on hold at least once.
AgentOutCalls HeldTime Session	Total number of seconds outbound ACD calls were placed on hold.
HandledCalls Session	The number of inbound ACD calls handled by the agent.
HandledCalls TalkTime Session	Total talk time in seconds for Inbound ACD calls counted as handled by the agent. Includes hold time associated with the call.
HandledCalls AfterCall TimeSession	Total after call work time in seconds for Inbound ACD calls counted as handled by the agent.
HandledCalls Time Session	Total handle time, in seconds, for inbound ACD calls counted as handled by the agent. The time spent from the call being answered by the agent to the time the agent completed after call work time for the call. Includes hold time associated with the call.
IncomingCalls Held Session	The total number of completed inbound ACD calls the agent placed on hold at least once.

Statistic	Definition
IncomingCalls HeldTime Session	Total number of seconds completed inbound ACD calls were placed on hold.
InternalCallsSession	Number of internal calls initiated by the agent.
InternalCalls TimeSession	Number of seconds spent on internal calls initiated by the agent.
InternalCalls RcvdSession	Number of internal calls received by the agent.
InternalCalls RcvdTime Session	Number of seconds spent on internal calls received by the agent.
InternalCalls Held Session	The total number of internal calls the agent placed on hold at least once.
InternalCalls HeldTime Session	Total number of seconds completed internal calls were placed on hold.
AutoOutCalls Session	Total number of AutoOut (predictive) calls completed by the agent.
AutoOutCalls TalkTime Session	Total talk time, in seconds, of AutoOut (predictive) calls completed by the agent. The value includes the time spent from the call being initiated by the agent to the time the agent begins after call work for the call. The time includes hold time associated with the call.
AutoOutCalls Time Session	Total handle time, in seconds, for AutoOut (predictive) calls completed by the agent. The value includes the time spent from the call being initiated by the agent to the time the agent completes after call work time for the call. The time includes hold time associated with the call.
AutoOutCalls Held Session	The total number of completed AutoOut (predictive) calls the agent has placed on hold at least once.
AutoOutCalls HeldTime Session	Total number of seconds AutoOut (predictive) calls were placed on hold.
PreviewCalls Session	Total number of outbound Preview calls completed by the agent.
PreviewCalls TalkTime Session	Total talk time, in seconds, of outbound Preview calls completed by the agent. The value includes the time spent from the call being initiated by the agent to the time the agent begins after call work for the call. The time includes hold time associated with the call.

Statistic	Definition
PreviewCalls TimeSession	Total handle time, in seconds, outbound Preview calls completed by the agent. The value includes the time spent from the call being initiated by the agent to the time the agent completes after call work time for the call. The time includes hold time associated with the call.
PreviewCalls HeldSession	The total number of completed outbound Preview calls the agent has placed on hold at least once.
PreviewCalls HeldTime Session	Total number of seconds outbound Preview calls were placed on hold.
Reservation CallsSession	Total number of agent reservation calls completed by the agent.
Reservation CallsTalk TimeSession	Total talk time, in seconds, of agent reservation calls completed by the agent. The value includes the time spent from the call being initiated by the agent to the time the agent begins after call work for the call. The time includes hold time associated with the call.
Reservation CallsTime Session	Total handle time, in seconds, agent reservation calls completed by the agent. The value includes the time spent from the call being initiated by the agent to the time the agent completes after call work time for the call. The time includes hold time associated with the call.
Reservation CallsHeld Session	The total number of completed agent reservation calls the agent has placed on hold at least once.
Reservation CallsHeld TimeSession	Total number of seconds agent reservation calls were placed on hold.
BargeInCalls Session	Total number of supervisor call barge-ins completed.
InterceptCalls Session	Total number of supervisor call intercepts completed.
MonitorCalls Session	Total number of supervisor call monitors completed.
WhisperCalls Session	Total number of supervisor whisper calls completed.
EmergencyCallsSession	Total number of emergency calls.
AvailTimeToday	Total time, in seconds, the agent was in the Available state for any skill group.
LoggedOnTime Today	Total time, in seconds, the agent has been logged in.
NotReadyTime Today	Total time, in seconds, the agent was in the Not Ready state for all skill groups.

Statistic	Definition
ICMAvailable TimeToday	Total time, in seconds, the agent was in the Unified ICM Available state.
RoutableTime Today	Total time, in seconds, the agent was in the Routable state for all skill groups.
AgentOutCalls Today	Total number of completed outbound ACD calls made by agent.
AgentOutCalls TalkTime Today	Total talk time, in seconds, for completed outbound ACD calls handled by the agent. The value includes the time spent from the call being initiated by the agent to the time the agent begins after call work for the call. The time includes hold time associated with the call.
AgentOutCalls Time Today	Total handle time, in seconds, for completed outbound ACD calls handled by the agent. The value includes the time spent from the call being initiated by the agent to the time the agent completes after call work time for the call. The time includes hold time associated with the call.
AgentOutCalls HeldToday	The total number of completed outbound ACD calls the agent has placed on hold at least once.
AgentOutCalls HeldTime Today	Total number of seconds outbound ACD calls were placed on hold.
HandledCalls Today	The number of inbound ACD calls handled by the agent.
HandledCalls TalkTime Today	Total talk time in seconds for Inbound ACD calls counted as handled by the agent. Includes hold time associated with the call.
HandledCalls AfterCall TimeToday	Total after call work time in seconds for Inbound ACD calls counted as handled by the agent.
HandledCalls TimeToday	Total handle time, in seconds, for inbound ACD calls counted as handled by the agent. The time spent from the call being answered by the agent to the time the agent completed after call work time for the call. Includes hold time associated with the call.
IncomingCalls HeldToday	The total number of completed inbound ACD calls the agent placed on hold at least once.
IncomingCalls HeldTime Today	Total number of seconds completed inbound ACD calls were placed on hold.
InternalCalls Today	Number of internal calls initiated by the agent.

Statistic	Definition
InternalCalls TimeToday	Number of seconds spent on internal calls initiated by the agent.
InternalCalls RcvdToday	Number of internal calls received by the agent.
InternalCalls RcvdTime Today	Number of seconds spent on internal calls received by the agent.
InternalCalls HeldToday	The total number of internal calls the agent placed on hold at least once.
InternalCalls HeldTime Today	Total number of seconds completed internal calls were placed on hold.
AutoOutCalls Today	Total number of AutoOut (predictive) calls completed by the agent.
AutoOutCalls TalkTime Today	Total talk time, in seconds, of AutoOut (predictive) calls completed by the agent. The value includes the time spent from the call being initiated by the agent to the time the agent begins after call work for the call. The time includes hold time associated with the call.
AutoOutCalls TimeToday	Total handle time, in seconds, for AutoOut (predictive) calls completed by the agent. The value includes the time spent from the call being initiated by the agent to the time the agent completes after call work time for the call. The time includes hold time associated with the call.
AutoOutCalls HeldToday	The total number of completed AutoOut (predictive) calls the agent has placed on hold at least once.
AutoOutCalls HeldTime Today	Total number of seconds AutoOut (predictive) calls were placed on hold.
PreviewCalls Today	Total number of outbound Preview calls completed by the agent.
PreviewCalls TalkTimeToday	Total talk time, in seconds, of outbound Preview calls completed by the agent. The value includes the time spent from the call being initiated by the agent to the time the agent begins after call work for the call. The time includes hold time associated with the call.
PreviewCalls TimeToday	Total handle time, in seconds, outbound Preview calls completed by the agent. The value includes the time spent from the call being initiated by the agent to the time the agent completes after call work time for the call. The time includes hold time associated with the call.

Statistic	Definition
PreviewCalls HeldToday	The total number of completed outbound Preview calls the agent has placed on hold at least once.
PreviewCalls HeldTimeToday	Total number of seconds outbound Preview calls were placed on hold.
Reservation CallsToday	Total number of agent reservation calls completed by the agent.
Reservation CallsTalk TimeToday	Total talk time, in seconds, of agent reservation calls completed by the agent. The value includes the time spent from the call being initiated by the agent to the time the agent begins after call work for the call. The time includes hold time associated with the call.
Reservation CallsTimeToday	Total handle time, in seconds, agent reservation calls completed by the agent. The value includes the time spent from the call being initiated by the agent to the time the agent completes after call work time for the call. The time includes hold time associated with the call.
Reservation CallsHeldToday	The total number of completed agent reservation calls the agent has placed on hold at least once.
Reservation CallsHeldTimeToday	Total number of seconds agent reservation calls were placed on hold.
BargeInCalls Today	Total number of supervisor call barge-ins completed.
InterceptCalls Today	Total number of supervisor call intercepts completed.
MonitorCalls Today	Total number of supervisor call monitors completed.
WhisperCalls Today	Total number of supervisor whisper calls completed.
EmergencyCalls Today	Total number of emergency calls.
AvailTime Session	Total time, in seconds, the agent was in the Available state for any skill group.
LoggedOnTime Session	Total time, in seconds, the agent has been logged in.
NotReadyTime Session	Total time, in seconds, the agent was in the Not Ready state for all skill groups.
ICMAvailable TimeSession	Total time, in seconds, the agent was in the Unified ICM Available state.
RoutableTime Session	Total time, in seconds, the agent was in the Routable state for all skill groups.

Statistic	Definition
AgentOutCalls Session	Total number of completed outbound ACD calls made by agent.

Methods

The following table lists the Agent object methods.

Table 3: Agent Object Methods

Method	Description
DisableAgentStatistics	Disables agent statistic messages.
DisableSkillGroupStatistics	Disables skill group statistic messages.
DumpProperties	For more information, see CtiOs Object
EnableAgentStatistics	Enables agent statistic messages.
EnableSkillGroupStatistics	Enables skill group statistic messages.
GetAgentState	Returns the current agent state.
GetAllProperties	For more information, see CtiOs Object
GetElement	For more information, see CtiOs Object
GetMonitoredAgent	Returns the Agent object that is currently being monitored.
GetMonitoredCall	Returns the Call object that is currently being monitored.
GetNumProperties	For more information, see CtiOs Object
GetPropertyName	For more information, see CtiOs Object
GetPropertyType	For more information, see CtiOs Object
GetSkillGroups	Returns an array of SkillGroups objects
GetValue	For more information, see CtiOs Object
GetValueArray	For more information, see CtiOs Object
GetValueInt	For more information, see CtiOs Object
GetValueString	For more information, see CtiOs Object
IsAgent	Checks the current mode and returns true if agent mode.

Method	Description
IsSupervisor	Checks the current mode and returns true if supervisor mode.
IsValid	For more information, see CtiOs Object
Login	Logs an agent in to the ACD.
Logout	Logs an agent out of the ACD.
MakeCall	Initiates a call to a device or agent.
MakeEmergencyCall	Lets an agent make an emergency call to the supervisor.
QueryAgentState	Gets the current agent state from CTI Server and retrieves it.
ReportBadCallLine	Informs the CTI OS Server of a bad line.
RequestAgentTeamList	Retrieves the current agent team list.
RequestSupervisorAssist	Allows the agent to call an available supervisor for assistance.
SendChatMessage	Send asynchronous messages between CTI clients.
SetAgentGreetingAction	Sets the value of the Agent Greeting Action to enable or disable Agent Greeting for the logged in agent.
SetAgentState	Requests a new agent state.
SetValue	Sets the value of the property whose name is specified.
StartMonitoringAgent	Enables monitoring of a specified agent.
StartMonitoringAgentTeam	Enables monitoring of a specified agent team.
StartMonitoringAllAgentTeams	Enables monitoring of all agent teams.
StartMonitoringCall	Enables monitoring of a specified Call object.
StopMonitoringAgent	Disables monitoring of a specified agent.
StopMonitoringAgentTeam	Disables monitoring of a specified agent team.
StopMonitoringAllAgentTeams	Disables monitoring of all agent teams.
SuperviseCall	Enables monitoring a call of an agent on your team.

Arguments Parameters

The following rules apply to the optional_args and reserved_args parameters in Call Object methods:

- In VB, you can ignore these parameters altogether. For example, you can treat the line:

```
Answer([reserved_args As IArguments]) As Long
```

as follows:

```
Answer()
```

- To ignore these parameters in COM you must send a NULL, as shown:

```
Answer (NULL)
```

DisableAgentStatistics

The DisableAgentStatistics method is sent by an agent to request that real-time statistics stop being sent to that agent.

Syntax

C++

```
int DisableAgentStatistics (Arguments& reserved_args)
```

COM

```
HRESULT DisableAgentStatistics (/*[in]*/ IArguments reserved_args, /* [out, retval]*/
int * errorcode)
```

VB

```
DisableAgentStatistics (reserved_args As CTIOSCLIENTLib.IArguments) As Long
```

Java

```
int DisableAgentStatistics (Arguments reservedargs)
```

.NET

```
CilError DisableAgentStatistics(Arguments args)
```

Parameters

.NET:args

Not currently used, reserved for future use.

All Others:reserved_args

Not currently used, reserved for future use.

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Value

Default CTI OS return values. For more information, see CIL Coding Conventions.

DisableSkillGroupStatistics

The DisableSkillGroupStatistics method is sent by an agent to request that real-time statistics stop being sent to that agent.

Syntax

C++

```
int DisableSkillGroupStatistics (Arguments& optional_args)
```

COM

```
HRESULT DisableSkillGroupStatistics (/* [in, optional]*/ IArguments * optional_args,
/* [out, retval]*/ int * errorcode)
```

VB

```
DisableSkillGroupStatistics (optional_args As CTIOSCLIENTLib.IArguments) As Long
```

Java

```
int DisableSkillGroupStatistics (Arguments optional_args
```

.NET

```
CilError DisableSkillGroupStatistics(Arguments args)
```

Parameters

optional_args

An optional input parameter containing a pointer or a reference to an Arguments array containing a member that is a nested Arguments array with the keyword SkillGroupNumbers. Within this array, for each skill group to be disabled, specify a string key of an integer starting with 1 and an integer value for skill group number and specify a string key of an integer and integer value for skill group priority. If the parameter is NULL or missing, statistics are disabled for all skill groups to which the agent belongs.

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Value

Default CTI OS return values. For more information, see CIL Coding Conventions.

EnableAgentStatistics

The EnableAgentStatistics method is sent by an agent to request that real-time statistics be sent to that agent.

Syntax

C++

```
int EnableAgentStatistics(Arguments& reserved_args)
```

COM

```
HRESULT EnableAgentStatistics (/*[in]*/ IArguments* reserved_args, /* [out, retval]*/
int * errorcode)
```

VB

```
EnableAgentStatistics (reserved_args As CTIOSCLIENTLib.IArguments) As Long
```

Java

```
int EnableAgentStatistics(Arguments args)
```

.NET

```
CilError EnableAgentStatistics(Arguments args)
```

Parameters

reserved_args

Not currently used, reserved for future use.

Java/.NET:args

Not currently used, reserved for future use.

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Value

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

The CTI OS server sends agent statistics in an OnAgentStatistics event. For more information about the PollingIntervalSec and PollForAgentStatsAtEndCall registry settings and how these settings affect the refresh rate of agent statistics, see OnAgentStatistics in Chapter 6, Event Interfaces and Events

EnableSkillGroupStatistics

The EnableSkillGroupStatistics method is sent by an agent to request that real-time statistics be sent to that agent. If the Argument array is empty, then statistics for all skill groups are sent. This is useful when a monitoring application needs to view all statistics without having to enumerate and loop over each statistic to enable it.

Syntax

C++

```
int EnableSkillGroupStatistics (Arguments& optional_args)
```

COM

```
HRESULT EnableSkillGroupStatistics (/*[in]*/ IArguments * optional_args, /* [out,
retval]*/ int * errorcode)
```

VB

```
EnableSkillGroupStatistics (optional_args As CTIOSCLIENTLib.IArguments) As Long
```

Java

```
Java:int EnableSkillGroupStatistics(Arguments optional_args)
```

.NET

```
CilError EnableSkillGroupStatistics(Arguments args)
```

Parameters

optional_args

An optional input parameter containing a pointer or a reference to an Arguments array containing a member that is a nested Arguments array with the keyword SkillGroupNumbers. Within this array, each member has a string key of an integer starting with 1 and an integer value that is a skill group number to be enabled and a string key of an integer and integer value that is a skill group priority to be enabled. If the parameter is NULL or missing, statistics are enabled for all skill groups to which the agent belongs.

args

Refer to the description for optional_args above.

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Value

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

The CTI OS server sends SkillGroup statistics in the OnSkillGroupStatisticsUpdated event of the SkillGroup object.

GetAgentState

The GetAgentState method returns the current state of the agent.

Syntax

C++

```
enumCTIOS_AgentState GetAgentState()
```

COM

```
HRESULT GetAgentState (/*[in]*/ long *state)
```

VB

```
GetAgentState () As Long
```

Java

```
int GetAgentState()
```

.NET

```
AgentState GetAgentState()
```

Parameters

state

Output parameter (return parameter in VB) containing the current agent state in the form of one of the values in Table 2.

Return Value

For C++, VB, Java, and .NET, this method returns the current state of the agent.

GetAllProperties

For more information about the GetAllProperties method, see CtiOs Object.

GetElement

For more information about the GetElement method, see CtiOs Object.

GetMonitoredAgent

The GetMonitoredAgent method returns the Agent object that is currently being monitored.

Syntax

C++

```
CAgent* GetMonitoredAgent()
```

COM

```
HRESULT GetMonitoredAgent (/*[out, retval]*/IAgent **agent)
```

VB

```
GetMonitoredAgent () As CTIOSCLIENTLib.IAgent
```

Java

```
Agent GetMonitoredAgent()
```

.NET

```
Agent GetMonitoredAgent()
```

Parameters

agent

Output parameter (return parameter in VB) that contains a pointer to a pointer to an Agent object containing the currently monitored agent.

Return Value

This method returns the current monitored agent. The C++, Java, and .NET versions return null if no agent is currently being monitored.

Remarks

Supported for use with Unified CCE only.

GetMonitoredCall

The GetMonitoredCall method returns the Call object that is currently being monitored.

Syntax

C++

```
CCall* GetMonitoredCall()
```

COM

```
HRESULT GetMonitoredCall (/*[out, retval]*/ICall **call)
```

VB

```
GetMonitoredCall () As CTIOSCLIENTLib.ICall
```

Java

```
Call GetMonitoredCall()
```

.NET

```
Call GetMonitoredCall()
```

Parameters

call

Output parameter (return parameter in VB) that contains a pointer to a pointer to a Call object containing the currently monitored call.

Return Value

This method returns the current monitored call. The C++, Java, and .NET versions return null if no call is currently being monitored.

Remarks

Supported for use with Unified CCE only.

GetNumProperties

For more information about the GetNumProperties method, see CtiOs Object.

GetPropertyName

For more information about the GetNumProperties method, see CtiOs Object.

GetPropertyType

For more information about the GetNumProperties method, see CtiOs Object.

GetSkillGroups

If skillgroupstats is enabled, the GetSkillGroups method allows a client to retrieve a list that contains references to all the skill group objects to which the agent belongs. To retrieve skill groups enable skill group statistics, and turn off agent event minimization by setting its value to 0 on the CTI OS server in the registry key, for example:

```
HKLM\SOFTWARE\Cisco
Systems,Inc.\Ctios\<Customer-Instancename>\CTIOS1\Server\Agent\MinimizeAgentStateEvents
```

The skill group information is available on the agent state change event if the minimization is turned off. The following code example shows how to access the skill group properties of the Agent object:

```
    Log m_Agent.DumpProperties
    Dim i As Integer

    For i = 1 To 20
    If m_Agent.IsValid("SkillGroup[" & i & "]") Then
       Set argskills = m_Agent.GetValueArray("SkillGroup[" & i & "]")
       Log "SkillGroup[" & i & "]:" & argskills.DumpArgs
     Else
        Log "SkillGroup[" & i & "] args doesnt exist"
     End If
     Next i
```

Syntax

C++

```
Arguments & GetSkillGroups();
```

COM

```
HRESULT GetSkillGroups (/*[out,retval]*/ VARIANT * pVariantArgs);
```

VB

```
GetSkillGroups () As Variant
```

Java

```
Arguments GetSkillGroups()
```

.NET

```
Arguments GetSkillGroups()
```

Parameters

None.

Return Value

This method returns -1 if skillgroupstats is not enabled.

C++

In C++ the GetSkillGroups method returns an Arguments array containing references to CSkillGroup objects.

Each element in the returned Arguments array consists of a key/value pair, in which the element key is the Unique Object Id of the skill group object and the value is a reference to a CILRefArg object instance that contains the actual reference to a CSkillGroup object. To retrieve a reference to a skill group object, you need to do something similar to what is shown in the following code example.

```
Arguments & arSkills = m_Agent->GetSkillGroups();
if(Arguments::IsValidReference(arSkills)){
     for(int nI = 1; nI <= arSkills.NumElements(); nI ++){
         string strUOID = arSkills.GetElementKey(nI);
         CilRefArg & pRefArg = (CilRefArg &) arSkills.GetValue(strUOID);
```

```
          if(Arg::IsValidReference(*pRefArg)){
               CSkillGroup * pSkill = pRefArg->GetValue();
               pRefArg->Release();

               cout << "Skill Object (" << strUOID << ") ;
               cout << " Skill Group Number: " <<  ;
                            pSkill->GetValueInt(CTIOS_SKILLGROUPNUMBER);
          }
}
```

COM

In COM the GetSkillGroups method returns a pointer to a variant that encapsulates a Safearray where each element is a pointer to an ISkillGroup object.

To retrieve references to skill group objects, you need to do something similar to what is shown in the following code example.

```
HRESULT hr = S_OK;VARIANT varSkills;

VariantInit(&varSkills)

hr  = m_Agent->GetSkillGroups(&varSkills);

if(SUCCEDED(hr)){
     if(varSkills.vt  == (VT_ARRAY | VT_DISPATCH) ){
         long                  lNumElements = 0;

          SafeArrayGetUBound(varSkills.parray,1,&lNumElements);

         for(long nI = 0; nI < lNumElements; nI ++){
             ISkillGroup * pSkill= NULL;
             hr=SafeArrayGetElement(varSkills.parray,&nI,&pSkill);
             if(SUCCEDED(hr)){
                  int nSkillGrpNumber = 0;
                   VARIANT vPropKey;
                   VariantInit(&vPropKey);
                   vPropKey.vt = VT_BSTR;
                   vPropKey.bstr =  OLESTR("SkillGroupNumber");
                   pSkill->GetValueInt(vPropKey,&nSkillGrpNumber);
                  pSkill->Release();
                  VariantClear(&vPropKey);
             }
         }
     }
}
```

VB

In VB, the GetSkillGroups method returns a variant array where each element is a reference to a CTIOSClientLib.SkillGroup object.

To retrieve references to skill group objects you need to do something similar to what is shown in the following code example:

```
Dim  obSkill As CTIOSClientLib.SkillGroupDim arSkills As Variant
              Dim lNumElements as Long

              arSkills = m_Agent.GetSkillGroups()
```

```
              lNumElements = UBound(arSkills,1)
             For nI = 0 to lNumElements
                  Set obSkill = arSkills(nI)
                  Print "SkillGroup" &
obSkill.GetValueString(CStr("UniqueObjectId")) & _
                                "Skill Group Number: " &
obSkill.GetValueInt(CStr("SkillGroupNumber"))
            Next
            End For
```

GetValue Methods

For more information about the GetValue, GetValueInt, GetValueArray, and GetValueString methods, see CtiOs Object.

IsAgent

The IsAgent method determines whether the AgentMode connection is for an agent rather than a supervisor.

Syntax

C++

```
bool IsAgent()
```

COM

```
HRESULT IsAgent (VARIANT_BOOL *bIsAgent)
```

VB

```
IsAgent () As Boolean
```

Java

```
boolean IsAgent()
```

.NET

```
bool IsAgent()
```

Parameters

IsAgent

Output parameter (return parameter in VB) that returns true if the current AgentMode connection is for an agent and false if it is for a supervisor.

Return Value

Returns true if the current AgentMode connection is for an agent and false if the connection is for a supervisor.

IsSupervisor

The IsSupervisor method determines whether the AgentMode connection is for a supervisor.

Syntax

C++

```
bool IsSupervisor()
```

COM

```
HRESULT IsSupervisor (VARIANT_BOOL * bIsSupervisor)
```

VB

```
IsSupervisor () As Boolean
```

Java

```
boolean IsSupervisorMode()
```

.NET

```
bool IsSupervisor()
```

Parameters

bIsSupervisor

Output parameter (return parameter in VB) that returns true if the current AgentMode connection is for a supervisor and false if it is for an agent.

Return Values

If the current session is for a supervisor, this method returns true. Otherwise the method returns false.

Login

The Login method performs a login to the ACD (if supported). Generally, the minimum parameters required to log in to an ACD are AgentID and AgentInstrument. Often, based on customer configuration, the minimum requirements include an ACD password (AgentPassword). Some switches require PositionID in place of (or in addition to) AgentInstrument. Optional arguments include Extension or AgentWorkMode.

To sign on a mobile agent, you must set the following parameters:

- CTIOS_REMOTELOGIN set to true
- CTIOS_AGENTREMOTENUMBER
- CTIOS_AGENTCALLMODE

Example

rArgs.SetValue(Enum_CtiOs.CTIOS_REMOTELOGIN, "true");

rArgs.SetValue(Enum_CtiOs.CTIOS_AGENTREMOTENUMBER,"777989");

rArgs.SetValue(Enum_CtiOs.CTIOS_AGENTCALLMODE, 4);

Syntax

C++

```
virtual int Login(Arguments & args);
```

COM

```
HRESULT Login ( /*[in]*/ IArguments * pVariantArgs, /*[out]*/ int * errorcode );
```

VB

```
Login (args As CTIOSCLIENTLib.IArguments) As Long
```

Java

```
int Login(Arguments args)
```

.NET

```
CilError Login(Arguments args)
```

Input Parameters

args

Arguments array that contains the login parameters that are listed in the following table:

Table 4: Login Parameters

Keyword	Type	Description
AgentID (required)**	STRING*	The agent's login ID.
AgentInstrument	STRING*	The agent's instrument number.
LoginName (required)**	STRING	The agent's login name.
AgentExtension	STRING*	The agent's teleset extension. Optional if AgentInstrument is provided.
AgentPassword (optional)	STRING*	The agent's password.
AgentWorkMode (optional)	INT	A value representing the desired work mode of the agent. Used by Avaya Communications Manager (ACM) ECS with default value of ManualIn.
NumSkillGroups (optional)	INT	The number of Skill Groups that the agent is currently associated with, up to a maximum of 20.
PeripheralID (optional)	INT	The Unified ICM Peripheral ID of the ACD the agent is attached to.
SkillGroupNumber (optional)	INT	The number of an agent skill group associated with the agent.
SkillGroupPriority (optional)	INT	The priority of an agent skill group associated with the agent.
Agent CallMode	INT	A value that indicates the agent's call mode. Valid values are call-by-call (3) and nailed-up (4).

Keyword	Type	Description
AgentRemote Number	STRING	The phone number that the agent uses for remote login.
RemoteLogin	INT	A value that indicates the agent is configured for remote login as a remote agent.

*The CTI OS server imposes no restriction on the maximum length of this string. However, such restrictions are generally imposed by your switch/ACD and Cisco CTI Server. Consult the documentation for the switch/ACD or CTI Server for information on length restrictions for this string.

** Either AgentID or LoginName is required.

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Values

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

If the Login request is successful, it returns a CIL_OK CtiOs_Enums.CilError code In addition, the requesting client can expect an AgentStateChange event if the request is successful with an Arguments member with keyword “AgentState” and value of the agent's current state. (For more information about possible values, see GetAgentState.)

If the Login request is unsuccessful, the client receives an OnControlFailureConf event and the request returns one of the following CtiOs_Enums.CilError codes:

- E_CTIOS_INVALID_SESSION -- either the agent is not associated with the session or the session is not connected.
- E_CTIOS_INVALID_ARGUMENT -- null or invalid arguments were provided.
- E_CTIOS_LOGIN_INCONSISTENT_ARGUMENTS -- Login request argument values for AgentId and/or PeripheralID do not match the values that were set by SetAgent() prior to the Login request.

Logout

The Logout method logs the agent out of the ACD. If the ACD configuration requires or supports other parameters, you can pass these in as logout parameters. Examples are logout reason codes (supported on ACM ECS, Unified CCE).

Syntax

C++

```
int Logout (Arguments& args)
```

COM

```
HRESULT Logout (/*[in]*/ IArguments args, /*[out,retval]*/ int * errorcode)
```

VB

```
Logout (args As CTIOSCLIENTLib.IArguments) As Long
```

Java

```
int Logout(Arguments args)
```

.NET

```
CilError Logout(Arguments args)
```

Input Parameters

args

Input parameter in the form of an Arguments array that contains the Logout parameters that are listed in the following table:

Table 5: Logout Parameters

Keyword	Type	Description
EventReasonCode	INT	Reason for logging out. Required for Unified CCE , optional for all other switches.
AgentPassword (optional)	STRING*	The agent's password.
NumSkillGroups (optional)	INT	The number of Skill Groups that the agent is currently associated with, up to a maximum of 20.
SkillGroupNumber (optional)	INT	The number of an agent skill group associated with the agent.
SkillGroupPriority (optional)	INT	The priority of an agent skill group associated with the agent.
AgentID (optional)	STRING*	The agent's login ID.
AgentInstrument	STRING*	The agent's instrument number.
PeripheralID (optional)	INT	The Unified ICM Peripheral ID of the ACD the agent is attached to.

*The CTI OS server imposes no restriction on the maximum length of this string. However, such restrictions are generally imposed by your switch/ACD and Cisco CTI Server. Consult the documentation for the switch/ACD or CTI Server for information on length restrictions for this string.

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Values

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

If the request is successful, the client receives an OnAgentStateChange event with an Arguments member with keyword “AgentState” and value eLogout. If it is unsuccessful, the client receives an OnControlFailureConf event. The client also receives an OnPreLogout event before the OnAgentStateChange event, and an OnPostLogout event afterwards.

MakeCall

The MakeCall method initiates a call to a device or agent. The simplest form of the request requires only a DialedNumber.

Note You can select and make the call against the skillgroup. Do not set the value if the default skillgroup is desired.

Syntax

C++

```
int MakeCall (Arguments& args)
```

COM

```
HRESULT MakeCall (/*[in]*/ IArguments *args, /*[out,retval]*/ int * errorcode)
```

VB

```
MakeCall (args As CTIOSCLIENTLib.IArguments) As Long
```

Java

```
int MakeCall(Arguments args)
```

.NET

```
CilError MakeCall(Arguments args)
```

Input Parameters

args

Input parameter in the form of an Arguments array that contains the MakeCall parameters that are listed in the following table:

Table 6: MakeCall Parameters

Keyword	Type	Description
DialedNumber (required)	STRING, maximum length 40	The number to be dialed to establish the new call.
PeripheralID (optional)	INT	The Unified ICM Peripheral ID of the ACD the agent is attached to.
AgentInstrument (optional)	STRING*	The agent's instrument number.

Keyword	Type	Description
CallPlacementType (optional)	STRING, maximum length 40	A value specifying how the call is to be placed is identified in Table 7: Call Placement Types, on page 27.
CallMannerType (optional)	INT	A value specifying additional call processing options is identified in Table 8: Unified CM Type, on page 28.
AlertRings (optional)	INT	The maximum amount of time that the call's destination remains alerting, specified as an approximate number of rings. A zero value indicates that the peripheral default (typically 10 rings) is used.
CallOption (optional)	INT	A value from Table 9: Peripheral-Specific Call Options, on page 28 specifying additional peripheral-specific call options.
FacilityType (optional)	INT	A value from Table 10: Facility Types, on page 29 indicating the type of facility to use.
AnsweringMachine (optional)	INT	A value from Table 11: Answering Machine Actions, on page 29 specifying the action to be taken if the call is answered by an answering machine.
Priority (optional)	BOOL	This field should be set to TRUE if the call receives priority handling.
PostRoute (optional)	BOOL	When this field is set to TRUE, the Post-Routing capabilities of the Unified ICM are used to determine the new call destination.
UserToUserInfo (optional)	STRING, maximum length 40	The ISDN user-to-user information.
CallVariable1 (optional)	STRING, maximum length 40	Call variable data set in the new call in place of the corresponding data in the active call.
...	...	...
CallVariable10 (optional)		

Keyword	Type	Description
ECC (optional)	ARGUMENTS	ECC data that is set in the new call in place of the corresponding data in the active call.
CallWrapupData (optional)	STRING, maximum length 40	Call-related wrapup data.
FacilityCode (optional)	STRING, maximum length 40	Set the FacilityType to 1 for trunk groups and enter the trunk access code in the FacilityCode. Set the FacilityType to 2 for skill groups and enter the SkillGroupID in the FacilityCode. Set the FacilityType to 0 for unspecified and enter a split extension or other data needed to access the chosen facility in the FacilityCode.
AuthorizationCode (optional)	STRING, maximum length 40	An authorization code needed to access the resources required to initiate the call. **Note** The AuthorizationCode parameter is not used and is not supported.
AccountCode (optional)	STRING, maximum length 40	A cost-accounting or client number used by the peripheral for charge-back purposes.
SkillGroupNumber	INT	This keyword is not functional in MakeCall. Instead, to specify the skill group in MakeCall, enter a FacilityType of 2 and enter the SkillGroupID in the FacilityCode.

Table 7: Call Placement Types

CallPlacementType	Description	Value
CPT_UNSPECIFIED	Use default call placement.	0
CPT_LINE_CALL	An inside line call.	1
CPT_OUTBOUND	An outbound call.	2
CPT_OUTBOUND_NO_ ACCESS_CODE	An outbound call that does not require an access code.	3
CPT_DIRECT_POSITION	A call placed directly to a specific position.	4

CallPlacementType	Description	Value
CPT_DIRECT_AGENT	A call placed directly to a specific agent.	5
CPT_SUPERVISOR_ASSIST	A call placed to a supervisor for call handling assistance.	6

*The CTI OS server imposes no restriction on the maximum length of this string. **However, such restrictions are generally imposed by your switch/ACD and Cisco CTI Server.** Consult the documentation for the switch/ACD or CTI Server for information on length restrictions for this string.

Table 8: Unified CM Type

CallMannerType	Description	Value
CMT_UNSPECIFIED	Use default call manner.	0
CMT_POLITE	Attempt the call only if the originating device is idle.	1
CMT_BELLIGERENT	Always attempt the call, disconnecting any currently active call.	2
CMT_SEMI_POLITE	Attempt the call only if the originating device is idle or is receiving dial tone.	3

Table 9: Peripheral-Specific Call Options

CallOption	Description	Value
COPT_UNSPECIFIED	No call options specified, use defaults.	0
COPT_CALLING_ AGENT_ONLINE	Attempt the call only if the calling agent is “online” (available to interact with the destination party).	1
COPT_CALLING_ AGENT_RESERVED	Attempt the call only if ACDNR on the calling agent's set is activated.	2
COPT_CALLING_ AGENT_NOT_ RESERVED	Attempt the call only if ACDNR on the calling agent's set is not activated.	3
COPT_CALLING_ AGENT_BUZZ_BASE	Applies a buzz to the base of the telephone set as the call is initiated.	4
COPT_CALLING_ AGENT_BEEP_HSET	Applies a tone to the agent headset as the call is initiated.	5

CallOption	Description	Value
COPT_SERVICE_ CIRCUIT_ON	Applies a call classifier to the call (ACM ECS).	6

Table 10: Facility Types

FacilityType	Description	Value
FT_UNSPECIFIED	Use default facility type.	0
FT_TRUNK_GROUP	Facility is a trunk group.	1
FT_SKILL_GROUP	Facility is a skill group or split.	2

Table 11: Answering Machine Actions

AnsweringMachine	Description	Value
AM_UNSPECIFIED	Use default behavior.	0
AM_CONNECT	Connect call to agent when call is answered by an answering machine.	1
AM_DISCONNECT	Disconnect call when call is answered by an answering machine.	2
AM_NONE	Do not use answering machine detection.	3
AM_NONE_NO_ MODEM	Do not use answering machine detection, but disconnect call if answered by a modem.	4
AM_CONNECT_NO_MODEM	Connect call when call is answered by an answering machine, disconnect call if answered by a modem.	5

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Value

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

If the request is successful, the client receives one or more of the following call related events:

- OnCallBegin
- OnCallDelivered

- OnServiceInitiated
- OnCallOriginated
- OnCallReachedNetwork

If the request is unsuccessful, the client receives an OnControlFailureConf event.

MakeEmergencyCall

The MakeEmergencyCall method makes an emergency call to the Agent's supervisor.

Syntax

C++

```
int MakeEmergencyCall ()
int MakeEmergencyCall (Arguments& reserved_args)
```

COM

```
HRESULT MakeEmergencyCall (/*[in, optional]*/ IArguments reserved_args, /* [out, retval]*/
int * errorcode)
```

VB

```
MakeEmergencyCall () As Long
MakeEmergencyCall (reserved_args As CTIOSCLIENTLib.IArguments) As Long
```

Java

```
int MakeEmergencyCall (Arguments args)
```

.NET

```
CilError MakeEmergencyCall(Arguments args)
```

Parameters

reserved_args

Not currently used, reserved for future use.

args

Not currently used, reserved for future use.

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Value

Default CTI OS return values. For more information, see CIL Coding Conventions

Remarks

The MakeEmergencyCall request is very similar to the RequestSupervisorAssist request in the following two ways:

- Both requests place a call from the requesting agent to a supervisor and are routed employing the same script. A typical script might attempt to route the call to the primary supervisor first (if logged in and in available state) and, failing that, to route the call to a skillgroup that all supervisors belong to.
- You can configure Unified ICM Agent Desk Settings to make both call requests via a single step conference or consult call. If the consult method is chosen, the agent can complete the established consult call as a transfer or conference.

These two requests have the following important differences:

- Only Emergency calls can be recorded, if so configured in the Unified ICM Agent Desk Settings.
- The calls are reported separately in Unified ICM reporting.

Having these two separate requests gives a site some flexibility in implementing supervisor help for its agents, instructing agents to use one for certain cases and the other for different situations. In general, use the MakeEmergencyCall method for higher priority calls than calls made with the RequestSupervisorAssist method. For example, you can train agents to click the Emergency button if the customer has more than $1,000,000 in an account, and otherwise to click the Supervisor Assist button. The Supervisor can differentiate the agent's request by noting the CallType.

The MakeEmergencyCall request is specific to the Supervisor feature and should only be used on switches or configurations that have the necessary support (currently,

QueryAgentState

The QueryAgentState method lets a client retrieve the current state of the agent.

Syntax

C++

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

COM

```
HRESULT QueryAgentState ( /*[in]*/ IArguments * args, /*[out,retval]*/ int * errorcode
);
```

VB

```
QueryAgentState (ByVal args as CTIOSCLIENTLIB.IArguments) As Long
```

Java

```
int QueryAgentState (Arguments args)
```

.NET

```
CilError QueryAgentState(Arguments args)
```

Input Parameters

args

Arguments array that contains the parameters listed in the following table.

Table 12: QueryAgentState parameters

Keyword	Type	Description
Agent ID	STRING	Agent's login ID.
AgentInstrument	STRING	Agent's instrument number.

Return Values

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

If the request is successful, the client receives an OnQueryAgentStateConf event. If it is unsuccessful, the client receives an OnControlFailureConf event.

ReportBadCallLine

The ReportBadCallLine method informs the CTI OS server of the poor quality of the agent's line. A note of this is recorded in the database.

Syntax

C++

```
int ReportBadCallLine ()
int ReportBadCallLine (Arguments& reserved_args)
```

COM

```
HRESULT ReportBadCallLine (/*[in, optional]*/ IArguments reserved_args, /* [out, retval]*/
int * errorcode)
```

VB

```
ReportBadCallLine () As Long
```

Java

```
int ReportBadCallLine (Arguments args)
```

.NET

```
CilError ReportBadCallLine(Arguments args)
```

Parameters

reserved_args

Not currently used, reserved for future use.

Java/.NET: args

Not currently used, reserved for future use.

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Values

Default CTI OS return values. For more information, see CIL Coding Conventions

RequestAgentTeamList

The RequestAgentTeamList method is called by a supervisor to make a request to the CTI OS server for a list of agents in the supervisor's team.

Syntax

C++

```
int RequestAgentTeamList ()
int RequestAgentTeamList (Arguments& reserved_args)
```

COM

```
HRESULT RequestAgentTeamList (/*[in, optional]*/ IArguments reserved_args, /* [out,
retval]*/ int * errorcode)
```

VB

```
RequestAgentTeamList () As Long
```

Java

```
int RequestAgentTeamList ()
int RequestAgentTeamList (Arguments args)
```

.NET

```
CilError RequestAgentTeamList(Arguments args)
```

Parameters

reserved_args

Not currently used, reserved for future use.

Java/.NET: args

Not currently used, reserved for future use.

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Value

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

Supported for use with Unified CCE only.

If this request is successful, the CTI OS server sends a separate OnNewAgentTeamMember event for each agent in the supervisor's team. If this request is unsuccessful, the client receives an OnControlFailureConf event.

RequestSupervisorAssist

The RequestSupervisorAssist method allows the agent to call an available supervisor for assistance.

Syntax

C++

```
virtual int RequestSupervisorAssist();
int RequestSupervisorAssist (Arguments& reserved_args)
```

COM

```
HRESULT RequestSupervisorAssist (/*[in, optional]*/ IArguments reserved_args, /* [out,
retval]*/ int * errorcode)
```

VB

```
RequestSupervisorAssist () As Long
```

Java

```
int RequestSupervisorAssist(Arguments args)
```

.NET

```
CilError RequestSupervisorAssist(Arguments args)
```

Parameters

reserved_args

Not currently used, reserved for future use.

Java/.NET: args

Not currently used, reserved for future use.

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Values

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

Supported for use with Unified CCE only. For more information, see MakeEmergencyCall, on page 30.

SendChatMessage

The SendChatMessage method sends asynchronous chat-like messages between CTI OS clients. Users can specify a distribution of one or more clients, and attach a text message.

Syntax

C++

```
int SendChatMessage (Arguments& args)
```

COM

```
HRESULT SendChatMessage (/*[in]*/ IArguments *args, /*[out,retval]*/ int * errorcode)
```

VB

```
SendChatMessage (args As CTIOSCLIENTLib.IArguments) As Long)
```

Java

```
int SendChatMessage(Arguments args)
```

.NET

```
CilError SendChatMessage(Arguments args)
```

Parameters

args

Input parameter in the form of an Arguments array that contains one or more of the SendChatMessage parameters listed in the following table.

Keyword	Type	Description
Distribution (required)	STRING	Currently the only supported value is “agent”.
Target (optional)	STRING	When the Distribution is set to DistributeToAgent, you must include this field with the AgentID of the intended recipient. When the LoginName is set to the LoginName of the agent to receive the chat message, you must also set this field to the login name of the agent to which to chat.
Message (optional)	STRING	The text of the user message. Maximum message size is 255 bytes.
LoginName (optional)	STRING	Login name of the agent to receive the chat message. To chat to an agent by login name, set “LoginName” and “Target” to the login name of the agent to which to chat.

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Values

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

The recipient receives the message via the OnChatMessage event.

SetAgentGreetingAction

The SetAgentGreetingAction Sets the value of the Agent Greeting Action to enable or disable Agent Greeting for the logged in agent.

Agent Greeting is supported with CTI OS desktops created using the COM or C++ CILs.

Syntax

C++

```
int SetAgentGreetingAction(Arguments& args)
```

COM

```
HRESULT SetAgentGreetingAction (/*[in]*/ IArguments *args, int * errorcode)
```

VB

```
SetAgentGreetingAction (args As CTIOSCLIENTLib.IArguments) As Long
```

Input Parameters

args

Arguments array containing the following fields.

Table 13: SetAgentGreetingAction Parameters

Keyword	Type	Description
AgentAction	INT	1 = Disable Agent Greeting for the logged in agent. 2 = Enable agent greeting for the logged in agent/-The state to which to set the specified agent. The value of this field must be one of the values in Table 2.

Return Values

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

A successful request results in an OnAgentGreetingControlConf event. If this request is unsuccessful, the client receives an OnControlFailureConf event.

SetAgentState

The SetAgentState method requests a new agent state. Login and Logout are valid agent states and can be set using the SetAgentState method as well as by using the Login and Logout methods.

Syntax

C++

```
int SetAgentState(Arguments& args)
```

COM

```
HRESULT SetAgentState (/*[in]*/ IArguments *args, /*[out,retval]*/ int * errorcode)
```

VB

```
SetAgentState (args As CTIOSCLIENTLib.IArguments) As Long
```

Java

```
int SetAgentState(Arguments args)
```

.NET

```
CilError SetAgentState(Arguments args)
```

Input Parameters

args

Input parameter in the form of an Arguments array that contains one or more of the SetAgentState parameters listed in the following table.

Table 14: SetAgentState Parameters

Keyword	Type	Description
AgentState (required)	INT	The state to which to set the specified agent. The value of this field must be one of the values in Table 2.
AgentID (required)	STRING*	The agent's login ID.
AgentInstrument	STRING*	The agent's instrument number. Optional if Agent Extension is provided.
AgentPassword (optional)	STRING*	The agent's password.
AgentWorkMode (optional)	INT	A value representing the desired work mode of the agent. Used by ACM ECS with default value of ManualIn.
NumSkillGroups (optional)	INT	The number of Skill Groups that the agent is currently associated with, up to a maximum of 20.
EventReasonCode (optional)	INT	Reason for logging out. Required for Unified CCE , optional for all other switches.
PeripheralID (optional)	INT	The Unified ICM Peripheral ID of the ACD the agent is attached to.
SkillGroupNumber (optional)	INT	The optional, user-defined number of an agent skill group associated with the agent.
SkillGroupPriority (optional)	INT	The priority of an agent skill group associated with the agent.

*The CTI OS server imposes no restriction on the maximum length of this string. However, such restrictions are generally imposed by your switch/ACD and Cisco CTI Server. Consult the documentation for the switch/ACD or CTI Server for information on length restrictions for this string.

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Values

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

A successful request results in an OnAgentStateChanged event. It can also result in OnPreLogout, OnPostLogout, and/or OnLogoutFailed events. If this request is unsuccessful, the client receives an OnControlFailureConf event.

StartMonitoringAgent

The StartMonitoringAgent method allows the client, which must be a supervisor, to start monitoring the specified Agent object. This call causes the supervisor to receive all of the monitored call events (See IMonitoredCallEvents Interface in Event Interfaces and Events) for this agent until the supervisor calls StopMonitoringAgent.

Syntax

C++

```
int StartMonitoringAgent(Arguments& args)
```

COM

```
HRESULT StartMonitoringAgent (/*[in]*/ IArguments * args, /*[out,retval]*/ int *
errorcode)
```

VB

```
StartMonitoringAgent (args As CTIOSCLIENTLib.IArguments) As Long
```

Java

```
int StartMonitoringAgent(Arguments args)
```

.NET

```
CilError StartMonitoringCall(Arguments args)
```

Parameters

args

Arguments array that contains the constant CTIOS_AGENTREFERENCE set to the string value of the UniqueObjectID of the agent to be monitored.

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Value

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

This request is specific to the Supervisor feature and should only be used on switches or configurations that have the necessary support (currently, Unified CCE only).

The following code snippet gets the unique object ID string for an agent, then uses uses the SetValue method to store the Agent object ID and string constant CTIOS_AGENTREFERENCE in an Arguments array.

```
String StrUID = agent.GetValueString(CTIOS_UNIQUEOBJECTID Id);
arg.SetValue(CTIOS_AGENTREFERENCE, StrUID);
```

StartMonitoringAgentTeam

The StartMonitoringAgentTeam method allows the client, which must be a supervisor, to start monitoring the specified agent team. A client supervisor uses this method to receive all of the OnMonitorAgentStateChange events for every agent on the specified team.

Syntax

C++

```
int StartMonitoringAgentTeam (Arguments& args)
```

COM

```
HRESULT StartMonitoringAgentTeam (/*[in]*/ IArguments args, /*[out,retval]*/ int *
errorcode)
```

VB

```
StartMonitoringAgentTeam (args as CTIOSCLIENTLib.IArguments) As Long
```

Java

```
int StartMonitoringAgentTeam (Arguments args)
```

.NET

```
CilError StartMonitoringAgentTeam(Arguments args)
```

Parameters

args

Arguments array that contains the constant CTIOS_TEAMID set to the integer TeamID to be monitored.

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Value

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

This request is specific to the Supervisor feature and should only be used on switches or configurations that have the necessary support (currently, Unified CCE only).

StartMonitoringAllAgentTeams

The StartMonitoringAllAgentTeams method allows the client, which must be a supervisor, to start monitoring all the agents on all the supervisor's teams. This causes the supervisor to receive monitored agent events for all of the agents in the supervisor's team (for more information, see IMonitoredAgentEvents Interface in Event Interfaces and Events).

Syntax

C++

```
int StartMonitoringAllAgentTeams (Arguments& reserved_args)
```

COM

```
HRESULT StartMonitoringAllAgentTeams (/*[in, optional]*/ IArguments reserved_args,
/*[out,retval]*/ int * errorcode)
```

VB

```
StartMonitoringAllAgentTeams ([reserved_args as CTIOSCLIENTLib.IArguments]) As Long
```

Java

```
int StartMonitoringAllAgentTeams (Arguments args)
```

.NET

```
CilError StartMonitoringAllAgentTeams(Arguments args)
```

Parameters

reserved_args

Not currently used, reserved for future use.

args

Not currently used, reserved for future use.

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Value

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

This request is specific to the Supervisor feature and should only be used on switches or configurations that have the necessary support (currently, Unified CCE only).

StartMonitoringCall

Description

The StartMonitoringCall method allows the client, which must be a supervisor, to set the value of the currently monitored call that is used in the SuperviseCall method. Since there is no StopMonitoringCall, call this method with an empty args parameter to clear the value of the currently monitored call.

Syntax

C++

```
int StartMonitoringCall(Arguments& args)
```

COM

```
HRESULT StartMonitoringCall (/*[in]*/ IArguments * args, /*[out,retval]*/ int * errorcode)
```

VB

```
StartMonitoringCall (args As CTIOSCLIENTLib.IArguments) As Long
```

Java

```
int StartMonitoringCall(Arguments args)
```

.NET

```
CilError StartMonitoringCall(Arguments args)
```

Parameters

args

Arguments array that contains the constant CTIOS_CALLREFERENCE set to the string value of the UniqueObjectID of the call to be monitored.

errorCode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Value

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

This request is specific to the Supervisor feature and should only be used on switches or configurations that have the necessary support (currently,

StopMonitoringAgent

The StopMonitoringAgent method allows the client, which must be a supervisor, to stop monitoring the specified Agent object. This stops all Monitored Call events being sent to the supervisor.

Syntax

C++

```
int StopMonitoringAgent(Arguments& args)
```

COM

```
HRESULT StopMonitoringAgent (/*[in]*/ IArguments * args, /*[out,retval]*/ int * errorcode)
```

VB

```
StopMonitoringAgent (args As CTIOSCLIENTLib.IArguments) As Long
```

Java

```
int StopMonitoringAgent(Arguments args)
```

.NET

```
CilError StopMonitoringAgent(Arguments args)
```

Parameters

args

Arguments array that contains the constant CTIOS_AGENTREFERENCE set to the string value of the UniqueObjectID of the agent to stop monitoring.

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Value

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

This request is specific to the Supervisor feature and should only be used on switches or configurations that have the necessary support (currently,

StopMonitoringAgentTeam

The StopMonitoringAgentTeam method allows the client, which must be a supervisor, to stop monitoring all the agents on all the supervisor's teams.

Syntax

C++

```
int StopMonitoringAgentTeam (Arguments& args)
```

COM

```
HRESULT StopMonitoringAgentTeam (/*[in]*/ IArguments args, /*[out,retval]*/ int *
errorcode)
```

VB

```
StopMonitoringAgentTeam (args as CTIOSCLIENTLib.IArguments) As Long
```

Java

```
int StopMonitoringAgentTeam(Arguments args)
```

.NET

```
CilError StopMonitoringAgentTeam(Arguments args)
```

Parameters

args

Arguments array that contains a constant CTIOS_TEAMID set to the integer TeamID of the team to stop monitoring.

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Value

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

This request is specific to the Supervisor feature and should only be used on switches or configurations that have the necessary support (currently,

StopMonitoringAllAgentTeams

The StopMonitoringAllAgentTeams method allows the client, which must be a supervisor, to stop monitoring all of the agents on all the supervisor's teams.

Syntax

C++

```
int StopMonitoringAllAgentTeams (Arguments& reserved_args)
```

COM

```
HRESULT StopMonitoringAllAgentTeams (/*[in,optional]*/ IArguments reserved_args,
/*[out,retval]*/ int * errorcode)
```

VB

```
StopMonitoringAllAgentTeams([reserved_args as CTIOSCLIENTLib.IArguments]) As Long
```

Java

```
int StopMonitoringAllAgentTeams(Arguments args)
```

.NET

```
CilError StopMonitoringAgentTeam(Arguments args)
```

Parameters

reserved_args

Not currently used, reserved for future use.

Java/.NET: args

Not currently used, reserved for future use.

errorcode

An output parameter (return parameter in VB) that contains an error code from Table 1.

Return Value

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

This request is specific to the Supervisor feature and should only be used on switches or configurations that have the necessary support (currently,

SuperviseCall

The SuperviseCall method allows the client, which must be a supervisor, to perform a supervisory action specified by the args parameter.

The SuperviseCall method is the CTI OS version of the SUPERVISE_CALL_REQ message. This method is used to barge-into and intercept agent calls by specifying a supervisory action of eSupervisorBargeIn and eSupervisorIntercept respectively. To support Cisco Unified Communications Manager silent monitor, the supervisory action eSupervisorMonitor was added. For more information, see Unified CM-Based Silent Monitoring in Your Application.

Syntax

C++

```
int SuperviseCall(Arguments& args)
```

COM

```
HRESULT SuperviseCall (/*[in]*/ IArguments * args, /*[out,retval]*/ int errorCode)
```

VB

```
SuperviseCall (args As CTIOSCLIENTLib.IArguments ) As Long
```

Java

```
int SuperviseCall(Arguments args)
```

.NET

```
CilError SuperviseCall(Arguments args)
```

Parameters

args

An input parameter in the form of a pointer to an Arguments array that contains members with string values that are the UniqueObjectIDs of the desired agent (AgentUniqueObjectID) and call (CallUniqueObjectID). Package these with the keywords “AgentReference” and “CallReference” respectively.

The third required parameter is one of the following integers representing the desired supervisory action.

Table 15: SuperviseCall Parameters

Value	Enum	Description
3	eSupervisorBargeIn	BargeIn to the specified call of the specified agent.
4	eSupervisorIntercept	Intercept the specified call of the specified agent.
1	eSupervisorMonitor	Used to silently monitor the call of the specified agent.
0	eSupervisorClear	Used to clear the silent monitor call.

Note Both SupervisorMonitor and eSupervisorClear only apply to Cisco Unified Communications Manager based silent monitor.

This is packaged with the constant CTIOS_SUPERVISORYACTION or the string “SupervisoryAction”.

Return Values

Default CTI OS return values. For more information, see CIL Coding Conventions.

Remarks

This request is specific to the Supervisor feature and should only be used on switches or configurations that have the necessary support (currently,

A BargeIn action is very similar to a Single Step Conference where the agent is the conference controller. As such, only this agent can add other parties to the conference; the supervisor cannot do this.

An Intercept can only be performed by a supervisor who has already performed a BargeIn. The Intercept simply hangs up the original agent, leaving only the customer and the supervisor talking.

E_CTIOS_INVALID_SILENT_MONITOR_MODE is returned when Agent.SuperviseCall() is called when CTI OS Based silent monitor is configured.