



## SilentMonitorManager Object

---

The SilentMonitorManager object provides developers with an interface to silently monitor behavior. The SilentMonitorManager object performs all silent monitor tasks, such as starting, stopping, and managing silent monitor sessions. The SilentMonitorManager object stores specific silent monitor session information as properties.

You can use the SilentMonitorManager object in two different modes:

- In Monitoring Mode, an application that wants to silently monitor conversation without being noticed by the calling parties must create a SilentMonitorManager object and set the mode to eSMMonitoringMode using the StartSMMonitoringMode method.
- In Monitored Mode, an application accepts requests to initiate silent monitor sessions to forward the voice conversations to the remote monitoring application. The application creates a SilentMonitorManager object and sets the mode to eSMMonitoredMode using the StartSMMonitoredMode method.

For more information about these modes see [Silent Monitor Session](#) in [Building Your Custom CTI Application](#)



**Note**

---

SilentMonitorManager Object methods and properties are not available in the Java or .NET CILs.

---



**Note**

---

SilentMonitorManager Object methods and properties are supported for use with Unified CCE only.

---



**Note**

---

SilentMonitorManager Object methods and properties are only supported for CTI OS based silent monitoring.

---

- [Properties, page 2](#)
- [Methods, page 3](#)

# Properties

The following table lists the SilentMonitorManager object properties.

**Table 1: SilentMonitorManager Object Properties**

Keyword	Type	Description
HeartbeatInterval	INT	Heartbeat interval for the silent monitor session.
HeartbeatTimeout	INT	Timeout for no activity.
IPPhoneInformation	ARGUMENTS	This property is only accessible via the GetIPPhoneInfo method. It contains all the information related to the IP Phone used by the application.
MediaTerminationPort	INT	TCP/IP port where monitored conversation is sent for playback on system sound card.
SessionInformation	ARGUMENTS	This property is only accessible via the GetSessionInfo method. It contains all the information related to the current active silent monitor session.
SManagerMode	SHORT	Mode in which the manager object operates (for more information, see the table below).  If SetIPPhoneInfo is used, SManagerMode attempts to use the information provided.  <b>Note</b> Only applies to CTI OS based Silent Monitor.

**Table 2: SManagerMode Values**

enum Value	Description	Numeric Value
eSModeNotSet	Mode not set.	-1
eSModeMonitoredMode	The manager accepts request for silent monitor sessions and forwards voice to the monitoring application.	0

enum Value	Description	Numeric Value
eSMMonitoringMode	The manager can make requests to remote client to start a silent monitor session and send voice.	1

## Methods

The following table lists the SilentMonitorManager object methods.

**Table 3: SilentMonitorManager Object Methods**

Method	Description
AcceptSilentMonitoring	Establishes a silent monitor session and immediately starts sending audio.
GetIPPhoneInfo1	Retrieves the information of the IP Phone used by the client application. Gets its information from the RTP events that occur when RTP streams are created and modified.
GetSessionInfo	Retrieves the information related to the current silent monitor session.
GetSMSSessionList	Retrieves a list of all active silent monitor sessions.
IsMonitoredTarget	Determines if the device/agent is a target being monitored.
SetIPPhoneInfo <sup>1</sup>	Saves the information of the IP Phone used by the client application.
StartSilentMonitorRequest	Sends a silent monitor session start request to a targeted client.
StartSMMonitoredMode	Puts the SilentMonitorManager in Monitored mode.
StartSMMonitoringMode	Puts the SilentMonitorManager in Monitoring mode.
StopSilentMonitorMode	Sets the SilentMonitorManager mode to eSMModeNotSet. If a silent monitor session is active at this time, the session is terminated.

Method	Description
StartSilentMonitorRequest	The StartSilentMonitorRequest () method is used to initiate a CTI OS based silent monitor session. When this method is called and Cisco Unified Communications Manager based silent monitor is configured, it returns E_CTIOS_INVALID_SILENT_MONITOR_MODE.
StopSilentMonitorRequest	Stops the active silent monitor session.

1

[2](#)

## Argument Parameter Rules

The following rules apply to the optional `_args` and reserved `_args` parameters in SilentMonitorManager object methods:

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

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

as follows:

```
StartSMMonitoringMode()
```

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

```
StartSMMonitoringMode(NULL)
```

## AcceptSilentMonitoring

The AcceptSilentMonitoring method establishes the silent monitor session requested by the OnSilentMonitorRequestedEvent and immediately starts sending audio to the monitoring client. You should only use this method if the parameter DoDefaultMessageHandling was set to False when the subscriber handled the OnSilentMonitorRequestedEvent event.

### Syntax

C++

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

<sup>2</sup> GetIPPhoneInfo and SetIPPhone Info are used by SilentMonitorManager in the following manner. The RTPStartedEvent arrives and SilentMonitorManager uses SetIPPhoneInfo to store the IP address and port carried in the RTPStartedEvent. The SilentMonitorStartRequestedEvent arrives and SilentMonitorManager uses GetIPPhoneInfo to retrieve the stored IP address and port to build the packet filter. The SilentMonitorManager uses SetIPPhoneInfo internally to populate IP phone information carried in RTPStartedEvents.

**COM**

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

**VB**

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

**Parameters**

args

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

**Table 4: AcceptSilentMonitoring arguments array parameters**

Keyword	Type	Description
MonitoredUniqueObjectID	STRING	Unique Object ID of the object being monitored.
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.
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 value is set to False, it instructs the SilentMonitorManager not to send voice and not to establish the silent monitor session. It is then the responsibility of the subscriber to report this status accordingly.

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](#).

## GetIPPhoneInfo

The GetIPPhoneInfo method gets the information about the client application IP Phone.

**Note**

You do not have to use this method. You can use the defaults to figure out the information to sniff packets from.

**Syntax****C++**

```
C++: Arguments * GetIPPhoneInfo(void);
```

**COM**

```
HRESULT GetIPPhoneInfo ( /*[out,retval]*/ IArguments ** IPPhoneInfo);
```

**VB**

```
GetIPPhoneInfo () as CTIOSCLIENTLIB.IArguments
```

**Parameters**

None.

**Return Value**

This method returns an Arguments array that contain the parameters listed in the following table.

**Table 5: GetIPPhoneInfo Return Arguments Array**

Keyword	Type	Description
ClientAddress	STRING	IP Address of the IP Phone to be used by the client application.
BitRate	INT	Audio transmission bit rate.
PacketSize	INT	Number of milliseconds of audio stored in a packet.
Direction	SHORT	One of the following values that indicates the direction of voice flow between the calling party and the called party: 0: Input 1: Output 2: Bidirectional

Keyword	Type	Description
RTPTypea	SHORT	One of the following values that indicates the type of RTP messages between the calling party and the called party: 0: audio 1: video 2: data
EchoCancelation	SHORT	One of the following values that indicates whether the echo cancellation feature is enabled on this IP Phone: 0: Off 1: On
PayLoadType	SHORT	Audio codec type.

## GetSessionInfo

The GetSessionInfo method retrieves the information related to the current silent monitor session.

### Syntax

#### C++

```
Arguments * GetSessionInfo(Arguments & args) ;
```

#### COM

```
HRESULT GetSessionInfo ( /*[in]*/ IArguments * args, /*[out,retval]*/ IArguments * SMSessionInfo );
```

#### VB

```
GetSessionInfo (ByVal args as CTIOSCLIENTLIB.IArguments) As CTIOSCLIENTLIB.IArguments
```

### Parameters

args

Arguments array that contains one of the parameters listed in the following table:

**Table 6: GetSessionInfo Arguments Array Parameters**

Keyword	Type	Description
SMSessionKey	UNSIGNED SHORT	Unique silent monitor session Object ID of the target object that is being monitored.

Keyword	Type	Description
MonitoredUniqueObjectID	STRING	Unique Object ID of the target object that is being monitored.

### Return Values

This method returns an Arguments array containing the key/value pairs listed in the following table:

**Table 7: GetSessionInfo Return Arguments Array Parameters**

Keyword	Type	Description
SMSessionKey	UNSIGNED SHORT	Unique silent monitor session Object ID of the target object that is being monitored.
SMSessionStatus	SHORT	One of the ISilentMonitorEvent status codes in <a href="#">Table 2</a> .
AudioMode	INT	Reserved. Specifies the audio mode bitmask.
AgentID/DeviceID	STRING	Agent ID or DeviceID of the target being monitored.
MonitoredUniqueObjectID	STRING	Unique Object ID of the target object being monitored.
MonitoredDeviceIPAddress	STRING	TCP/IP Address of the monitored IP Phone.
PeripheralID	INT	ID of the peripheral associated with the agent and IP phone.
MonitoringIPAddress	STRING	TCP/IP Address of the system receiving audio.
MonitoringIPPort	INT	TCP/IP port on which receiving system is listening for audio.

## GetSMSessionList

The GetSMSessionList method returns an Arguments array that contains the parameters listed in [Table 10: StartSilentMonitorRequest Arguments Array Parameters](#), on page 12. All parameters are required.

### Syntax

C++

```
Arguments * CIL_API GetSMSessionList(void)
```



**COM**

```
HRESULT GetSMSSessionList([out,retval] IArguments **pIArguments );
```

**VB**

```
GetSMSSessionList () as CTIOSCLIENTLIB.IArguments
```

**Parameters**

None.

**Return Values**

Arguments array that contains a list of all Silent Monitor sessions. The current version only allows one active session, so the main use for this function is to use the NumElements method on the returned Arguments array to determine if the current SilentMonitorManager is in an active Silent Monitor session.

## IsMonitoredTarget

The IsMonitoredTarget method determines if the specified device or agent is a target that is being monitored.

**Syntax**

**C++**

```
bool IsMonitoredTarget (Arguments & args);
```

**COM**

```
HRESULT IsMonitoredTarget ( /*[in]*/ IArguments * args, /*[out,retval]*/ VARIANT_BOOL * bMonitored );
```

**VB**

```
IsMonitoredTarget () As Boolean
```

**Parameters**

args

Arguments array that contains the parameter listed in the following table:

**Table 8: IsMonitoredTarget arguments array parameter**

Keyword	Type	Description
MonitoredUniqueObjectID	STRING	Unique Object ID of the target object being monitored.

**Return Value**

True if the specified MonitoredUniqueObjectID corresponds to the monitored agent or device; False otherwise.

## SetIPPhoneInfo

The SetIPPhoneInfo method saves the information of the IP Phone used by the client application.

You use the SetIPPhoneInfo() function to set the specific IP address/port to sniff on for RTP packets on the agent system. If you call StartSMMonitoredMode() and have not called SetIPPhoneInfo(), then the silent monitor library sniffs on all IP interfaces on the agent system and figures out the correct interface. If you set a specific ip address/port to sniff with SetIPPhoneInfo(), then the silent monitor library sniffs for RTP packets on the agent system only on that specific address and specific port. SetIPPhoneInfo() is used externally by the Agent control to set a specific address for silent monitor sniffing.



### Note

AgentState control only uses SetIPPhoneInfo in a Citrix environment. Since the IP address/port that the phone is connected to is not local to the Citrix server, the AgentState control uses terminal services APIs to figure out the IP address of the real network interface.

### Syntax

#### C++

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

#### COM

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

#### VB

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

### Parameters

args

Arguments array that can contain the parameters listed in the following table:

**Table 9: SetIPPhoneInfo arguments array parameters**

Keyword	Type	Description
ClientAddress (required)	STRING	IP Address of the IP Phone to be used by the client application.
BitRate (optional)	INT	Audio transmission bit rate.
PacketSize (optional)	INT	Number of milliseconds of audio stored in a packet.

Keyword	Type	Description
Direction (optional)	SHORT	One of the following values that indicates the direction of voice flow between the calling party and the called party: 0: Input 1: Output 2: Bidirectional
RTPType (optional)	SHORT	One of the following values that indicates the type of RTP messages between the calling party and the called party: 0: audio 1: video 2: data
EchoCancelation (optional)	SHORT	One of the following values that indicates whether the echo cancellation feature is enabled on this IP Phone: 0: Off 1: On
PayLoadType (optional)	SHORT	Audio codec type.

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](#).

## StartSilentMonitorRequest

The StartSilentMonitorRequest method sends a silent monitor session start request to a targeted client.

### Syntax

C++

```
int StartSilentMonitorRequest (Arguments & args, unsigned short * SMSessionKey );
```

COM

```
HRESULT StartSilentMonitorRequest ( /*[in]*/ IArguments * args, /*[out]*/ unsigned short * SMSessionKey, /*[out,retval]*/ int * errorcode );
```

**VB**

```
StartSilentMonitorRequestInt (ByVal args as CTIOSCLIENTLIB.IArguments, ByRef
    SMSessionKey AsLong) As Long
```

**Parameters**

args

Arguments array that contains the parameters listed in the following table. All parameters are required.

**Table 10: StartSilentMonitorRequest Arguments Array Parameters**

Keyword	Type	Description
AgentID or DeviceID	STRING	AgentID or DeviceID of the target to monitor. Specify either an AgentID or a DeviceID, not both,
PeripheralID	INT	ID of the peripheral associated with the agent or device.
MonitoringIPAddress	STRING	TCP/IP address of the system receiving audio.
MonitoringIPPort (Optional)	INT	TCP/IP port where the monitoring application is listening for audio.
HeartbeatInterval	INT	Interval in seconds between heartbeats.
HeartbeatTimeout	INT	Seconds elapsing before a Silent Monitor session is aborted because of no heartbeats received from the monitored peer.

SMSessionKey

An output parameter that contains the unique key to the started silent monitor session. You must use this key to perform any action on the currently active silent monitor session.

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 you use the DeviceID, there must be an agent associated with the device. The session timeouts if there is no agent logged into the device. An established silent monitor session ends if the associated agent logs out of the device.

E\_CTIOS\_INVALID\_SILENT\_MONITOR\_MODE is returned when SilentMonitorManager.Start

SilentMonitorRequest() is called when Cisco Unified Communications Manager based silent monitor is configured.

If an application using a version of the CIL that is older than 7.2(1) connects to a 7.2(1) CTI OS Server configured for Cisco Unified Communications Manager Based Silent Monitor and calls SilentMonitorManager.StartSilentMonitor

Request(), the application receives an OnSilentMonitorStatusReportEvent carrying a status code of eSMStatusCCMSilentMonitor.

## StartSMMonitoredMode

The StartSMMonitoredMode method puts the SilentMonitorManager in Monitored mode.

### Syntax

#### C++

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

#### COM

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

#### VB

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

### Parameters

args

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

**Table 11: StartSMMonitoredMode Arguments Array Parameters**

Keyword	Type	Description
Cluster	ARRAY	An array of IP addresses and/or hostnames for silent monitor services. These silent monitor service should all be members of the same cluster to ensure that the agent's calls can be silently monitored. The CIL randomly chooses one silent monitor service to which to connect. For more information about silent monitor service cluster configuration, see <i>CTI OS System Manager's Guide for Cisco Unified ICM/Contact Center Enterprise &amp; Hosted</i> .

Keyword	Type	Description
SMSAddr	STRING	If Cluster is not present, you can use this parameter to specify the address of a silent monitor service to which to connect.
SMSListenport	INT	The port on which the silent monitor services listen for connections.
SMSTOS	INT	The QoS setting for the connection.
SMSHeartbeats	INT	The interval in milliseconds between heartbeat packets.
SMSRetries	INT	The number of heartbeats that can be missed before the connection is aborted.

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](#).

## StartSMMonitoringMode

The StartSMMonitoringMode method puts the SilentMonitorManager in Monitoring mode.

#### Syntax

##### C++

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

##### COM

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

##### VB

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

## Parameters

**Table 12: StartSMMonitoringMode Arguments Array Parameters**

Keyword	Type	Description
SMSAddr	STRING	A string that contains the address of the silent monitor service used to decode and play back the agent's phone call.
SMSListenport	INT	The port on which the silent monitor services listen for connections.
SMSTOS	INT	The QoS setting for the connection.
SMSHeartbeats	INT	The interval in milliseconds between heartbeat packets.
SMSRetries	INT	The number of heartbeats that can be missed before the connection is aborted.

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](#).

# StopSilentMonitorMode

The StopSilentMonitorMode method sets the SilentMonitorManager mode to eSMMModeNotSet. If a silent monitor session is active at the time, the session is terminated.

## Syntax

### C++

```
int StopSilentMonitorMode (Arguments & reserved_args );
```

### COM

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

### VB

```
StopSilentMonitorMode (ByVal reserved_args as CTIOSCLIENTLIB.IArguments) As Long
```

**Parameters**

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 Values**

Default CTI OS return values. For more information, see [CIL Coding Conventions](#).

## StopSilentMonitorRequest

The StopSilentMonitorRequest method stops the Active silent monitor session.

**Syntax**

C++

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

COM

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

VB

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

**Parameters**

args

Arguments array that contains the parameter listed in the following table:

Keyword	Type	Description
SMSessionKey	UNSIGNED SHORT	Unique key of the current active silent monitor session

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](#).