



**VTGO for PC API  
Release 2.10.1.43**

## **Programmer's Reference Guide**

### **Corporate Headquarters**

IP blue Software Solutions  
15 East 26<sup>th</sup> Street  
New York, NY 10010, USA  
<http://www.ipblue.com>  
Phone: (212) 485-1200  
Fax: (212) 485-1380

Library **VTGOApi**

Public class **clsVTGO**

CLSID: {**CC6BA85F-A65C-45C1-BA99-C5EFD9E7DFAC**}

Library **ExtensionMobility**

Public class **clsMobile**

CLSID: {**806C9BAA-577A-4793-8D6A-A97F3F19B8AF**}

---

*Object Properties*

| <b>Object Summary</b> |                    |
|-----------------------|--------------------|
| Instancing            | Multiuse           |
| MTS Object            | No                 |
| Persistable           | No                 |
| Threading             | Apartment Threaded |

**Pre-conditions for using the API :**

Applicable with VTGO-PC API version 2.9.22 to next version of a major release of the API.

VTGO-PC.exe should be an active process.


Extension mobility will work for CallManager 3.x and Call Manager 4.1

**Post-conditions for using the API :**

Referenced to zero completely to ensure APIProxy.exe is no longer active.

Must be referenced to zero completely upon "PeerLost" event.

**Event Summary**

|   |
|---|
| <b>Event ConnectionKeepAlive()</b><br>When VTGO is connected to CallManager, CallManager periodically (once every 30 seconds) sends KeepAlive message to VTGO, expecting to receive another KeepAlive message back from VTGO – this way CallManager knows that this softphone is still there. Every time VTGO receives KeepAlive from CallManager, it will generate ConnectionKeepAlive() to its API clients. |
| <b>Event ConnectionStateChanged(IngNewState As Long).</b><br>When VTGO connects to CallManager, it will generate this event with state 1. When VTGO disconnects from CallManager, it will generate this event with state 0.   |
|  DestinationBusy<br><b>Event DestinationBusy(CallId As Long)</b><br>Event DestinationBusy triggered after a call is placed with MakeCall if a line reorder occurs because invalid state of call due to various reasons like destination busy. Call is represented by its ID.   |

|  |
|--|
| <p>⚡ DestinationHangUp</p> <p><b>Event DestinationHangUp(CallId As Long)</b></p> <p>Event DestinationHangUP triggered after the destination of a call that was placed with MakeCall, hangs up. Call is represented by its ID.</p>  |
| <p>⚡ DestinationPickUp</p> <p><b>Event DestinationPickUp(CallId As Long)</b></p> <p>Event DestinationPickUp triggered after the destination of a call that was placed with MakeCall, picks up. Call is represented by its ID.</p>  |
| <p>⚡ IncomingCall</p> <p><b>Event IncomingCall(CallId As Long, line As String, callerId As String)</b></p> <p>Event IncomingCall triggered once every time an incoming call occurs on the device. Call is represented by its ID, other parameters include line number and caller id.</p> |
| <p>⚡ PeerLost</p> <p><b>Event PeerLost()</b></p> <p>Event PeerLost triggered when the process, VTGO terminates.</p>  |
| <p>⚡ PeerConnected</p> <p><b>Event PeerConnected()</b></p> <p>When VTGO starts running or is already running</p>   |
| <p>⚡ SnapShotStart</p> <p><b>Event SnapShotStart()</b></p> <p>Before first snapshot event</p>  |
| <p>⚡ SnapShotEnd</p> <p><b>Event SnapShotEnd()</b></p> <p>After last snapshot event</p>  |
| <p>⚡ Buttons</p> <p><b>Buttons(Value)</b></p>  |

 LineUpdate

**LineUpdate(LinId, CallId, LineLabel, LineNumber, State, OtherParty, Origin, MWI, Button)**

Snapshot of requested line

State:

Idle = 0,

OffHook = 1,

OnHook = 2,

RingOut = 3,

RingIn = 4,

Connected = 5,

Busy = 6,

Congestion = 7,

Hold = 8,

CallWaiting = 9,

CallTransfer = 10,

CallPark = 11,

Proceed = 12,

CallRemoteMultiline = 13,

InvalidNumber = 14

 SpeedDialUpdate

**SpeedDialUpdate(LineID, LineLabel, State, Button)**

**Event StationRegister(succ As Boolean)**

## Method Summary

☞ Conference

### **Function Conference(destination As String) As Boolean**

Conference initiates a new call and joins it to the bridge if the conditions mentioned below are met.

☞ EndCall

### **Function EndCall() As Boolean**

EndCall hangs up the active call.

☞ HoldCall

### **Function HoldCall() As Boolean**

HoldCall changes the state of the active call to OnHold state.

☞ MakeCall

### **Function MakeCall(destination As String) As Boolean**

MakeCall places a new call to the destination.

☞ MakeCallLine

### **Function MakeCallLine(lineId As Long, destination As String) As Boolean**

Places new call from the selected line represented by lineId. Add char (comma) to introduce one second delay between dialed digits.

☞ OffHook

### **Sub OffHook() As Boolean**

OffHook places device to off hook.

☞ OnHook

### **Sub OnHook() As Boolean**

OnHook places device to on hook.

☞ OffHookLine

### **Function OffHookLine(lineId As Long) As Boolean**

Off hooke's the selected line.

☞ SendDTMFKey

### **Function SendDTMFKey(dtmfKey As String) As Boolean**












SendDTMFKey send key represented by string parameter.




☞ Transfer




### **Function Transfer(destination As String) As Boolean**

Transfer forwards the active call to a new destination.

SupervisedTransfer(Destination) as Boolean

|  |   |
|--|---|
|  | SupervisedTransferComplete() as Boolean   |
|  |  ResumeCall<br><b>Function ResumeCall(CallId As Long) As Boolean</b><br>ResumeCall changes the state of the held call corresponding to its ID to active state.   |
|  |  ExitProgram<br><b>Function ExitProgram() As Boolean</b><br>ExitProgram will cause VTGO-PC to exit.  |
|  |  RunProgram<br><b>Function RunProgram()</b><br>Can use this to run the program.  |
|  |  SnapShot<br><b>Function SnapShot() As Boolean</b><br>To initiate SnapShot Event sequence <ol style="list-style-type: none"> <li>1. SnapShotStart</li> <li>2. LineUpdate</li> <li>3. SpeedDialUpdate</li> <li>4. SnapShotEnd</li> </ol>  |
|  |  AdjustSpeakerVolDown<br> AdjustSpeakerVolUp  |
|  |  AdjustMicVolDown<br> AdjustMicVolUp  |
|  |  MuteMic<br><b>Function MuteMic(Flag As Long) As Boolean</b><br>Flag = 0 Un-Mute<br>Flag = 1 Mute  |
|  |  CallPark<br><b>Function CallPark() As Boolean</b>   |
|  | <b>Event CallParkReturned(Number As String)</b> when call is parked this event is fired to client containing the string with Call Park DN .   |
|  |  ACDLogin<br><b>Function ACDLogin(Server As String, Port As String, UID As String, PWD As String, Extn As String) As Boolean</b><br>Agent Login<br>Server: ACD server address<br>Port: ACD server port<br>UID: Agent User ID<br>PWD: Agent password<br>Extn: Extension assigned to the agent and must have a line appearance on the device agent is authenticating<br><br>Preconditions: VTGO-PC must be running and registered with CallManager and also with IP Phone Agent application.<br><br>Return |

|  |  |
|--|--|
|  | <p>True: Successful<br/> False: Failed login due to reasons below<br/> Authentication failure<br/> Server failure<br/> Communications failure<br/> Failure to meet the preconditions</p>   |
|  | <p> <b>ACDReady</b></p> <p><b>Function ACDReady(Server As String, Port As String) As Boolean</b><br/> To change working status of the agent<br/> Server: ACD server address<br/> Port: ACD server port</p> <p>Preconditions: VTGO-PC must be running and registered with CallManager and also with IP Phone Agent application.<br/> Agent ACD Login succeeds</p> <p>Return<br/> True: Successful<br/> False: Server failure<br/> Communications failure<br/> Failure to meet the preconditions</p>  |
|  | <p> <b>ACDNotReady</b></p> <p><b>Function ACDNotReady(Server As String, Port As String, State As String) As Boolean</b><br/> To change working status of the agent<br/> Server: ACD server address<br/> Port: ACD server port<br/> State: Name of the state parsed from ReturnACDNotReadyStates</p> <p>Preconditions: VTGO-PC must be running and registered with CallManager and also with IP Phone Agent application.<br/> Agent ACD Login succeeds<br/> Agent state is 'Ready'<br/> State value used defined in IP Phone Agent application</p> <p>Return<br/> True: Successful<br/> False: Server failure<br/> Communications failure<br/> Failure to meet the preconditions</p> |
|  | <p> <b>ACDNotReadyStates</b></p> <p><b>Function ACDNotReadyStates(Server As String, Port As String) As Boolean</b><br/> To retrieve defined ACD not ready states<br/> To change working status of the agent<br/> Server: ACD server address<br/> Port: ACD server port</p> <p>Preconditions: VTGO-PC must be running and registered with CallManager and</p>  |

|  |   |
|--|---|
|  | <p>also with IP Phone Agent application.<br/> Agent ACD Login succeeds<br/> Agent state is 'Ready'<br/> One or more state value used must be defined in IP Phone Agent application</p> <p>Return<br/> True: Successful<br/> False: Server failure<br/> Communications failure<br/> Failure to meet the preconditions</p>  |
|  | <p> ReturnACDNotReadyStates</p> <p><b>Function ReturnACDNotReadyStates() As String</b><br/> Preconditions: VTGO-PC must be running and registered with CallManager and also with IP Phone Agent application.<br/> Agent ACD Login succeeds<br/> One or more state values used must be defined in IP Phone Agent application<br/> Call ACDNotReadyStates</p> <p>Return<br/> ':' delimited string if more than one state defined</p>   |
|  | <p> ACDStatus</p> <p><b>Function ACDStatus(Server As String, Port As String) As Boolean</b><br/> To retrieve status of the agent<br/> Server: ACD server address<br/> Port: ACD server port</p> <p>Preconditions: VTGO-PC must be running and registered with CallManager and also with IP Phone Agent application.<br/> Agent ACD Login succeeds</p> <p>Return<br/> True: Successful<br/> False: Server failure<br/> Communications failure<br/> Failure to meet the preconditions<br/> Agent log out</p> |
|  | <p><b>Function getACDStatus() As String</b><br/> Preconditions: VTGO-PC must be running and registered with CallManager and also with IP Phone Agent application.<br/> Agent ACD Login succeeds<br/> One or more state value used must be defined in IP Phone Agent application<br/> Call ACDStatus</p> <p>Return<br/> ':' delimited string formatted as &lt;Agent&gt;:&lt;Status&gt;<br/> &lt;Status&gt; = 'Ready' (or) 'Not Ready'</p>  |
|  | <p> ACDLogout</p> <p><b>Function ACDLogout(Server As String, Port As String) As Boolean</b></p>  |

Enum: BreakType no longer used

To change working status of the agent

Server: ACD server address

Port: ACD server port

Sate: Name of the state parsed from ReturnACDLogoutSates

Preconditions: VTGO-PC must be running and registered with CallManager and also with IP Phone Agent application.

Agent ACD Login succeeds

Agent state is 'Ready'

State value used defined in IP Phone Agent application










Return

True: Successful

False: Server failure

Communications failure


Failure to meet the preconditions

|  |   |
|--|---|
|  | <p> ACDLogoutStates</p> <p><b>Function ACDLogoutStates(Server As String, Port As String) As Boolean</b></p> <p>To retrieve defined ACD logout states<br/> To change working status of the agent<br/> Server: ACD server address<br/> Port: ACD server port</p> <p>Preconditions: VTGO-PC must be running and registered with CallManager and also with IP Phone Agent application.<br/> Agent ACD Login succeeds<br/> One or more state value used must be defined in IP Phone Agent application</p> <p>Return<br/> True: Successful<br/> False: Server failure<br/> Communications failure<br/> Failure to meet the preconditions</p> |
|  | <p> ReturnACDLogoutStates</p> <p><b>Function ReturnACDLogoutStates() As String</b></p> <p>Preconditions: VTGO-PC must be running and registered with CallManager and also with IP Phone Agent application.<br/> Agent ACD Login succeeds<br/> One or more state values used must be defined in IP Phone Agent application<br/> Call ACDLogoutStates</p> <p>Return<br/> ':' delimited string if more than one state defined</p>   |
|  | <p> MeetMe</p> <p><b>Function MeetMe(destination As String) As Boolean</b></p>   |
|  | <p> Pickup</p> <p><b>Function Pickup() As Boolean</b></p>  |
|  | <p> Messages</p> <p><b>Function Messages() As Boolean</b><br/> For voicemail</p>   |
|  | <p> Redial</p> <p>Function RunProgram()</p>  |
|  | <p> SupervisedTransfer</p> <p> SupervisedTransferCommit</p> <p><b>Function SupervisedTransfer(destination As String) As Boolean</b><br/> <b>Function SupervisedTransferCommit() As Boolean</b></p> <p>Step1. Initiate transfer using SupervisedTransfer<br/> Step2. Commit the transfer using SupervisedTransferCommit<br/> Perquisite: Cannot commit without initiate.</p>   |
|  | <p> InitiateConference</p>   |

|  |   |
|--|---|
|  | <p>☞ CompleteConference</p> <p><b>Function InitiateConference(destination As String)</b><br/> <b>Function CompleteConference()</b><br/> Step1. Initiate conference<br/> Step2. Commit the conference using CompleteConference<br/> Perquisite: Cannot commit without initiate.</p>                              |
|  | <p>☞ PlayFile</p> <p><b>Function PlayFile(FileName As String) As Boolean</b><br/> Only valid file path of the type .wav and format 16 bit mono 8KHz PCM.<br/> To stop send '*' DTMF tone.</p>   |
|  | <p>☞ VoiceWithSmile</p> <p><b>Function VoiceWithSmile(Activate As Boolean)</b><br/> Enable/Disable the feature to play pre-configured file upon call answer.</p>  |
|  | <p>☞ RunProgramSilentMode</p> <p><b>Function RunProgramSilentMode()</b><br/> Starts the program in silent mode.</p>   |
|  | <p>☞ ShowApp</p> <p><b>Function ShowApp(Show As Boolean)</b><br/> Hide/Unhide the user interface dialog.</p>  |
|  | <p>☞ RecordCalls</p> <p><b>Function RecordCalls(bEnable As Boolean)</b><br/> When true, VTGO-PC records call when the call is connected.<br/> Applies to all calls.</p>   |
|  | <p>☞ RecordingDir</p> <p><b>Function RecordingDir(DirPath As String)</b><br/> Change directory path of the recorded file to overwrite default directory VTGO InstallRoot\VoiceRecordings.</p>   |
|  | <p>☞ RecordingFile</p> <p><b>Function RecordingFile(FileName As String)</b><br/> Change file name of the recorded file to overwrite default file name {GUID}.VTA or .WAV. Method must be called every time when a call is connected, otherwise the GUID is going to be used as the file name for that call.</p> |
|  | <p>☞ RecordingFileFormat</p> <p><b>Function RecordingFileFormat(Format As RecordedFileType)</b><br/> Default format is VTA</p>  |


|  |
|--|
| Members of 'RecordedFileType'                                |
| <input type="checkbox"/> VTA<br><input type="checkbox"/> WAV |


|   |
|---|
|  RecordingCompressionFormat<br><b>Function RecordingCompressionFormat (Format As RecordingCompressionFormat)</b><br>Default compression format is GSM<br>Members of 'RecordingCompressionFormat' |
| <input type="checkbox"/> GSM_RTP_PAYLOAD<br><input type="checkbox"/> PCM_ADPCM<br><input type="checkbox"/> PCM_RAW<br><input type="checkbox"/> PCMU_RTP_PAYLOAD   |

|  |
|--|
|  StartRecording<br><b>Function StartRecording (bStart As Boolean)</b><br>True to start recording and False to stop. |
|--|

|   |
|---|
| <b>Function RecordCall (CallId As Long) As Boolean</b><br>Record individual call referenced with its Call ID. |
|---|

## Extension Mobility

|  |
|--|
|  ErrorNum<br><b>Property ErrorNum As Long</b><br>Read only returns error code refer below<br><br><b>Error Code Description</b><br><br>0 Unknown Error<br><br>1 XML Validation Error: The request or query was incorrectly formed, and cannot be properly processed<br><br>2 Authentication Error: An error occurred in the authentication process, and the validity of the appID and appPassword submitted cannot be confirmed<br><br>3 Invalid Authentication: The appID and/or appPassword provided are incorrect<br><br>4 Policy Validation Error: Some generic issue regarding determination if the request is allowed or not<br><br>5 Request Denied: The request has been denied (failed Policy Validation) for one or more reasons<br><br>6 Database Error: The Extension Mobility Service received an error while trying to communicate with the database |
|--|

|  |  |
|--|--|
|  | <p>7 AutoLogout Setup Error: Could not communicate with the AutoLogout service</p> <p>8 Query Type Unknown: could not determine whether the query is Device-User or User-Devices</p> <p>9 DirUser Creation Error: Directory integration error</p> <p>10 Proxy Authentication Not Allowed: the appID specified does not have rights to login or logout other users</p> <p>11 Device Does Not Exist: The specified device for login or logout does not exist in the system</p> <p>12 Device Profile Does Not Exist: Either a profile was specified which does not exist or there is no Logout Device Profile configured for the specified user</p> <p>13-17 Various Pipe Errors: Could not communicate with the AutoLogout service</p> <p>18 Device Already Logged In: Could not log in to the specified device because there is already a user logged in</p> <p>19 Device Not Logged In: Could not log out of the specified device because there is no user logged in</p> <p>20 Get Device Hoteling Flag Error: Could not determine whether the device allows Cisco CallManager Extension Mobility (also called "hoteling") or not</p> <p>21 Get Device Hoteling Status Error: Could not determine whether these is a user currently logged in or not</p> <p>22 Device Does Not Allow Extension Mobility: The device specified is not configured for Extension Mobility</p> <p>23 User Does Not Exist: The userid entered for login does not exist in the system</p> <p>24 System Disabled: The Extension Mobility service has been disabled from the service parameters</p> <p>25 User Already Logged In Elsewhere: The login is denied because the specified user is already logged into a different device</p> |
|  | <p> Login</p> <p><b>Function Login(ServerAddress As String, UserId As String, Password As String) As Boolean</b></p> <p>Return</p>  |

|  |  |
|--|--|
|  | True: Succeeds<br>False: Use ErrorNum for error code   |
|  | Logout<br><b>Function Logout(ServerAddress As String, UserId As String, Password As String) As Boolean</b><br>Return<br>True: Succeeds<br>False: Use ErrorNum for error code |

## Event Detail

### DestinationBusy

Event **DestinationBusy(CallId As Long)**

#### Pre-conditions:

1. Call state : Ringing.

---

### DestinationHangUp

Event **DestinationHangUp(CallId As Long)**

#### Pre-conditions:

1. Call state : Active.

---

### DestinationPickUp

Event **DestinationPickUp(CallId As Long)**

#### Pre-conditions:

1. Call state : Ringing.

#### Post-conditions:

1. The call state is Active.

---

### DestinationPickUp

Event **IncomingCall(CallId As Long, line As String, callerId As String)**

**Pre-conditions:**

1. Call state : Ringing.

---

## PeerLost

Event **PeerLost()**

### Pre-conditions:

1. Process, VTGO active.

## Method Detail

### HoldCall

Function **HoldCall()** As Boolean

### Pre-conditions:

1. Active call.

### Post-conditions:

1. The call state is OnHold.
- 

### Conference

Function **Conference(destination As String)** As Boolean

### Pre-conditions:

1. Active call.
2. Call not on hold.
3. Valid destination number.

### Post-conditions:

1. The call state is active.
2. The call not routed to voice mail.

### Parameters:

destination - The destination address string for this conference call.

---

## **MakeCall**

Function **MakeCall**(destination As String) As Boolean

### **Pre-conditions:**

1. Available inactive line.
2. Valid destination number.

### **Post-conditions:**

1. The call state is active.
- 

## **EndCall**

Function **EndCall**() As Boolean

### **Pre-conditions:**

1. Active call.
2. Call not on hold.

### **Post-conditions:**

1. The call state is inactive.
  2. The call does not exist.
- 

## **OffHook**

Sub **OffHook**() As Boolean

### **Pre-conditions:**

1. Device onhook.
- 

## **OnHook**

Sub **OnHook**() As Boolean

### **Pre-conditions:**

1. Device offhook.

---

## SendDTMFKey

Function **SendDTMFKey**(dtmfKey As String) As Boolean

**Pre-conditions:**

1. Active call.
2. Call not on hold.
3. Valid DTMF key. Valid Keys:  
"1","2","3","4","5","6","7","8","8","9","0","\*","#"

**Parameters:**

dtmfKey - The key char string corresponding to the DTMF to be generated.

---

## Transfer

Function **Transfer**(destination As String) As Boolean

**Pre-conditions:**

1. Active call.
2. Call not on hold.
3. Valid destination number.

**Post-conditions:**

1. The call state is transferred and active.

**Parameters:**

destination - The destination address string for this transfer call.

---

## ResumeCall

Function **ResumeCall**(CallId As Long) As Boolean

**Pre-conditions:**

1. The call state must be OnHold
2. Valid destination number.

**Post-conditions:**

1. The call state is active.

**Parameters:**

CallId- Long value representing reference ID of the call.