



Skype4COM

Visual Basic examples



The whole world can talk for free. Skype.com

Answer call automatically

The following example provides automatic call answering.

- Create a Skype4COM object:

```
Set oSkype =
WScript.CreateObject("Skype4COM.Skype", "Skype_")
```
- Start the Skype client:

```
If Not oSkype.Client.IsRunning
    Then oSkype.Client.Start
End If
```
- Connect to the Skype API:

```
oSkype.Attach
```
- Run an infinite loop to ensure the connection remains active:

```
Do While True
    WScript.Sleep(60000)
Loop
```
- The AttachmentStatus event handler monitors attachment status and automatically attempts to reattach to the API following loss of connection:

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status "
    oSkype.Convert.AttachmentStatusToText(aStatus)
If aStatus =
    oSkype.Convert.TextToAttachmentStatus("AVAILABLE")
    Then oSkype.Attach() End If
End Sub
```
- The CallStatus event handler monitors call status and if the status is "ringing" and it is an incoming call, it attempts to answer the call:

```
Public Sub Skype_CallStatus(ByRef aCall, ByVal aStatus)
    WScript.Echo ">Call " & aCall.Id & " status " &
aStatus & " " &
    oSkype.Convert.CallStatusToText(aStatus)
If oSkype.Convert.TextToCallStatus("RINGING") = aStatus
And _
    (oSkype.Convert.TextToCallType("INCOMING_P2P") =
aCall.Type
    Or _
    oSkype.Convert.TextToCallType("INCOMING_PSTN") =
aCall.Type)
    Then
        aCall.Answer()
    End If
End Sub
```

Attach to the Skype API

The following example forms and tests a connection to the Skype API.

- Create a Skype4COM object:

```
Set oSkype = Wscript.CreateObject
("Skype4COM.Skype", "Skype_")
```
- Start the Skype client:

```
If Not oSkype.Client.IsRunning Then
oSkype.Client.Start() End If
```
- Connect to the Skype API with protocol version 5 (the default version):

```
oSkype.Attach 5
```
- When attached to the API, send repeated commands to the API to test if they succeed:

```
Do While True
Wscript.Sleep(1000)
If oSkype.AttachmentStatus =
oSkype.Convert.TextToAttachmentStatus("SUCCESS")
Then
WScript.Stdout.Write "PING.."
oSkype.SendCommand(oSkype.Command
(0,"PING","PONG",TRUE))
oSkype.SendCommand(oSkype.Command(0, "PING"))
WScript.Stdout.Write "ok"
WScript.Echo ""
End If
Loop
```
- The AttachmentStatus event handler monitors attachment status and returns the Skype name of the user when the status is "success":

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)
WScript.Echo ">Attachment status " &
oSkype.Convert.AttachmentStatusToText(aStatus)
If aStatus =
oSkype.Convert.TextToAttachmentStatus("SUCCESS")
Then WScript.Echo "Current user is " &
oSkype.CurrentUserHandle End If
End Sub
```

Use buttons to make a call

The following example emulates pressing buttons in the client to make calls.

- Create a Skype4COM object:
Set oSkype =
Wscript.CreateObject("Skype4COM.Skype", "Skype_")
- Start the Skype client:
If Not oSkype.Client.IsRunning Then
oSkype.Client.Start() End If
- Connect to the Skype API:
oSkype.Attach
- Bring the Skype client into focus and open the Contacts tab:
oSkype.Client.Focus
oSkype.Client.OpenContactsTab()
- Emulate typing "ECHO123" (appears in the address bar or the Skype client):
oSkype.Client.ButtonPressed "E"
oSkype.Client.ButtonReleased "E"
Wscript.Sleep(1000)
oSkype.Client.ButtonPressed "C"
oSkype.Client.ButtonReleased "C"
Wscript.Sleep(1000)
oSkype.Client.ButtonPressed "H"
oSkype.Client.ButtonReleased "H"
Wscript.Sleep(1000)
oSkype.Client.ButtonPressed "O"
oSkype.Client.ButtonReleased "O"
Wscript.Sleep(1000)
oSkype.Client.ButtonPressed "1"
oSkype.Client.ButtonReleased "1"
Wscript.Sleep(1000)
oSkype.Client.ButtonPressed "2"
oSkype.Client.ButtonReleased "2"
Wscript.Sleep(1000)
oSkype.Client.ButtonPressed "3"
oSkype.Client.ButtonReleased "3"
Wscript.Sleep(1000)
- Emulate pressing the Green (YES) button and the client starts a call:
oSkype.Client.ButtonPressed "YES"
oSkype.Client.ButtonReleased "YES"
Wscript.Sleep(10000)
- Emulate pressing the Red (NO) button and the client ends the call:
oSkype.Client.ButtonPressed "NO"
oSkype.Client.ButtonReleased "NO"

- The AttachmentStatus event handler monitors attachment status and automatically attempts to reattach to the API following loss of connection:

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " &
        oSkype.Convert.AttachmentStatusToText(aStatus)
If aStatus =
    oSkype.Convert.TextToAttachmentStatus("AVAILABLE")
    Then oSkype.Attach() End If
End Sub
```

Place a call

The following example places a call.

- Create a Skype4COM object:

```
Set oSkype = WScript.CreateObject("Skype4COM.Skype",  
"Skype_")
```
- Start the Skype client:

```
If Not oSkype.Client.IsRunning Then  
oSkype.Client.Start() End If
```
- Declare the following Skype constants:

```
cUserStatus_Offline =  
oSkype.Convert.TextToUserStatus("OFFLINE")  
cUserStatus_Online =  
oSkype.Convert.TextToUserStatus("ONLINE")  
cCallStatus_Ringing =  
oSkype.Convert.TextToCallStatus("RINGING")  
cCallStatus_Inprogress =  
oSkype.Convert.TextToCallStatus("INPROGRESS")  
cCallStatus_Failed =  
oSkype.Convert.TextToCallStatus("FAILED")  
cCallStatus_Refused =  
oSkype.Convert.TextToCallStatus("REFUSED")  
cCallStatus_Cancelled =  
oSkype.Convert.TextToCallStatus("CANCELLED")  
cCallStatus_Finished =  
oSkype.Convert.TextToCallStatus("FINISHED")  
cCallStatus_Busy =  
oSkype.Convert.TextToCallStatus("BUSY")  
cAttachmentStatus_Available =  
oSkype.Convert.TextToAttachmentStatus("AVAILABLE")
```
- The PlaceCall command will fail if the user is offline. To avoid failure, check user status and change to online if necessary:

```
If cUserStatus_Offline = oSkype.CurrentUserStatus Then  
oSkype.ChangeUserStatus(cUserStatus_Online) End If
```
- Create a user object:

```
Set oUser = oSkype.User("echo123")  
WScript.Echo "User " & oUser.Handle &  
" online status is " &  
oSkype.Convert.OnlineStatusToText  
(oUser.OnlineStatus)
```
- Place a call:

```
Set oCall = oSkype.PlaceCall(oUser.Handle)
```

- Wait until the call is in progress and return the relevant error if any other status occurs:


```
Do While oCall.Status <> cCallStatus_Inprogress
  If oCall.Status = cCallStatus_Failed Or _
  oCall.Status = cCallStatus_Refused Or _
  oCall.Status = cCallStatus_Cancelled Or _
  oCall.Status = cCallStatus_Finished Or _
  oCall.Status = cCallStatus_Busy Then
  Err.Raise vbObjectError + 1, "", "Call status: " &
  oSkype.Convert.CallStatusToText(oCall.Status)
  End If
Wscript.Sleep(500)
Loop
```
- Check DTMF tones for use with interactive voice response (IVR) applications:


```
WScript.Sleep(10000)
  If cCallStatus_InProgress = oCall.Status Then
  oCall.DTMF = "0" End If
  If cCallStatus_InProgress = oCall.Status Then
  oCall.DTMF = "1" End If
  If cCallStatus_InProgress = oCall.Status Then
  oCall.DTMF = "2" End If
  If cCallStatus_InProgress = oCall.Status Then
  oCall.DTMF = "3" End If
  If cCallStatus_InProgress = oCall.Status Then
  oCall.DTMF = "4" End If
  If cCallStatus_InProgress = oCall.Status Then
  oCall.DTMF = "5" End If
  If cCallStatus_InProgress = oCall.Status Then
  oCall.DTMF = "6" End If
  If cCallStatus_InProgress = oCall.Status Then
  oCall.DTMF = "7" End If
  If cCallStatus_InProgress = oCall.Status Then
  oCall.DTMF = "8" End If
  If cCallStatus_InProgress = oCall.Status Then
  oCall.DTMF = "9" End If
  If cCallStatus_InProgress = oCall.Status Then
  oCall.DTMF = "#" End If
  If cCallStatus_InProgress = oCall.Status Then
  oCall.DTMF = "*" End If
```
- Finish the call:


```
If oCall.Status <> cCallStatus_Finished Then
  oCall.Finish() End If
```
- The AttachmentStatus event handler monitors attachment status and automatically attempts to reattach to the API following loss of connection:


```
Public Sub Skype_AttachmentStatus(ByVal aStatus)
  WScript.Echo ">Attachment status " &
  oSkype.Convert.AttachmentStatusToText(aStatus)
  If aStatus = cAttachmentStatus_Available Then
  oSkype.Attach() End If
End Sub
```

- If the call status is "failed", the CallStatus event handler returns the relevant failure reason as text:

```
Public Sub Skype_CallStatus(ByRef aCall, ByVal aStatus)
    WScript.Echo ">Call " & aCall.Id & " status " &
        aStatus & " " &
        oSkype.Convert.CallStatusToText(aStatus)
    If cCallStatus_Failed = aStatus Then
        WScript.Echo ">Failure reason:" &
            oSkype.Convert.CallFailureReasonToText
                (aCall.FailureReason)
        End If
    End Sub
```


Query chat history

The following example queries chat histories.

- Create a Skype4COM object:

```
Set oSkype =
WScript.CreateObject("Skype4COM.Skype", "Skype_")
```
- Start the Skype client:

```
If Not oSkype.Client.IsRunning Then
oSkype.Client.Start() End If
```
- Query the total number of messages in history:

```
WScript.Echo "Total message count:
" & oSkype.Messages.Count
```
- Query the timestamp, name, and display name of all chats in history:

```
WScript.Echo "All chats:"
For Each oChat In oSkype.Chats
WScript.Echo oChat.Timestamp & " " & oChat.Name
& " " & oChat.FriendlyName
Next
```
- Query the timestamp, name, and display name of all active chats:

```
WScript.Echo "Active chats:"
For Each oChat In oSkype.ActiveChats
WScript.Echo oChat.Timestamp & " " & oChat.Name
& " " & oChat.FriendlyName
Next
```
- Query the timestamp, name, and display name of all missed chats:

```
WScript.Echo "Missed chats:"
For Each oChat In oSkype.MissedChats
WScript.Echo oChat.Timestamp & " " & oChat.Name
& " " & oChat.FriendlyName
Next
```
- Query the timestamp, name, and display name of all recent chats:

```
WScript.Echo "Recent chats:"
For Each oChat In oSkype.RecentChats
WScript.Echo oChat.Timestamp & " " & oChat.Name
& " " & oChat.FriendlyName
Next
```
- Query the timestamp, name, and display name of all bookmarked chats:

```
WScript.Echo "Bookmarked chats:"
For Each oChat In oSkype.BookmarkedChats
WScript.Echo oChat.Timestamp & " " & oChat.Name
& " " & oChat.FriendlyName
Next
```
- The AttachmentStatus event handler monitors attachment status and automatically attempts to reattach to the API following loss of connection:

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)
WScript.Echo ">Attachment status " &
oSkype.Convert.AttachmentStatusToText(aStatus)
If aStatus =
oSkype.Convert.TextToAttachmentStatus("AVAILABLE") Then
oSkype.Attach() End If
End Sub
```

Managing the Skype client user interface

The following example includes a variety of commands to manage the user interface.

- Create a Skype4COM object:
Set oSkype =
WScript.CreateObject("Skype4COM.Skype", "Skype_")
- Start the Skype client:
If Not oSkype.Client.IsRunning Then
oSkype.Client.Start() End If
- Bring the client into focus:
oSkype.Client.Focus
WScript.Sleep(1000)
- Minimize the client to the system tray:
oSkype.Client.Minimize
WScript.Sleep(1000)
- Open the Add Contacts window and enter "echo123" in the contact field:
oSkype.Client.OpenAddContactDialog "echo123"
WScript.Sleep(1000)
- Open a Chat window with "echo123" and type "Hi" in the message pane:
oSkype.Client.OpenMessageDialog "echo123", "Hi"
WScript.Sleep(1000)
- Transfer a file to "echo123"
oSkype.Client.OpenFileTransferDialog "echo123", "c:\temp"
WScript.Sleep(1000)
- Open other windows as required, with the following commands:
oSkype.Client.OpenProfileDialog
WScript.Sleep(1000)
oSkype.Client.OpenUserInfoDialog "echo123"
WScript.Sleep(1000)
oSkype.Client.OpenConferenceDialog
WScript.Sleep(1000)
oSkype.Client.OpenSearchDialog
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("GENERAL")
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("PRIVACY")
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("NOTIFICATIONS")
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("SOUNDALERTS")
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("SOUNDDEVICES")
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("HOTKEYS")
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("CONNECTION")
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("VOICEMAIL")
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("CALLFORWARD")
WScript.Sleep(1000)

```
oSkype.Client.OpenOptionsDialog("VIDEO")
Wscript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("ADVANCED")
Wscript.Sleep(1000)
oSkype.Client.OpenCallHistoryTab
Wscript.Sleep(1000)
oSkype.Client.OpenContactsTab
Wscript.Sleep(1000)
oSkype.Client.OpenDialpadTab
Wscript.Sleep(1000)
oSkype.Client.OpenSendContactsDialog "echo123"
Wscript.Sleep(1000)
oSkype.Client.OpenBlockedUsersDialog
Wscript.Sleep(1000)
oSkype.Client.OpenImportContactsWizard
Wscript.Sleep(1000)
oSkype.Client.OpenGettingStartedWizard
Wscript.Sleep(1000)
oSkype.Client.OpenAuthorizationDialog "echo123"
Wscript.Sleep(1000)
oSkype.Client.OpenVideoTestDialog
WScript.Sleep(1000)
```

- The AttachmentStatus event handler monitors attachment status and automatically attempts to reattach to the API following loss of connection:

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " &
oSkype.Convert.AttachmentStatusToText(aStatus)
If aStatus =
oSkype.Convert.TextToAttachmentStatus("AVAILABLE") Then
oSkype.Attach() End If
End Sub
```

Blocking and non-blocking commands

The following example includes blocking and non-blocking commands.

- Create a Skype4COM object:

```
Set oSkype =
WScript.CreateObject("Skype4COM.Skype", "Skype_")
```
- Start the Skype client:

```
If Not oSkype.Client.IsRunning Then
oSkype.Client.Start() End If
```
- Connect to the Skype API:

```
oSkype.Attach
```

Skype4COM can send two types of command; blocking commands and non-blocking commands. Blocking is defined as a Boolean variant of the command object and the default value is False, non-blocking. A blocking command awaits a reply before continuing. A non-blocking command does not require a reply.

- Send the following non-blocking commands:

```
oSkype.SendCommand(oSkype.Command(0, "FOCUS"))
oSkype.SendCommand(oSkype.Command
(1, "GET AUDIO_IN", "AUDIO_IN"))
oSkype.SendCommand(oSkype.Command
(2, "GET AUDIO_OUT", "AUDIO_OUT"))
WScript.Sleep(1000)
```
- Send the following blocking commands (note the True variant for each command):

```
oSkype.SendCommand(oSkype.Command(3, "GET
CURRENTUSERHANDLE", "CURRENTUSERHANDLE", True))
oSkype.SendCommand(oSkype.Command(4, "GET USER echo123
FULLNAME", "USER echo123 FULLNAME", True))
oSkype.SendCommand(oSkype.Command(5, "GET USER echo123
DISPLAYNAME", "USER echo123 DISPLAYNAME", True))
oSkype.SendCommand(oSkype.Command(6, "GET USER echo123
COUNTRY", "USER echo123 COUNTRY", True))
WScript.Sleep(1000)
```

As with command objects, search commands can be defined as blocking or non-blocking, and the default value is False, non-blocking. A blocking search requires a response before activities continue.

- Following is a blocking search (note the True (blocking) variant):

```
Set oBlockingSearch = oSkype.Command(8888, "SEARCH USERS
echo123", "USERS ", True)
oSkype.SendCommand(oBlockingSearch)
WScript.Echo "Search Result:" & oBlockingSearch.Reply
```
- A non-blocking search does not require a response:

```
Set oSearchCommand = oSkype.Command(9999, "SEARCH USERS
john doe", "USERS ")
oSkype.SendCommand(oSearchCommand)
WScript.Echo "Sleeping ..."
WScript.Sleep(30000)
```
-

- The AttachmentStatus event handler monitors attachment status and automatically attempts to reattach to the API following loss of connection:

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " &
    oSkype.Convert.AttachmentStatusToText(aStatus)
    If aStatus =
    oSkype.Convert.TextToAttachmentStatus("AVAILABLE") Then
    oSkype.Attach() End If
End Sub
```

Create a conference call

The following example creates a conference call.

- Create a Skype4COM object:

```
Set oSkype = WScript.CreateObject("Skype4COM.Skype",  
"Skype_")
```
- Start the Skype client:

```
If Not oSkype.Client.IsRunning Then  
oSkype.Client.Start() End If
```
- The PlaceCall command will fail if the user is offline. To avoid failure, check user status and change to online if necessary:

```
If oSkype.Convert.TextToUserStatus("OFFLINE") =  
oSkype.CurrentUserStatus Then  
oSkype.ChangeUserStatus  
(oSkype.Convert.TextToUserStatus("ONLINE"))  
End If
```
- To create a conference call, use the PlaceCall command and add multiple users to the list of targets (to the maximum number of conference users supported by the OS environment):

```
Set oFirstCall = oSkype.PlaceCall("skypeuser1",  
"skypeuser2")  
WScript.Sleep(30000)
```
- The AttachmentStatus monitors and returns call status:

```
Public Sub Skype_CallStatus(ByRef aCall, ByVal aStatus)  
WScript.Echo ">Call " & aCall.Id & " status " & aStatus  
& " " & oSkype.Convert.CallStatusToText(aStatus)  
End Sub
```
- The AttachmentStatus event handler monitors attachment status and automatically attempts to reattach to the API following loss of connection:

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)  
WScript.Echo ">Attachment status " &  
oSkype.Convert.AttachmentStatusToText(aStatus)  
If aStatus =  
oSkype.Convert.TextToAttachmentStatus("AVAILABLE")  
Then oSkype.Attach() End If  
End Sub
```

Query conference calls

- Create a Skype4COM object:

```
Set oSkype =
WScript.CreateObject("Skype4COM.Skype","Skype_")
```
- Start the Skype client:

```
If Not oSkype.Client.IsRunning Then
oSkype.Client.Start() End If
```
- Count conference calls hosted by the current user. A conference call hosted by the current user is represented by multiple call objects which have a common conference ID. For each conference object, return the `conf_id`, the number of calls, and the number of active calls:

```
WScript.Echo "There are " & oSkype.Conferences.Count & "
conference calls hosted by " & oSkype.CurrentUser.Handle
For Each oConf In oSkype.Conferences
    WScript.Echo "Conference " & oConf.Id & " includes "
    & oConf.Calls.Count & " calls from which " &
    oConf.ActiveCalls.Count & " are active."
Next
```
- To list conference calls in which the current user is active but not the host, begin by listing active calls for a user:

```
WScript.Echo "There are total " &
oSkype.ActiveCalls.Count & " active calls."
```
- Conference calls not hosted by the current user are all calls which have one or more participants (participants do not include the current user or the conference host). For each active call, return the Skype name and display name of each participant:

```
For Each oCall In oSkype.ActiveCalls
If oCall.Participants.Count > 0 Then
    WScript.Echo "Conference call " & oCall.Id & "
    with " & oCall.PartnerHandle & " has " &
    oCall.Participants.Count & " participants."
    For Each oParticipant In oCall.Participants
        WScript.Echo "Participant " &
        oParticipant.Handle & " display name is " &
        oParticipant.DisplayName
    Next
Else
    WScript.Echo "Active call " & oCall.Id & " with " &
    oCall.PartnerHandle & " is not a conference call."
End If
Next
```
- The `AttachmentStatus` event handler monitors attachment status and automatically attempts to reattach to the API following loss of connection:

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " &
    oSkype.Convert.AttachmentStatusToText(aStatus)
    If aStatus =
    oSkype.Convert.TextToAttachmentStatus("AVAILABLE") Then
    oSkype.Attach() End If
End Sub
```

Managing contact groups

The following example provides commands to manage contact groups.

- Create a Skype4COM object:

```
Set oSkype =
WScript.CreateObject("Skype4COM.Skype", "Skype_")
```
- Start the Skype client:

```
If Not oSkype.Client.IsRunning Then
oSkype.Client.Start() End If
```
- The following commands create a group, add a user, and then query this group. However, this information will not be updated in the Skype4COM internal cache until a little later. To avoid delay, temporarily switch off the cache:

```
oSkype.Cache = False
```
- Create a custom group:

```
Set oMyGroup = oSkype.CreateGroup("MyGroup")
```
- Add user "echo123" to the group:

```
oMyGroup.AddUser "echo123"
```
- List the users in MyGroup and return the Skype name of each user:

```
WScript.Echo "Group (" & oMyGroup.Id & ") labelled [" &
oMyGroup.DisplayName & "] has " & oMyGroup.Users.Count &
" users."
For Each oUser In oMyGroup.Users
    WScript.Echo oUser.Handle
    & " (" & oUser.FullName & ")"
Next
```
- Remove a user from a group:

```
oMyGroup.RemoveUser "echo123"
```
- Delete a custom group:

```
oSkype.DeleteGroup oMyGroup.Id
```
- Switch the cache on:

```
oSkype.Cache = True
```
- List all groups:

```
WScript.Echo
"There are total " & oSkype.Groups.Count & " groups
(" & _ &
oSkype.HardwiredGroups.Count & " hardwired)"
& vbCrLf
```
- List all custom groups:

```
WScript.Echo "Custom groups are:" & vbCrLf
For Each oGroup In oSkype.CustomGroups
    WScript.Echo "Group (" & oGroup.Id & ") labelled [" &
oGroup.DisplayName & "] has " & oGroup.Users.Count
& " users."
For Each oUser In oGroup.Users
    WScript.Echo oUser.Handle & "
("& oUser.FullName & ")"
Next
WScript.Echo ""
Next
```


- List all hardwired groups and return the Skypename and full name of each member of each group:

```
WScript.Echo "Hardwired groups are:" & vbCrLf
For Each oGroup In oSkype.HardwiredGroups
    WScript.Echo "Group (" & oGroup.Id & ") type of [" &
        oSkype.Convert.GroupTypeToText(oGroup.Type) & "] has
        " & oGroup.Users.Count & " users."
For Each oUser In oGroup.Users
    WScript.Echo oUser.Handle & "
        (" & oUser.FullName & ")"
Next
    WScript.Echo ""
Next
```

- Keep the script running for 60 seconds:

```
WScript.Sleep(60000)
```

- The GroupVisible event handler returns information about whether a group is visible or hidden:

```
Public Sub Skype_GroupVisible
    (ByRef aGroup, ByVal aVisible)
WScript.Stdout.Write "Group (" & aGroup.Id & ") type of
[" & oSkype.Convert.GroupTypeToText(aGroup.Type) & "]
labelled [" & aGroup.DisplayName & "]"
If aVisible Then
    WScript.Echo " is visible."
Else
    WScript.Echo " is hidden."
End If
End Sub
```

- The GroupExpanded event handler returns information about whether a group is expanded or collapsed:

```
Public Sub Skype_GroupExpanded
    (ByRef aGroup, ByVal aExpanded)
WScript.Stdout.Write "Group (" & aGroup.Id & ") type of
[" & oSkype.Convert.GroupTypeToText(aGroup.Type) & "]
labelled [" & aGroup.DisplayName & "]"
If aExpanded Then
    WScript.Echo " is expanded."
Else
    WScript.Echo " is collapsed."
End If
End Sub
```

- The GroupUsers event handler gets the Skypenames of members of a group:

```
Public Sub Skype_GroupUsers(ByRef aGroup, ByRef aUsers)
WScript.Stdout.Write "Group (" & aGroup.Id & ") type of
[" & oSkype.Convert.GroupTypeToText(aGroup.Type) & "]
labelled [" & aGroup.DisplayName & "]"
Dim oUser
For Each oUser In aUsers
    WScript.Stdout.Write " " & oUser.Handle
Next
    WScript.Echo ""
End Sub
```

-

- Delete a group:

```
Public Sub Skype_GroupDeleted(ByRef aGroupId)
WScript.Echo "Group " & aGroupId & " deleted."
End Sub
```
- Bring a contact into focus:

```
Public Sub Skype_ContactsFocused(ByVal aHandle)
If Len(aHandle) Then
    WScript.Echo "Contact " & aHandle & " focused."
Else
    WScript.Echo "No contact focused."
End If
End Sub
```
- The AttachmentStatus event handler monitors attachment status and automatically attempts to reattach to the API following loss of connection:

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)
WScript.Echo ">Attachment status " &
oSkype.Convert.AttachmentStatusToText(aStatus)
If aStatus =
oSkype.Convert.TextToAttachmentStatus("AVAILABLE") Then
oSkype.Attach() End If
End Sub
```

Join to a conference call

The following example joins the user to a conference call.

- Create a Skype4COM object:

```
Set oSkype =
WScript.CreateObject("Skype4COM.Skype", "Skype_")
```
- Start the Skype client:

```
If Not oSkype.Client.IsRunning Then
oSkype.Client.Start() End If
```
- Declare the following Skype constants:

```
cUserStatus_Offline =
oSkype.Convert.TextToUserStatus("OFFLINE")
cUserStatus_Online =
oSkype.Convert.TextToUserStatus("ONLINE")
cCallStatus_Ringing =
oSkype.Convert.TextToCallStatus("RINGING")
cCallStatus_Inprogress =
oSkype.Convert.TextToCallStatus("INPROGRESS")
cCallStatus_Failed =
oSkype.Convert.TextToCallStatus("FAILED")
cCallStatus_Refused =
oSkype.Convert.TextToCallStatus("REFUSED")
cCallStatus_Cancelled =
oSkype.Convert.TextToCallStatus("CANCELLED")
cCallStatus_Finished =
oSkype.Convert.TextToCallStatus("FINISHED")
cCallStatus_Busy =
oSkype.Convert.TextToCallStatus("BUSY")
```
- The PlaceCall command will fail if the user is offline. To avoid failure, check user status and change to online if necessary:

```
If cUserStatus_Offline = oSkype.CurrentUserStatus Then
oSkype.ChangeUserStatus(cUserStatus_Online) End If
```
- Place a call (the first of more than one):

```
Set oCall1 = oSkype.PlaceCall("user1")
```
- Wait until the first call is in progress and return the relevant error if any other status occurs:

```
Do While oCall1.Status <> cCallStatus_Inprogress
If oCall1.Status = cCallStatus_Failed Or _
oCall1.Status = cCallStatus_Refused Or _
oCall1.Status = cCallStatus_Cancelled Or _
oCall1.Status = cCallStatus_Finished Or _
oCall1.Status = cCallStatus_Busy Then
Err.Raise vbObjectError + 1, "", "Call "
& _ oCall1.Id & " " oSkype.Convert.CallStatusToText
(oCall1.Status)
End If
Wscript.Sleep(500)
Loop
```
- Place the second call:

```
Set oCall2 = oSkype.PlaceCall("user2")
```
-

- Wait for the second call to be "in progress" and return the relevant error if any other status occurs:

```

Do While oCall2.Status <> cCallStatus_Inprogress
If oCall2.Status = cCallStatus_Failed Or _
    oCall2.Status = cCallStatus_Refused Or _
    oCall2.Status = cCallStatus_Cancelled Or _
    oCall2.Status = cCallStatus_Finished Or _
    oCall2.Status = cCallStatus_Busy Then
    Err.Raise vbObjectError + 1, "", "Call "
    & _ oCall2.Id & " " &
    oSkype.Convert.CallStatusToText(oCall2.Status)
End If
Wscript.Sleep(500)
Loop

```

- Join these two calls to a conference:

```

WScript.Echo "Join call " & oCall2.Id & " to call " &
oCall1.Id
oCall2.Join oCall1.Id
WScript.Sleep(60000)

```

- The AttachmentStatus event handler monitors attachment status and automatically attempts to reattach to the API following loss of connection:

```

Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " &
    oSkype.Convert.AttachmentStatusToText(aStatus)
If aStatus =
    oSkype.Convert.TextToAttachmentStatus("AVAILABLE")
    Then oSkype.Attach() End If
End Sub

```

- The CallStatus event handler monitors call status and returns status changes as text:

```

Public Sub Skype_CallStatus(ByRef oCall, ByVal Status)
    WScript.Echo ">Call " & oCall.Id & " status "
    & Status & " " &
    oSkype.Convert.CallStatusToText(Status)
End Sub

```

Create a multichat

The following example creates a multichat.

- Create a Skype4COM object:

```
Set oSkype =
WScript.CreateObject("Skype4COM.Skype", "Skype_")
```
- Start the Skype client:

```
If Not oSkype.Client.IsRunning Then
oSkype.Client.Start() End If
```
- Create a UserCollection object and add the multichat users to this collection:

```
Set oMembers = CreateObject("Skype4COM.UserCollection")
oMembers.Add(oSkype.User("skypeuser2"))
oMembers.Add(oSkype.User("skypeuser3"))
```
- Create a multichat:

```
Set oChat = oSkype.CreateChatMultiple(oMembers)
```
- Open a chat window:

```
oChat.OpenWindow
```
- Set a chat topic:

```
oChat.Topic = "Skype is cool"
```
- Send a chat message:

```
Set oMsg = oChat.SendMessage("Greetings!")
```
- Leave a chat:

```
oChat.Leave
```
- The AttachmentStatus event handler monitors attachment status and automatically attempts to reattach to the API following loss of connection:

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)
WScript.Echo ">Attachment status " &
oSkype.Convert.AttachmentStatusToText(aStatus)
If aStatus =
oSkype.Convert.TextToAttachmentStatus("AVAILABLE") Then
oSkype.Attach() End If
End Sub
```

Manage profile

The following example manages the user profile.

- Create a Skype4COM object:

```
Set oSkype =
WScript.CreateObject("Skype4COM.Skype", "Skype_")
```
- Start the Skype client:

```
If Not oSkype.Client.IsRunning Then
oSkype.Client.Start() End If
```
- List the profile properties for the user:

```
WScript.Echo "Full name: " &
oSkype.CurrentUserProfile.FullName
WScript.Echo "Birthday: " &
oSkype.CurrentUserProfile.Birthday
WScript.Echo "Sex: " & oSkype.CurrentUserProfile.Sex
WScript.Echo "Languages: " &
oSkype.CurrentUserProfile.Languages
WScript.Echo "Country: " &
oSkype.CurrentUserProfile.Country
WScript.Echo "Province: " &
oSkype.CurrentUserProfile.Province
WScript.Echo "City: " & oSkype.CurrentUserProfile.City
WScript.Echo "Phone home: " &
oSkype.CurrentUserProfile.PhoneHome
WScript.Echo "Phone office: " &
oSkype.CurrentUserProfile.PhoneOffice
WScript.Echo "Phone mobile: " &
oSkype.CurrentUserProfile.PhoneMobile
WScript.Echo "Homepage: " &
oSkype.CurrentUserProfile.Homepage
WScript.Echo "About: " & oSkype.CurrentUserProfile.About
WScript.Echo "Mood: " &
oSkype.CurrentUserProfile.MoodText
WScript.Echo "Timezone: " &
oSkype.CurrentUserProfile.Timezone
WScript.Echo "Noanswer timeout: " &
oSkype.CurrentUserProfile.CallNoAnswerTimeout
WScript.Echo "Forward rules: " &
oSkype.CurrentUserProfile.CallForwardRules
WScript.Echo "Call apply forward: " &
oSkype.CurrentUserProfile.CallApplyCF
WScript.Echo "Call send to voicemail: " &
oSkype.CurrentUserProfile.CallSendToVM
WScript.Echo "Balance: " &
oSkype.CurrentUserProfile.Balance / 100 & " "&
oSkype.CurrentUserProfile.BalanceCurrency
WScript.Echo "Balance Text: " &
oSkype.CurrentUserProfile.BalanceToText
```

- The AttachmentStatus event handler monitors attachment status and automatically attempts to reattach to the API following loss of connection:

```
Public Sub Skype_AttachmentStatus (ByVal aStatus)
    WScript.Echo ">Attachment status " &
    oSkype.Convert.AttachmentStatusToText (aStatus)
    If aStatus =
    oSkype.Convert.TextToAttachmentStatus ("AVAILABLE") Then
    oSkype.Attach () End If
End Sub
```

Create an autoresponse

The following example creates an autoresponse.

- Create a Skype4COM object:

```
Set oSkype = WScript.CreateObject("Skype4COM.Skype",
"Skype_")
```
- Start the Skype client:

```
If Not oSkype.Client.IsRunning Then
oSkype.Client.Start() End If
```
- Declare the following Skype constants:

```
cAttachmentStatus_Available =
oSkype.Convert.TextToAttachmentStatus("AVAILABLE")
cMessageStatus_Sending =
oSkype.Convert.TextToChatMessageStatus("SENDING")
cMessageStatus_Received =
oSkype.Convert.TextToChatMessageStatus("RECEIVED")
cMessageType_Said =
oSkype.Convert.TextToChatMessageType("SAID")
cMessageType_Left =
oSkype.Convert.TextToChatMessageType("LEFT")
```
- The PlaceCall command will fail if the user is offline. To avoid failure, check user status and change to online if necessary:

```
If cUserStatus_Offline = oSkype.CurrentUserStatus
Then oSkype.ChangeUserStatus(cUserStatus_Online)
End If
Sleep
Do While True
    Wscript.Sleep(60000)
Loop
```
- The AttachmentStatus event handler monitors attachment status and attempts to connect to the Skype API:

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " &
    oSkype.Convert.AttachmentStatusToText(aStatus)
If aStatus = cAttachmentStatus_Available Then
    oSkype.Attach() End If
End Sub
```
-

- The `MessageStatus` event handler monitors message status, decodes received messages and, for those of type "Said", sends an autoresponse quoting the original message:

```
Public Sub Skype_MessageStatus
    (ByRef aMsg, ByVal aStatus)
        WScript.Echo ">Message " & aMsg.Id & " status " &
            oSkype.Convert.ChatMessageStatusToText(aStatus)
    If aStatus = cMessageStatus_Received Then
        DecodeMsg aMsg
        If aMsg.Type = cMessageType_Said Then
            oSkype.SendMessage aMsg.FromHandle,
                "This is autoresponse."
            aMsg.Chat.SendMessage "You said
                [" & aMsg.Body & "]"
            End If
        End If
    End Sub
```

- The `DecodeMsg` event handler decodes messages in a chat and converts leave reasons to text for messages of type "Left":

```
Public Sub DecodeMsg(ByRef oMsg)
    sText = oMsg.FromHandle & " " &
        oSkype.Convert.ChatMessageTypeToText(oMsg.Type)
    & ":"
    If len(oMsg.Body) Then
        sText = sText & " " & oMsg.Body
    End If
    Dim oUser
    For Each oUser In oMsg.Users
        sText = sText & " " & oUser.Handle
    Next
    If oMsg.Type = cMessageType_Left Then
        sText = sText & " " &
            oSkype.Convert.ChatLeaveReasonToText
                (oMsg.LeaveReason)
    End If
    WScript.Echo ">" & sText
End Sub
```

Searching

The following example demonstrates search commands.

- Create a Skype4COM object:

```
Set oSkype =  
WScript.CreateObject("Skype4COM.Skype", "Skype_")
```
- Start the Skype client:

```
If Not oSkype.Client.IsRunning Then  
oSkype.Client.Start() End If
```
- Search for users John Doe and return the Skypname, full name and country of each matching user:

```
Set oUsers = oSkype.SearchForUsers("john doe")  
For Each oUser In oUsers  
    WScript.Echo oUser.Handle &" "&oUser.Fullname &" "&  
    oUser.Country  
Next
```
- The AttachmentStatus event handler monitors attachment status and automatically attempts to reattach to the API following loss of connection:

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)  
    WScript.Echo ">Attachment status " &  
    oSkype.Convert.AttachmentStatusToText(aStatus)  
    If aStatus =  
oSkype.Convert.TextToAttachmentStatus("AVAILABLE") Then  
oSkype.Attach() End If  
End Sub
```

Query Skype general parameters

The following example queries Skype general parameters.

- Create a Skype4COM object:

```
Set oSkype =  
WScript.CreateObject("Skype4COM.Skype", "Skype_")
```
- Start the Skype client:

```
If Not oSkype.Client.IsRunning Then  
oSkype.Client.Start() End If
```
- Query Skype general parameters:

```
WScript.Echo "Audio in:" & oSkype.Settings.AudioIn  
WScript.Echo "Audio out:" & oSkype.Settings.AudioOut  
WScript.Echo "Auto away:" & oSkype.Settings.AutoAway  
WScript.Echo "Ringer:" & oSkype.Settings.Ringer  
'oSkype.Settings.Avatar = "c:\temp\test.jpg"  
WScript.Echo "Ringtone status:" &  
oSkype.Settings.RingToneStatus  
'oSkype.Settings.RingTone = "c:\temp\test.mp3"  
WScript.Echo "Video in:" & oSkype.Settings.VideoIn  
WScript.Echo "PC speaker:" & oSkype.Settings.PCSpeaker  
WScript.Echo "AGC:" & oSkype.Settings.AGC  
WScript.Echo "AEC:" & oSkype.Settings.AEC  
WScript.Echo "Language:" & oSkype.Settings.Language
```
- The AttachmentStatus event handler monitors attachment status and automatically attempts to reattach to the API following loss of connection:

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)  
    WScript.Echo ">Attachment status " &  
        oSkype.Convert.AttachmentStatusToText(aStatus)  
    If aStatus =  
        oSkype.Convert.TextToAttachmentStatus("AVAILABLE")  
    Then oSkype.Attach() End If  
End Sub
```