

Piloter Skype en VB6 ou VBScript (Skype4com.Dll)

par bbil ([La page à bbil](#))

Date de publication : 12/12/2007

Dernière mise à jour :

Grâce à la Dll Skype4com, il devient possible de piloter Skype. Vous trouverez dans cet article quelques exemples d'utilisation.

I - Introduction.....	3
II - Intégration de Skype4Com dans un projet.....	3
II-A - En utilisant les références en VB6.....	3
II-B - En utilisant Create_Object en VBScript.....	3
III - Sécurité.....	4
IV - Gestion des amis.....	6
IV-A - Lister les amis.....	6
IV-B - Effacer un ami.....	7
IV-C - Recherche d'un nouvel ami.....	7
IV-D - Ajout d'un ami à la liste de contacts.....	8
V - Evènements.....	9
V-A - Documentation.....	9
V-B - MessageStatus.....	9
V-C - CallStatus.....	10
VI - Envoi de commande "directe".....	12
VI-A - Envoi de commande en VB6.....	12
VI-B - Envoi de commande en VBS.....	13
VII - Téléchargement.....	13
VIII - Conclusion.....	14
VIII-A - "à vous de jouer".....	14
VIII-B - Remerciements.....	14

I - Introduction

Skype, l'une des solutions de VoIP les plus connues, met à disposition une API dont la documentation est disponible ici :

 <https://developer.skype.com/Docs/ApiDoc>

Dans ce document nous utiliserons l'interface "com", représentée par Skype4com.dll qui encapsule les appels vers cette API...

Celle-ci est téléchargeable sur la zone développeur du site de Skype : <https://developer.skype.com/Download>, de plus elle est habituellement installée simultanément au client "Skype". Nous aurons aussi besoin d'un compte Skype (et d'amis possédant d'autres comptes Skype...).

II - Intégration de Skype4Com dans un projet

II-A - En utilisant les références en VB6

En VB6, on peut utiliser les références. Pour cela il suffit d'ajouter la dll au projet en utilisant le menu projet/propriétés/références et d'ajouter alors "Skype4Com librairie".

La création de l'objet skype peut alors être effectuée au démarrage de l'application, par exemple grâce à l'utilisation de l'opérateur new.

Par exemple dans un module :

en VB6

```
Public oSkype As New Skype
Sub init_Skype()
    If Not oSkype.Client.IsRunning Then
        oSkype.Client.Start
    End If
    While Not oSkype.Client.IsRunning
        DoEvents 'Attente démarrage client Skype
    Wend
    'Appel par exemple de la fenêtre profil de l'utilisateur connecté sur Skype :
    oSkype.Client.OpenDialog ("PROFILE")
End Sub
```

II-B - En utilisant Create_Object en VBScript

En VBS l'utilisation de la fonction Create_Object s'impose :

en VBScript

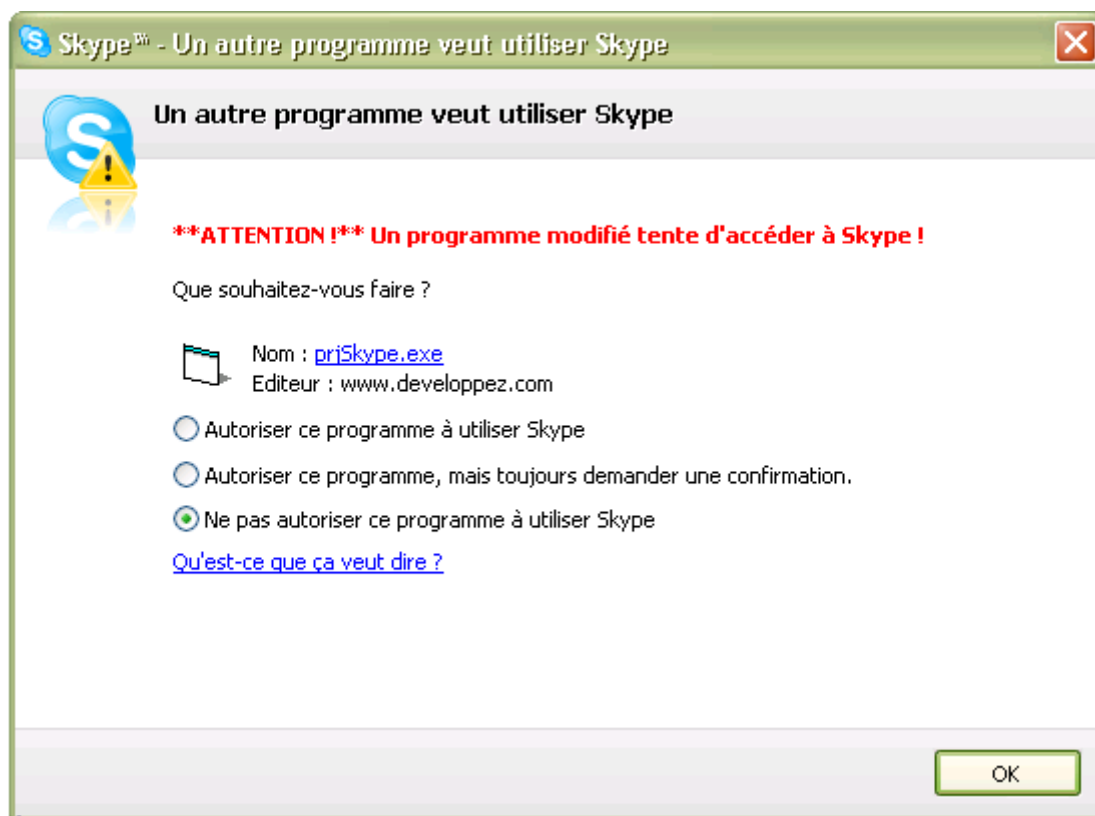
```
'-----
Dim oSkype 'As Skype 'L'objet public Sype

init_Skype
'Appel par exemple de la fenêtre profil de l'utilisateur connecté sur Skype :
oSkype.Client.OpenDialog ("PROFILE")
'-----
' Init_Skype :
' Fonction d'initialisation Skype permet de lancer le client skype
' si celui-ci ne l'est pas et effectue la "connection" du projet à l'API Skype.
'-----
Sub init_Skype()
    Set oSkype = WScript.CreateObject("Skype4COM.Skype", "")
    If Not oSkype.Client.IsRunning Then
        oSkype.Client.Start
    End If
    While Not oSkype.Client.IsRunning
        DoEvents 'Attente démarrage client Skype
    Wend
End Sub
```

en VBScript

III - Sécurité

Lors du premier lancement d'un programme faisant appel à l'API Skype, une fenêtre "sécurité" est affichée :

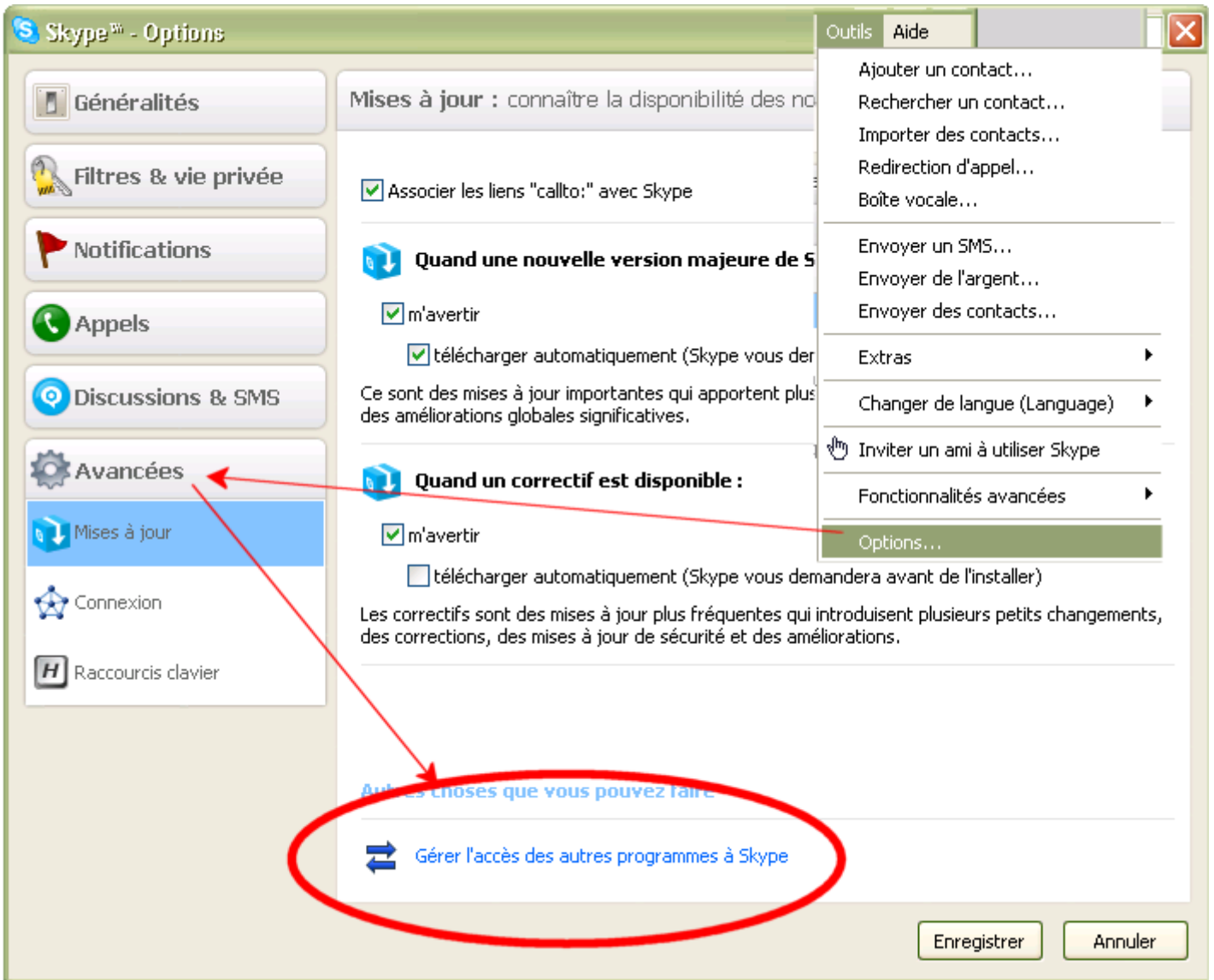


Fenêtre sécurité Skype

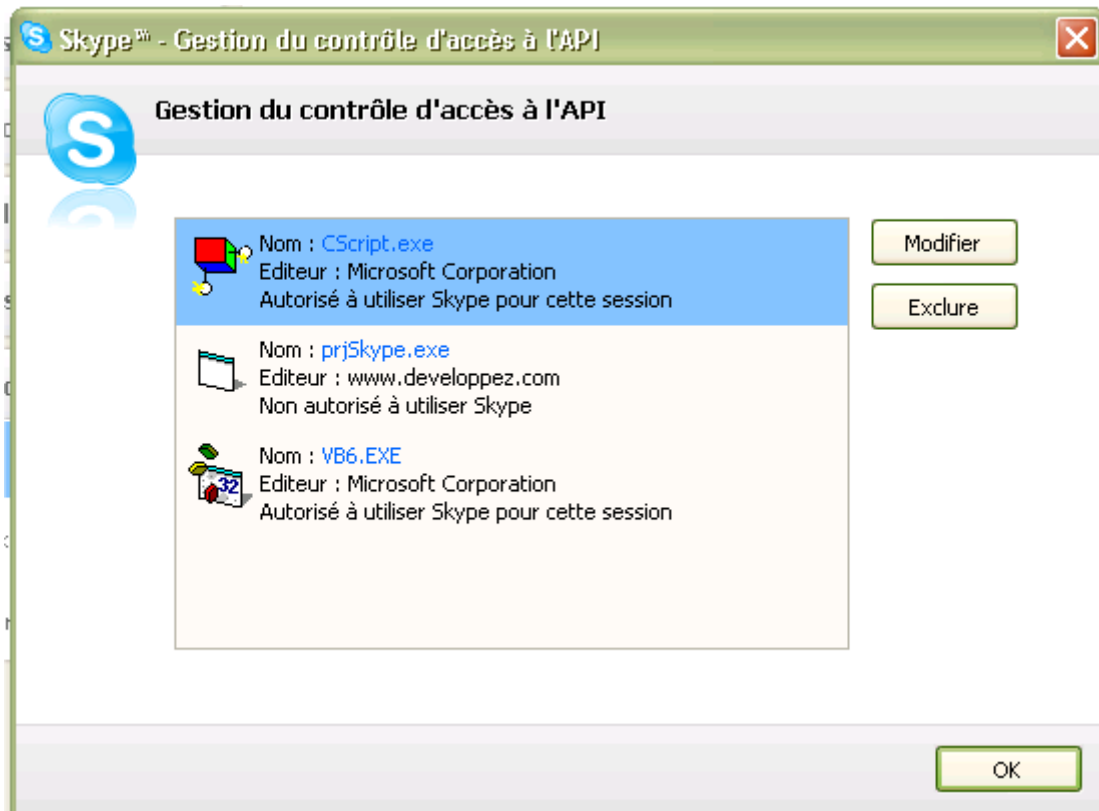
Trois choix s'offrent à nous

- 1 "Autoriser ce programme à utiliser Skype : " qui permet de donner l'autorisation 'définitive' d'accès à l'application ; si cette option est choisie la fenêtre "sécurité" n'apparaîtra plus aux prochains lancements du programme
- 2 "Autoriser ce programme, mais toujours demander une confirmation : " l'accès à l'API Skype est autorisé pour le lancement courant de l'application, la fenêtre "sécurité" sera à nouveau affichée au prochain lancement de l'application.
- 3 "Ne pas autoriser ce programme à utiliser Skype ." Permet de refuser de façon "définitive' l'accès à Skype par l'application citée.

Les choix 1 et 3 dit "définitifs" peuvent toutefois être modifiés. Pour cela, il faut utiliser menu Outils/Options du client Skype puis l'onglet avancées et enfin le lien "Gérer l'accès des autres programmes à skype"



La fenêtre "Gestion du contrôle d'accès à l'API" est alors affichée :



elle permet de modifier les droits d'accès des diverses applications à l'API Skype, grâce au bouton modifier, le bouton exclure permettant d'enlever une application de la liste.

IV - Gestion des amis


IV-A - Lister les amis

L'objet Skype nous donne accès à la collection "friends" qui regroupe la liste des contacts (user) du profil courant, collection que l'on peut parcourir grâce à une boucle For Each. Par exemple : en VB6, avec une listbox lstUser permettant d'afficher cette liste d'amis

Lecture liste d'amis en VB6

```

Private Sub Form_Load()
'
' Mise à jour liste des utilisateurs
'
Dim oFriend As User
lstUser.Clear
For Each oFriend In oSkype.Friends
    lstUser.AddItem oFriend.Handle
Next
End Sub
    
```

et ensuite l'on peut afficher diverses propriétés de l'utilisateur sélectionné (voir :  <https://developer.skype.com/Docs/Skype4COMLib/IUser>) :

```

Private Sub lstUser_Click()
Dim i As Integer
If lstUser.ListIndex >= 0 Then
    i = lstUser.ListIndex + 1
    
```

```

lblFullName = oSkype.Friends(i).FullName
lblBirthday = oSkype.Friends(i).Birthday
lblAbout = oSkype.Friends(i).About
lblRichMoodText = oSkype.Friends(i).RichMoodText
lblOnlineStatus = oSkype.Friends(i).OnlineStatus

End If
End Sub
    
```

En VBS (code simplifié en considérant que le client Skype est déjà en route...)

Lecture liste d'amis en VBScript

```

Set oSkype = WScript.CreateObject("Skype4COM.Skype", "")
'Liste des Amis :
For Each oFriend in oSkype.friends
    wScript.Echo oFriend.Handle & vbTab & oFriend.Homepage & vbTab & oFriend.Country &
    oFriend.LastOnline
next
    
```

IV-B - Effacer un ami

Pour enlever un "ami" de notre liste on utilise la propriété BuddyStatus de l'objet User. Par exemple pour effacer "l'ami" sélectionné dans la list box remplie au paragraphe précédent :

Effacer un contact en VB6

```

Private Sub cdSupprime_Click()
    If lstUser.ListIndex >= 0 Then
        i = lstUser.ListIndex + 1
        oSkype.Friends(i).BuddyStatus = budDeletedFriend
        MajListe
    End If
End Sub
    
```

ou en VBS ...

Effacer un contact en VBScript

```

Dim stUserAEffacer
Const budDeletedFriend = 1
stUserAEffacer = "toto"
Set oSkype = WScript.CreateObject("Skype4COM.Skype", "Skype_")
For Each oFriend in oSkype.friends
    if oFriend.Handle = stUserAEffacer Then
        oFriend.BuddyStatus = budDeletedFriend
        MsgBox stUserAEffacer & " à été supprimé!"
        Wscript.Quit
    end if
next
MsgBox stUserAEffacer & " Introuvable !"
    
```

IV-C - Recherche d'un nouvel ami

A la manière de la fenêtre "ajouter un contact de skype", la méthode SearchForUsers permet d'effectuer une recherche de contact. Utilisez par exemple le code suivant (après avoir rajouté une listbox et un textbox)

Recherche d'un contact en VB6

```

Private Sub Form_Load()
    Dim oSkype As New Skype
    Dim uContact As User
    If Not oSkype.Client.IsRunning Then
        oSkype.Client.Start
    End If
    
```

Recherche d'un contact en VB6

```

While Not oSkype.Client.IsRunning
DoEvents 'Attente démarrage client Skype
Wend
lstContacts.Clear
If txtFiltreRecherche <> "" Then

    Set ucContacts = oSkype.SearchForUsers(txtFiltreRecherche)
    For Each uContact In ucContacts
        lstContacts.AddItem uContact.Handle & " " & uContact.Country & " " & uContact.FullName
    Next
End If
end sub
    
```

la même recherche en VBS :

Recherche d'un contact en VBScript

```

Option Explicit
Dim oSkype ' Skype Application
Dim oUser ' User
Dim ucContacts' Collection d'User
Dim stUserAChercher ' string
stUserAChercher = "toto"
Set oSkype = WScript.CreateObject("Skype4COM.Skype", "Skype_")
Set ucContacts = oSkype.SearchForUsers(stUserAChercher)
For Each oUser in ucContacts
    Wscript.Echo oUser.Handle & " " & oUser.Country & " " & oUser.FullName
next
    
```

IV-D - Ajout d'un ami à la liste de contacts

Une fois la recherche effectuée, l'ajout à notre liste d'amis s'effectue en modifiant la propriété BuddyStatus de l'user et en la passant à budPendingAuthorization (2).

En VB6, on peut ajouter le code suivant par exemple, pour effectuer un ajout lors d'un double-clic sur la liste résultat de la recherche :

Ajout d'un contact en VB6

```

Private Sub lstContacts_DblClick()
Dim uContact As User
If lstContacts.ListIndex >= 0 Then
    Set uContact = ucContacts(lstContacts.ListIndex + 1)
    If Not uContact.Is Nothing Then
        If MsgBox("Ajout de : " & uContact.Handle, vbYesNo) = vbYes Then
            uContact.BuddyStatus = budPendingAuthorization
            FrmUsers.Show
            FrmUsers.MajListe
            Unload Me
        End If
    End If
End If
End Sub
    
```

En VBScript on peut faire la même chose en ajoutant le code suivant au précédent code de recherche:

Ajout d'un contact en VBScript

```

Const budPendingAuthorization = 2
if ucContacts.count = 1 then
    ucContacts(1).BuddyStatus = budPendingAuthorization
    MsgBox "Ajout de " & ucContacts(1).Handle
end if
    
```

V - Evènements

V-A - Documentation

Je ne vais pas vous lister ici tous les évènements de l'objet Skype, mais seulement 1 ou 2, histoire de vous montrer comment les utiliser.

Pour le reste, l'on retrouve sur le site développeur de Skype tout un chapitre sur les évènements de l'objet Skype:

 <https://developer.skype.com/Docs/Skype4COMLib/ISkypeEvents>

V-B - MessageStatus

L'évènement **HRESULT MessageStatus([in] IChatMessage *pMessage,[in] TChatMessageStatus Status)** est provoqué lors d'un changement dans le statut des messages de "chat", la fonction prend en paramètre un objet "Chatmessage" qui va nous permettre d'interagir sur le "Chat..", ainsi que la variable status de type TChatMessageStatus

Les diverses valeurs du type TChatMessageStatus

- -1 : Inconnu
- 0 : Envoi en cours...
- 1 : Envoyé
- 2 : Reçu
- 3 : Lu

Une application pourrait être le détournement des conversations "CHAT" pour lancer une commande à distance sur le pc accueillant l'application :

L'évènement MessageStatus en VB6

```

'Défini variable objetSkype avec "évènements"
Public WithEvents oSkype As Skype
Const MON_AMI = "toto" 'L'ami autorisé à lancer des commandes à distances
'
' Création de l'objet Skype
'
Private Sub Form_Load()
    Set oSkype = New Skype
    "Attache" les évènements à l'objet
    oSkype.Attach
End Sub
'
' "Surveillance" des messages de chat
'
Private Sub oSkype_MessageStatus(ByVal pMessage As SKYPE4COMLib.IChatMessage, ByVal Status As SKYPE4COMLib.TChatMessageStatus)
    Dim st As String
    Dim stMess As String
    If pMessage.FromHandle = MON_AMI And Status = cmsReceived Then
        Select Case pMessage.Body
            Case "DIR" ' Si commande DIR reçue renvoi la liste des fichiers..sous C:\
                'Creation de la liste de fichiers
                st = Dir("c:\*.*)"
                While st <> ""
                    stMess = stMess & st & vbCrLf
                    st = Dir
                Wend
                'Envoi de la liste de fichiers
                pMessage.Chat.SendMessage stMess
            Case Else
                List1.AddItem ">>Mess>>" & pMessage.Body
        End Select
    End If
End Sub

```

L'évènement MessageStatus en VB6

La même chose en VBScript :

L'évènement MessageStatus en VBScript

```

Dim oSkype 'As Skype 'L'objet public Sype
const MON_AMI = "toto"
const cmsReceived = 2
init_Skype

Do While True ' boucle sans fin d'attente évènements
    WScript.Sleep(60000)
Loop

'
' "Surveillance" des messages de chat
'
Public Sub oSkype_MessageStatus(ByVal pMessage , ByVal Status )
Dim st' As String
Dim stMess' As String
Dim oFso 'As FileSystemObject

If pMessage.FromHandle = MON_AMI And Status = cmsReceived Then
    Select Case pMessage.Body
        Case "DIR" ' Si commande DIR reçue renvoi la liste des fichiers..sous C:\
            'Creation de la liste de fichiers
            Set oFSO = CreateObject("Scripting.FileSystemObject")
            For each oFl in oFSO.GetFolder("c:\").Files
                stMess = stMess & oFl.name & vbCrLf
            Next
            'Envoi de la liste de fichiers
            pMessage.Chat.SendMessage stMess
        Case Else
            WScript.Echo ">>Mess>>" & pMessage.Body
        End Select
    End If
End Sub

'-----'
' Init_Skype :
' Fonction d'initialisation Skype permet de lancer le client skype
' si celui-ci ne l'est pas et effectue la "connection" du projet à l'API Skype.
'-----'
Sub init_Skype()
    Set oSkype = WScript.CreateObject("Skype4COM.Skype", "oSkype_")
    If Not oSkype.Client.IsRunning Then
        oSkype.Client.Start
    End If
    While Not oSkype.Client.IsRunning
        DoEvents 'Attente démarrage client Skype
    Wend
    oSkype.Attach 'Si l'on doit gérer des évènements
End Sub
    
```

On remarquera en VBScript le paramètre "*location*", ici **oSkype_**, qui permet de "lier" l'objet Skype aux fonctions événementielles que l'on désire lui associer.

V-C - CallStatus

L'évènement **HRESULT CallStatus([in] ICall *pCall,[in] TCallStatus Status)** est provoqué lors du changement d'état du status d'appel.

La fonction prend en paramètre un objet "call" qui va nous permettre d'interagir sur cet appel, ainsi que le statut de cet appel dans une variable de type TCallStatus

Quelques-unes des valeurs du CallStatus

- 0 : **UNPLACED** = "Jamais placé"
- 1 : **ROUTING** = "Routage"
- 2 : **EARLYMEDIA** = "Lecture flux média (Early Media)"
- 3 : **FAILED** = "Désolé, l'appel a échoué !"
- 4 : **RINGING** = "un appel"
- 5 : **INPROGRESS** = "Appel en cours..."
- 6 : **ONHOLD** = "En attente"
- 7 : **FINISHED** = "Terminé"
- 8 : **MISSED** = "Appel en absence"
- 9 : **REFUSED** = "Refusé"
- 10 : **BUSY** = "Occupé"
- 12 : **LOCALHOLD** = "En attente locale"
- 13 : **REMOTEHOLD** = "En attente à distance"

Un petit exemple d'utilisation, avec une réponse automatique sur réception d'un appel (télésurveillance bébé par exemple..)

L'évènement CallStatus en VB6

```
'Défini variable objetSkype avec "événements"
Public WithEvents oSkype As Skype
'
' Création de l'objet Skype
'
Private Sub Form_Load()
    Set oSkype = New Skype
    "Attache" les événements à l'objet
    oSkype.Attach
End Sub
' Détection d'un appel
'
Private Sub oSkype_CallStatus(ByVal pCall As SKYPE4COMLib.ICall, ByVal Status As
SKYPE4COMLib.TCallStatus)
    List1.AddItem "CallStatus : " & Status & " => " & oSkype.Convert.CallStatusToText(Status)
    'Détection d'un appel
    If Status = clsRinging Then
        List1.AddItem "Type : " & oSkype.Convert.CallTypeToText(pCall.Type)
        If pCall.Type = cltIncomingP2P Or pCall.Type = cltIncomingPSTN Then
            List1.AddItem "Réponse automatique à : " & pCall.PartnerHandle
            pCall.Answer
        End If
    End If
End Sub
```

La même chose en VBScript :

L'évènement CallStatus en VBScript

```
Option Explicit
Dim oSkype 'As Skype 'L'objet public Syype
Const clsRinging = 4
Const cltIncomingP2P = 2
Const cltIncomingPSTN = 0
init_Skype

Do While True ' boucle sans fin d'attente évènements
    WScript.Sleep(60000)
Loop

'
' Détection d'un appel
'
Public Sub oSkype_CallStatus(ByVal pCall , ByVal Status )
    Wscript.Echo "CallStatus : " & Status & " => " & oSkype.Convert.CallStatusToText(Status)
```

L'évènement CallStatus en VBScript


```

'Détection d'un appel
If Status = clsRinging Then
    Wscript.Echo "Type : " & oSkype.Convert.CallTypeToText(pCall.Type)
    If pCall.Type = cltIncomingP2P Or pCall.Type = cltIncomingPSTN Then
        Wscript.Echo "Réponse automatique à : " & pCall.PartnerHandle
        pCall.Answer
    End If
End If
End Sub

'-----'
' Init_Skype :
' Fonction d'initialisation Skype permet de lancer le client skype
' si celui-ci ne l'est pas et effectue la "connexion" du projet à l'API Skype.
'-----'
Sub init_Skype()
    Set oSkype = WScript.CreateObject("Skype4COM.Skype", "oSkype_")
    If Not oSkype.Client.IsRunning Then
        oSkype.Client.Start
    End If
    While Not oSkype.Client.IsRunning
        DoEvents 'Attente démarrage client Skype
    Wend
    oSkype.Attach 'Si l'on doit gérer des évènements
End Sub
End Sub

```

VI - Envoi de commande "directe"

Pour les "commandes" api qui n'auraient pas été implémentées dans l'interface SkypeCom vous pouvez utiliser la méthode "SendCommand" qui permet l'utilisation de toutes les commandes de l'API Skype, voir ici :  <https://developer.skype.com/Docs/ApiDoc/Commands>

VI-A - Envoi de commande en VB6

L'envoi de commande passe par la création d'un objet de type command.

```
Public oCde As SKYPE4COMLib.Command
```

Il convient ensuite de renseigner les propriétés de cet objet

- **Id** (long) : défini par l'utilisateur. Définit l'identificateur de la commande (il sera "renvoyé" lors du retour du résultat de cette commande.)
- **Timeout** (long) : Le "time-out" de la commande en milliseconde réglé par défaut à 30 000 (30 s)
- **Blocking** : à True (vrai), lors de l'envoi de commande le code est "bloqué" dans l'attente du retour; à False (faux), le code n'est pas bloqué et se déroule en parallèle
- **Command** (string) : La commande proprement dit
- **Reply** (string) : Le texte reçu en retour.
- **Expected** (string) : Le texte attendu en retour.

Pour une commande en mode "bloquant", un ping, par exemple :

```

oCde.Blocking = true
oCde.Id = 123 ' Numéro identificateur de la commande
oCde.Command = "PING"
oSkype.SendCommand oCde
Msgbox oCde.Reply 'Affichage du retour commande

```

on utilise la propriété Reply, comme retour de la commande immédiatement après sont l'instruction .SendCommand dans le cas de la même commande ("ou" d'une autre) lancée en mode non bloquant :

```
oCde.Blocking = false
oCde.Id = 123 ' Numéro identificateur de la commande
oCde.Command = "PING"
oSkype.SendCommand oCde
```

On utilise l'évènement .reply de l'objet skype pour traiter le retour de la commande :

```
Private Sub oSkype_Reply(ByVal pCommand As SKYPE4COMLib.ICommand)
    If pCommand.Id = 123 Then 'on teste par exemple l'ID utilisé lors du lancement de la commande
        MsgBox pCommand.Reply
    End If
End Sub
```

VI-B - Envoi de commande en VBS

En Vbs on utilise la fonction "command" de l'objet skype pour créer l'objet de type command, et l'utiliser avec le send command

oSkype.[Command](#)(Id As Long, Command As String, [Reply As String], [Block As Boolean = False], [Timeout As Long = 3000])

L'on retrouve en paramètre les propriétés de l'objet command.

- **Id** (long) : L'identifiant de la commande
- **Command** (String) : la commande à utiliser sous forme de chaîne.
- **Reply** (String) : facultatif, la chaîne reçue par défaut.
- **Block** (Boolean = Faux) : facultatif, le mode de fonctionnement de la commande (bloquant/non-bloquant)
- **Timeout** (Long = 3000) : Temps avant passage en time-out de la commande, en millisecondes (30s par défaut)

Pour ajouter un ami à notre liste de contact au [chapitre IV-D](#), nous avons modifié la propriété BuddyStatus. Cependant la commande API d'ajout d'un USER dans la liste de contacts permet de définir le message de bienvenue envoyé au contact :

SET USER <NomDucontact> BUDDYSTATUS 2 <Message à envoyer>

On peut écrire cela en VBScript sous la forme:

```
Set oSkype = WScript.CreateObject("Skype4COM.Skype", "")
'Lance le client skype
If Not oSkype.Client.IsRunning Then oSkype.Client.Start() End If
'Connection à l'API
oSkype.Attach
set oCde =
    oSkype.Command(123,"SET USER toto.test3 BUDDYSTATUS 2 S.V.P accepte moi dans ta liste d'amis","Retour par défaut")
oSkype.SendCommand(oCde)
Wscript.echo "Retour : " & oCde.Reply
```

VII - Téléchargement

[Source](#) Exemple de code gestion des contacts Skype en VB6

[Sources](#) Exemple de code gestion des contacts Skype en VBScript

[Source](#) Exemple de code gestion des évènements de l'objet Skype en VB6

[Sources](#) Exemple de code gestion des évènements de l'objet Skype en VBScript

VIII - Conclusion

VIII-A - "à vous de jouer"

Pour finir j'espère que cette introduction à la programmation autour de Skype vous donnera des idées pour développer une "vraie" application. Vous pourrez alors utiliser **le forum VB** pour nous en donner la primeure.

VIII-B - Remerciements

Tous mes remerciements à **Pierre Fauconnier** et **ThierryAIM** pour leurs relectures.