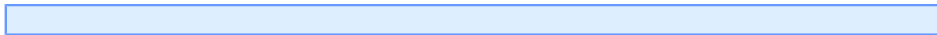


# Tutoriel 23 : Le contrôle documenté "MCI32.ocx"

par Gilbert Miralles ([gilmir.developpez.com](http://gilmir.developpez.com))

Date de publication : Lundi 18 mars 2003

Dernière mise à jour : Lundi 4 février 2008



## I - Présentation

Multimedia MCI, contrôle, constantes

Prochain tutoriel :

## I - Présentation

### MULTIMEDIA MCI32 [MCI32.OCX]

**Utilisation :** Affichage et manipulation de fichiers multimédias

**EDITEUR :** Microsoft

Le contrôle MCI a été fourni avec toutes les versions de Visual Basic depuis la version 3.0

Ce contrôle sert à gérer différents périphériques multimédia comme les cartes son, les séquenceurs MIDI et les lecteurs de CD audio et de CD-ROM. Il peut aussi diffuser des fichiers au formats audio Wave(.WAV), MIDI(.MID) et vidéo(.AVI)



Le contrôle MCI ressemble à un clavier de commande de magnéto-scope.

Pour que le contrôle MCI fonctionne parfaitement, les services appropriés de l'interface de Contrôle Multimédia (Services MCI) doivent être disponibles et être capable d'accéder au périphérique son, grâce au pilote spécifique installé sous Windows.

Extrait du code pour charger un fichier Audio

```

Private Sub Form_Load ()
    ' Définit les propriétés nécessaires à l'ouverture du périphérique MCI.
    MMControll.Notify = FALSE
    MMControll.Wait = TRUE
    MMControll.Shareable = FALSE
    MMControll.DeviceType = "WaveAudio"
    MMControll.FileName = "C:\WINDOWS\MMDATA\GONG.WAV"

    ' Ouvre le périphérique MCI WaveAudio.
    MMControll.Command = "Open"
End Sub
    
```

Les propriétés et événements spécifiques au contrôle MCI sont présentées dans les tableaux suivants

### Multimedia MCI, contrôle, constantes

Constantes du mode		
Constante	Valeur	Description
mciModeNotOpen	524	Périphérique non ouvert.
mciModeStop	525	Arrêt du périphérique.
mciModePlay	526	Lecture du périphérique.
mciModeRecord	527	Enregistrement du périphérique.
mciModeSeek	528	Recherche du périphérique.
mciModePause	529	Pause.
mciModeReady	530	Périphérique prêt.

Constantes du mode		
Constante	Valeur	Description
mciNotifySuccessful	1	La commande a abouti.
mciNotifySuperseded	2	La commande a été annulée par une autre commande.
mciAborted	4	La commande a été annulée par l'utilisateur.
mciFailure	8	La commande a échoué.

Constantes d'orientation		
Constante	Valeur	Description
mciOrientHorz	0	Boutons disposés horizontalement.
mciOrientVert	1	Boutons disposés verticalement.

Constantes RecordMode		
Constante	Valeur	Description
mciRecordInsert	0	Insérer le mode d'enregistrement.
mciRecordOverwrite	1	Remplacer le mode d'enregistrement.

Constantes Format		
Constante	Valeur	Description
mciFormatMilliseconds	0	Format millisecondes.
mciFormatHms	1	Format heures, secondes et minutes.
mciFormatMsf	2	Format minutes, secondes et images.
mciFormatFrames	3	Format images.
mciFormatSmppte24	4	Format SMPTE 24 images.
mciFormatSmppte25	5	Format SMPTE 25 images.
mciFormatSmppte30	6	Format SMPTE 30 images.
mciFormatSmppte30Drop	7	Format SMPTE 30 images cibles.
mciFormatBytes	8	Format octets.
mciFormatSamples	9	Format échantillons.
mciFormatTmsf	10	Suit le format minutes, secondes et images.

Lorsque j'ai décidé de construire ce Tutoriel autour du contrôle MCI, plusieurs choix s'offraient à moi, je pouvais avec toute la documentation en ma possession trouver les instructions afférentes à une application multimédia.

Mais...partant du principe qu'il ne faut pas refaire ce qui est déjà fait, je me suis posé la question à savoir si je ne pouvais pas trouver dans mes fonds de tiroir une application déjà toute faite.

J'ai trouvé en effet dans un programme écrit sous VB3.0 une application fonctionnant avec un contrôle MCI issu d'un fichier à extension .vbx!

Pour ceux qui suivent mes Tutoriels, j'ai évoqué précédemment une méthode pour convertir les applications fonctionnant sous 16 bits en applications 32 bits, j'ai donc pensé qu'il fallait essayer de mettre à profit les précieux conseils que je divulgue sans modération.

J'ai donc utilisé la méthode que je vous ai déjà indiqué et je n'ai trouvé qu'un seul problème au niveau d'un appel d' "API", je vous donne la modification que j'ai apportée afin de faire fonctionner ma feuille écrite sous VB3(16bits) en instructions pouvant être lues sous VB6.0.



Une vue de la démo du contrôle MCI

que **Vous pouvez télécharger ici!**

La zone d'image est tronquée sur ce cliché, mais elle existe et se trouve cachée par la bordure inférieure.

On peut ne pas tenir compte de la zone d'image si on se contente d'utiliser la partie son du contrôle MCI

Nous écrivons dans "Général" et "Déclarations" de la feuille "Form1"

```

'BEGIN
Option Explicit
'-----
' Variables, constantes et déclarations utilisées dans la
' démo du contrôle Multimédia MCI
'-----

' Stocke le Handle de fenêtre du contrôle d'image
' utilisé pour afficher les fichiers AVI      (si vous voulez afficher des images)
Dim HwndFrame As Integer

' Récupère le Handle de fenêtre du contrôle d'image
'Private Declare Function GetFocus Lib "user" ( ) As Integer           'Déclaration d'origine
    
```

Ndla - Nous aurions pu ne pas tenir compte de cette déclaration si nous nous servons uniquement que de la partie son de cette application Et voici le code que j'ai récupéré dans le fichier des API de Windows qui a une relation avec la fonction GetFocus qui se trouve dans le fichier DLL intitulé "User32"

```

'Declare Function GetFocus Lib "user32" ( ) As Long
    
```

Je l'ai modifié en :

```

Private Declare Function GetFocus Lib "user32" ( ) As Long

' Mémoire la largeur et la hauteur initiale de la fenêtre.
'La feuille ne peut être redimensionnée en dessous de ces valeurs.
Dim MinWinWidth As Integer
Dim MinWinHeight As Integer

' Color constants
Const BLUE = &H800000
    
```

```
Const RED = &H80&
```

Dans cette procédure j'ai modifié uniquement le code relatif à l'affichage des fichiers AVI Et j'ai traduit les commentaires en Français(Sur le module que vous téléchargerez, les commentaires seront en Anglais, désolé!)

```
Private Sub GetNewFileName(AFile As String)
'-----
' Utilise la boîte de dialogue standard du menu fichier pour chercher un nom de fichier
'-----
Dim Canceled As Integer

Canceled = False
On Error GoTo GetNewFileName_Error

' Configure la boîte de dialogue fichier standard
CMDialog1.Filter = "Multimedia Files|*.wav;*.mid"           '*.*.avi Attention coder les
extensions en minuscule
CMDialog1.DialogTitle = "Select Multimedia File"
CMDialog1.CancelError = True

' Ouvre la boîte de dialogue
CMDialog1.Action = 1

If Not Canceled Then
    AFile = CMDialog1.FileName
Else
    AFile = ""
End If
Exit Sub
'
GetNewFileName_Error:
Canceled = True
Resume Next
End Sub
```

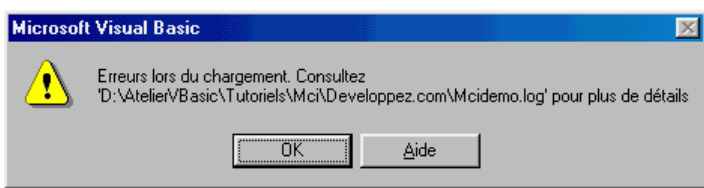
Les autres procédures sont identiques et ne demandent pas de modifications

Vous avez besoin du contrôle "MCI32.ocx" et du contrôle "Comdlg32.ocx" pour faire fonctionner cette application, ce dernier servant au fonctionnement des boîtes de dialogues sous Windows.

Pour que le module de démonstration fonctionne, vous devez créer un répertoire "Sons" à la racine de votre disque dur"C:" et insérer les fichiers sons qui figurent dans le fichier ZIP compressé qui accompagnent cet exposé.

Pour des raisons que j'ignore, j'ai pu constater que certains fichiers Wav ne pouvaient pas être lus par le contrôle MCI que ce soit sous plate forme 16 bits ou indifféremment sous 32 bits.

J'attire votre attention sur le fait que lorsque vous allez lancer l'application "MciDemo" que vous allez télécharger, celle-ci va engendrer une erreur du genre :



Fenêtre d'erreur générée par

Visual Basic6

**Que nous dit ce fichier ?(mcidemo.log)**

*Ligne 18: La classe `CommonDialog` du contrôle `CMDialog1` n'est pas une classe de contrôle chargée.*

*Ligne 31: La classe `MMControl` du contrôle `MMControl1` n'est pas une classe de contrôle chargée.*

### Qu'est ce que cela veut bien dire ?

Cela veut dire que cette feuille ayant été écrite avec une version précédente de VB et faisant appel à des DLL incompatibles (\*.vbx) qui ne peuvent pas être lues par VB6.0, le système nous prévient d'une incompatibilité majeure et qu'il a été dans l'impossibilité de charger les contrôles sollicités par les appels aux DLL.

Charge à nous d'apporter les modifications nécessaires pour charger les contrôles adéquats qui rétabliront le bon fonctionnement de ce contrôle Multimédia.

Je vous laisse le soin de modifier en conséquence les instructions inopportunes ce qui à mon avis ne devrait pas vous poser de problème avec les explications que je vous ai données.

Je vous souhaite beaucoup de plaisir à réaliser cette application.

Prochain tutoriel :

 ***Les bases de la gestion de fichiers***

