

Créer un lecteur de CD audio dans Excel

par SilkyRoad (silkyroad.developpez.com)

Date de publication : 07/10/2006

Dernière mise à jour : 21/04/2007

Ce document montre comment créer et utiliser un lecteur de CD Audio dans Excel.

Les procédures sont basées sur l'interface multimédia MCI. Vous verrez aussi comment utiliser une macro complémentaire (xla) et comment ajouter un menu personnalisé dans Excel.

Les procédures ont été adaptées d'un code VB6 anonyme et testées avec Excel2002 et WinXP.

- I - La description du lecteur
- II - Les procédures
- III - Les macros complémentaires xla
- IV - Configurations de test
- V - Téléchargement

I - La description du lecteur

Ce chapitre décrit les éléments constitutifs du lecteur. La démo complète est téléchargeable au format xla à la fin de ce document.


Un menu personnalisé nommé "XL Music Player" va être ajouté dans Excel lors de l'ouverture du classeur.



Ce menu permet d'afficher le lecteur de CD.

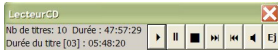
L>UserForm va se placer automatiquement en bas et à droite de l'écran.

Sa taille est volontairement compacte et de type **non modal** pour continuer à travailler dans votre tableur préféré tout en écoutant de la musique...;o)

 *Le CD audio doit être inséré dans le lecteur avant d'afficher l'UserForm.*

Les options de la boîte de dialogue:

- *La lecture
- *La mise en pause
- *L'arrêt du lecteur
- *Le passage à la séquence musicale suivante
- *Le passage à la séquence musicale précédente
- *La modification du volume sonore
- *L'éjection du CD
- *L'affichage du nombre de titres contenu dans le CD
- *L'affichage de la durée totale du CD
- *L'affichage de la durée de la séquence en cours
- *L'affichage de l'index de la séquence en cours



Remarque:

Les symboles affichés sur certains boutons sont obtenus en appliquant la police Webdings.

II - Les procédures

Les premières procédures doivent être placées dans le module objet **ThisWorkbook**:

Elles permettent de créer le menu personnalisé lors de l'ouverture du classeur puis de le supprimer quand vous refermez le classeur.

```
Vba

Option Explicit

'Permet d'ajouter un menu personnalisé dans Excel, lors de l'ouverture
'du classeur.
Private Sub Workbook_Open()
    Dim Nouveau As CommandBarControl

    On Error Resume Next
    Set Nouveau = Application.CommandBars(1).Controls.Add(msoControlPopup, , , , True)

    With Nouveau
        .Caption = "XL Music Player"
        'Attribue une macro à la barre de menu.
        .OnAction = "Lancer"
    End With

End Sub

'Evenement fermeture classeur.
Private Sub Workbook_BeforeClose(Cancel As Boolean)
    On Error Resume Next

    'Ferme la boîte de dialogue s'il elle est affichée au moment de la
    'fermeture du classeur.
    Unload LecteurCD
    'Supprime le menu personnalisé lors de la fermeture du classeur
    Application.CommandBars(1).Controls("XL Music Player").Delete

End Sub
```

Ensuite, les fonctions et macros ci dessous sont insérées dans un **module standard**:

```
Vba

Option Explicit

Public PresenceCD As Boolean
Public CDenCours As Boolean

'Permet de trouver le Handle du UserForm.
Public Declare Function FindWindow Lib "user32" Alias "FindWindowA" _
    (ByVal lpClassName As String, ByVal lpWindowName As String) As Long

'Permet d'envoyer une commande à l'interface multimédia MCI.
Declare Function mciSendString Lib "winmm.dll" Alias "mciSendStringA" _
    (ByVal lpstrCommand As String, ByVal lpstrReturnString As String, ByVal uReturnLength As Long, _
    ByVal hwndCallback As Long) As Long

'Permet de retrouver la description des erreurs lors de l'utilisation de MCI.
Declare Function mciGetErrorString Lib "winmm.dll" Alias "mciGetErrorStringA" _
```

Vba

```
(ByVal dwError As Long, ByVal lpstrBuffer As String, ByVal uLength As Long) As Long

'La macro "Lancer" est liée au bouton du menu personnalisé "XL Music Player"
Sub Lancer()
    On Error GoTo Fin

    CDenCours = False
    PresenceCD = True

    'Affiche l'UserForm (non modal)
    LecteurCD.Show 0
    'Lance la procédure pour vérifier s'il y a un CD audio dans le lecteur
    If PresenceCD = False Or CDenCours = True Then Unload LecteurCD

Fin:
    If Err.Number = 91 Then Exit Sub
End Sub
```

Les procédures suivantes sont placées dans un **UserForm** nommé "LecteurCD". La boîte de dialogue contient:

2 labels nommés Label1 et Label2.

7 CommandButton nommés CommandButton1 à CommandButton7.

Vba

```
Option Explicit
'Source: http://www.vbcode.com/asp/showzip.asp?ZipFile=playcd.zip&theID=42
'

Dim Valeur As Long
Dim strFormClassName As String
Dim fPlaying As Boolean, fCDLoaded As Boolean, CtrlEject As Boolean
Dim NbTitres As Integer
Dim DureeTitres() As String
Dim Cmd As String
Dim Mini As Integer, Sec As Integer, Track As Integer

Private Sub UserForm_Initialize()

    fCDLoaded = False
    CtrlEject = False

    ' Quitte si le CD est déjà utilisé
    If SendMCIStrng("open cdaudio alias cd wait shareable", True) = False Then _
        CDenCours = True

    Static s As String * 30
    ' Contrôle si le CD est dans le lecteur
    mciSendString "status cd media present", s, Len(s), 0
    If CBool(s) = False Then PresenceCD = False

    SendMCIStrng "set cd time format tmsf wait", True
    Update

End Sub
```

Vba

```

Private Sub UserForm_Activate()
    With Me 'affiche l'USF en bas et à droite de l'écran
        .Top = Application.Height - Me.Height
        .Left = Application.Width - Me.Width
    End With
End Sub

Private Function SendMCIStrng(Cmd As String, fShowError As Boolean) As Boolean
    Static Rc As Long
    Static errStr As String * 200

    'Récupère le Handle de l'Usf: Daniel Klann, mpep
    If Val(Application.Version) < 9 Then 'Excel 2000
        strFormClassName = "ThunderXFrame"
    Else
        strFormClassName = "ThunderDFrame" 'Excel 2000/2002
    End If
    Valeur = FindWindow(strFormClassName, "Lecteur CD") 'Récupère le Handle du UserForm

    '*****
    If CtrlEject = True Then Exit Function

    Rc = mciSendString(Cmd, 0, 0, Valeur)

    If fShowError And Rc <> 0 And Len(errStr) <> 200 Then
        mciGetErrorString Rc, errStr, Len(errStr)
        MsgBox errStr
    End If

    On Error GoTo 0
    SendMCIStrng = (Rc = 0)
End Function

Private Sub CommandButton1_Click()
    ' Lecture
    SendMCIStrng "play cd", True
    fPlaying = True
End Sub

Private Sub CommandButton2_Click()
    ' Revenir à la séquence précédente
    Dim from As String
    If Mini = 0 And Sec = 0 Then
        If Track > 1 Then
            from = CStr(Track - 1)
        Else
            from = CStr(NbTitres)
        End If
    Else
        from = CStr(Track)
    End If

    If fPlaying Then
        Cmd = "play cd from " & from
    End If
End Sub

```

Vba

```
SendMCIStrng Cmd, True
Else
    Cmd = "seek cd to " & from
    SendMCIStrng Cmd, True
End If

Update

End Sub

Private Sub CommandButton3_Click()
    ' Passer à la séquence suivante
    If Track < NbTitres Then
        If fPlaying Then
            Cmd = "play cd from " & Track + 1
            SendMCIStrng Cmd, True
        Else
            Cmd = "seek cd to " & Track + 1
            SendMCIStrng Cmd, True
        End If
    Else
        SendMCIStrng "seek cd to 1", True
    End If

    Update
End Sub

Private Sub CommandButton4_Click()
    ' Pause
    SendMCIStrng "pause cd", True
    fPlaying = False

    Update
End Sub

Private Sub CommandButton5_Click()
    ' Stop
    SendMCIStrng "stop cd wait", True
    Cmd = "seek cd to " & Track
    SendMCIStrng Cmd, True
    fPlaying = False

    Update
End Sub

Private Sub CommandButton6_Click()
    fPlaying = False

    'éjecte le CD
    SendMCIStrng "set cd door open", True
    CtrlEject = True
    Unload Me
End Sub
```

Vba

```

Private Sub CommandButton7_Click()
    'règlage son
    Dim R As Long
    R = Shell("sndvol32 /t", vbNormalFocus)
End Sub

Private Sub Update()
    Dim j As Byte
    Dim i As Integer
    Dim NumPlage As String
    Static s As String * 30

    j = 0
    ' Controle si le CD est dans le lecteur
    mciSendString "status cd media present", s, Len(s), 0
    If CBool(s) = True Then

        If fCDLoaded = False Then
            mciSendString "status cd number of tracks wait", s, Len(s), 0
            NbTitres = CInt(Mid$(s, 1, 2))
            CommandButton6.Enabled = True

            If NbTitres = 1 Then Exit Sub

            mciSendString "status cd length wait", s, Len(s), 0
            Label1 = "Nb de titres: " & NbTitres & " Durée : " & s
            ReDim DureeTitres(1 To NbTitres)

            'Récupère dans un tableau la durée de chaque titre du CD
            For i = 1 To NbTitres
                Cmd = "status cd length track " & i
                mciSendString Cmd, s, Len(s), 0
                DureeTitres(i) = s
            Next

            For j = 1 To 6
                Controls("CommandButton" & j).Enabled = True
            Next j

            fCDLoaded = True
            SendMCIString "seek cd to 1", True
        End If

        mciSendString "status cd position", s, Len(s), 0
        Track = CInt(Mid$(s, 1, 2))
        Mini = CInt(Mid$(s, 4, 2))
        Sec = CInt(Mid$(s, 7, 2))

        '*****
        If Track = 0 Then
            MsgBox "impossible de lire ce CD"
            Unload Me
            Exit Sub
        End If

        '*****

        NumPlage = "[" & Format(Track, "00") & "]"
        Label2 = "Durée du titre " & NumPlage & " : " & DureeTitres(Track)

        ' vérifie si le CD est en cours de lecture
    
```

Vba

```
mciSendString "status cd mode", s, Len(s), 0
fPlaying = (Mid$(s, 1, 7) = "playing")
Else
  CommandButton6.Enabled = False

  If fCDLoaded = True Then

    For j = 1 To 6
      Controls("CommandButton" & j).Enabled = False
    Next j
    fCDLoaded = False
    fPlaying = False
    Label2 = ""
  End If

End If
End Sub

Private Sub UserForm_Terminate()
  Cmd = "seek cd to " & Track
  SendMCIStrng Cmd, True
  fPlaying = False
  Update

  SendMCIStrng "close all", False
End Sub
```

III - Les macros complémentaires xla

Lorsque votre classeur est créé, vous avez la possibilité de l'enregistrer au format xla (macro complémentaire).

Une macro complémentaire est un fichier Excel, dont les feuilles ne sont pas visibles. Ce type de fichier est principalement conçu pour stocker des macros. Il est possible d'activer une macro complémentaire afin qu'elle se charge automatiquement dès le démarrage d'Excel. Vous disposez ainsi en permanence des macros contenues dans le fichier.

Pour activer une macro complémentaire:

Utilisez le menu Outils

Macros Complémentaires

Cliquez sur le bouton Parcourir

Sélectionnez le fichier (Par exemple "XL_Music_Player_V01.xla" téléchargeable en fin de tutoriel) , puis cliquez sur le bouton OK pour valider.

Le classeur est désormais disponible dans la liste des Macros Complémentaires.

Cochez "XL_Music_Player_V01" et cliquez sur le bouton OK pour valider.

Si vous stockez le fichier xla directement dans le répertoire des Macros Complémentaires par défaut, généralement:

<C:\Documents and Settings\NomUtilisateur\Application Data\Microsoft\Macro complémentaires>

vous n'avez pas besoin d'utiliser le bouton "Parcourir". Le fichier apparaîtra automatiquement dans la liste des macros complémentaires.

IV - Configurations de test

Ce chapitre récapitule les versions du système d'exploitation et de l'application Excel utilisées pour les tests:

* WinXP SP1 - Excel2002 : OK

* WinXP SP2 - Excel2002 : OK

* Win XP SP2 - Excel 2003 : OK

* Win2000 SP1 - Excel2000 : Pas OK

* Win2k SP4 - Excel 2003 : OK

* Win XP SP2 - Excel 2007 : OK

V - Téléchargement

Téléchargez le classeur démo, au format xla.

