1 MCIWndX OLE Control

1.1 What is MCIWndX ?

MCIWndX is a 32-bit OLE control wrapper around Windows' MCIWnd multimedia window class. The purpose of this class is to allow applications to easily add multimedia functionality to their applications. The MCIWndX control brings this same functionality with even greater ease to any development tool which can host OLE controls, including Microsoft Visual Basic 4.0, Microsoft Visual FoxPro 3.0, Microsoft Access 7.0 and Microsoft Visual C++ 4.0.

The MCIWndX control is derived from and completely compatible with the original MCIWndX VBX provided as part of the Video for Windows 1.1 Developers Kit. However, it has been extended to provide additional functionality and features, making it even more powerful and easy to use.

1.2 What's new in MCIWndX ?

Several new features have been added to MCIWndX:

1. An *EditCopy* method has been added. Calling this function causes the current frame to be copied to the clipboard (if the active device has a concept of "frames").
2. A new *ExtendedError* event has been added. In the original VBX, the *Error* event passed in a string parameter. However, because the error strings were localized, writing code which checked what type of error had occurred was difficult. To maintain compatibility with the VBX, the original *Error* event has been retained, and a property (*WantExtendedError*) has been added to determine whether the *Error* or the *ExtendedError* event will fire.
3. A new *ExtendedModeChange* event has been added. In the original VBX, the *ModeChange* event passed in a string parameter indicating the new mode. However, because the mode strings were localized, writing code which checked what the new mode was was difficult. To maintain compatibility with the VBX, the original *ModeChange* event has been retained, and a property (*WantExtendedMode*) has been added to determine whether the *ModeChange* or the *ExtendedModeChange* event will fire.
4. Support has been added for the *DblClick* event, *MouseDown* and *MouseUp* events. A new property, *DefaultContextMenuEnabled* has been added to support this.
5. A new *Signal* event has been added to provide support for MCI's signalling abilities.
6. An hMCIWnd property has been added - this gives the hWnd of the MCIWnd at runtime, which can be used with the MCIWnd API's and macros.

1.3 How to use MCIWndX

Please refer to the MCIWndX documentation in the Video for Windows DK or the MCIWnd documentation in the Win32 SDK for complete information about how to use MCIWndX.

1.4 Complete reference

MCIWndX is simply a wrapper for the MCIWnd window class, which is part of Windows. Therefore, most functionality is described in good detail in the MCIWnd reference in the Win32 SDK documentation. The MCIWndX source code shows exactly how the API's are being used - refer to it when in doubt.

1.4.1 Properties

DefaultContextMenuEnabled

This property determines whether or not the default context menu is enabled or not. When this property is TRUE, your application will not receive any mouse button events (MouseUp, MouseDown, DblClick) for the right mouse button. TRUE by default.

hMCIWnd

At runtime, this property gives the hWnd of the MCIWnd. This can be used with the MCIWnd API's and macros (see the SDK documentation) to extend MCIWnd. The control covers most callback situations, so one should be able to do almost anything that can be accomplished using traditional techniques.

WantExtendedError

This property determines whether or not the ExtendedError event is fired instead of the Error event. FALSE by default.

WantExtendedMode

This property determines whether or not the ExtendedModeChange event is fired instead of the ModeChange event. FALSE by default.

1.4.2 Methods

EditCopy

Copies the current frame to the clipboard. For devices which don't support images (eg waveaudio) and appropriate bitmap is copied to the clipboard instead.

1.4.3 Events

Error

Parameters: ErrorCode As String

Occurs when an error is encountered while executing an MCI command. The ErrorCode parameter contains the message string corresponding to the error.

ExtendedError

Parameters: ErrorCode As Long

A new form of the Error event. Instead of the error being provided as a string, it is provided numerically, which makes coding and internationalization easier. This event only fires if WantExtendedError is TRUE.

ExtendedModeChange

A new form of the Mode event. Instead of the error being provided as a string, it is provided numerically, which makes coding and internationalization easier. This event only fires if WantExtendedError is TRUE.

DblClick

Fires when user double-clicks on the control. Does not fire when the user double-clicks the right mouse button and DefaultContextMenuEnabled is TRUE.

Notify

Parameters: NotifyCode as Long

Fires when the controls receives an MM_MCINOTIFY message (as a result of performing an operation with the notify flag). The NotifyCode parameter contains the code corresponding to the notify status.

Notification	**Meaning**
mciSuccessful	Successful
mciSuperseded	Superseded
mciAborted	Aborted
mciFailure	Failure

MouseDown

Fires when user pushes the mouse button while the cursor is on the control. Does not fire when the user pushes the right mouse button and DefaultContextMenuEnabled is TRUE.

MouseUp

Fires when user releases the mouse button while the cursor is on the control. Does not fire when the user releases the right mouse button and DefaultContextMenuEnabled is TRUE.

Signal

Fires when the control receives an MM_MCISIGNAL message (in reponse to a signal command)