
CHAPTER 1

MCI Command Messages for MCIavi

This chapter describes the MCI command messages for the Microsoft MCI video device driver (MCIavi.DRV) that you can use with the **mciSendCommand** function that supports the MCI command-message interface. For more information on using command messages, see the *Microsoft Multimedia Programmer's Guide* and the *Microsoft Multimedia Programmer's Reference* in the Microsoft Windows Software Development Kit. The same information is also available in the *Multimedia Programmer's Workbook* and *Multimedia Programmer's Reference* in the Microsoft Windows Multimedia Development Kit.

The following list summarizes the MCI command messages supported by MCIavi. This command set is taken from the digital-video command set for MCI. Any device-specific behavior affecting the MCI commands is also noted in this appendix.

MCI Command Messages

XE "MCI devices:command messages"\$XE "AVI video devices:command messages"\$XE "MCIavi.DRV:command messages"\$XE "MCI:command messages|described"\$XE "Command messages, MCI"\$All commands accept the optional flags MCI_NOTIFY, MCI_WAIT, and MCI_TEST flags as described in the digital-video command set.

MCIavi.DRV uses the AVIVideo keywordXE "AVIVideo keyword"\$ to identify the driver type.

MCI_CLOSEXE "MCI_CLOSE command"\$

This message releases access to a device or device element.

Parameters

DWORD *lParam1*

Specifies the MCI_NOTIFY or MCI_WAIT flag.

LPMCI_GENERIC_PARMS *lParam2*

Specifies a far pointer to the **MCI_GENERIC_PARMS** data structure.

MCI_CONFIGUREXE "MCI_CONFIGURE command"\$

This message displays a dialog box for setting the operating options.

Parameters *DWORD lParam1*

Specifies the MCI_NOTIFY, MCI_WAIT, and MCI_TEST flags.

LPMCI_GENERIC_PARMS lParam2

Specifies a far pointer to the **MCI_GENERIC_PARMS** data structure.

MCI_CUEXE "MCI_CUE command"§

This message prepares a device instance so that it can begin playback with minimum delay.

Parameters *DWORD lParam1*

The following flags apply to MCI_AVI.DRV:

MCI_DGV_CUE_OUTPUT

Specifies an instance should be cued for playing.

MCI_TO

Specifies that a workspace position is included in the **dwTo** field of the data structure identified by *lParam2*.

LPMCI_DGV_CUE_PARMS lParam2

Specifies a far pointer to the **MCI_DGV_CUE_PARMS** data structure.

XE "MCI_CUE command"§

MCI_GETDEVCAPSXE "MCI_GETDEVCAPS command"§

This message obtains static information about a device.

Parameters *DWORD lParam1*

The following flags apply to the MCI_AVI:

MCI_GETDEVCAPS_ITEM

Specifies that the **dwItem** field of the data structure identified by *lParam2* contains a constant indicating which device capability to obtain. The following constants are recognized by MCI_AVI.DRV:

MCI_GETDEVCAPS_CAN_EJECT

MCI_AVI.DRV sets the **dwReturn** field to FALSE.

MCI_GETDEVCAPS_CAN_PLAY

MCI_AVI.DRV sets the **dwReturn** field to TRUE.

MCI_GETDEVCAPS_CAN_RECORD

MCI_AVI.DRV sets the **dwReturn** field to FALSE.

MCI_GETDEVCAPS_CAN_SAVE

MCI_AVI.DRV sets the **dwReturn** field to FALSE.

MCI_GETDEVCAPS_COMPOUND_DEVICE

MCI_AVI.DRV sets the **dwReturn** field to TRUE.

MCI_GETDEVCAPS_DEVICE_TYPE

MCI_AVI.DRV sets the **dwReturn** field to
MCI_DEVTTYPE_DIGITAL_VIDEO.

MCI_GETDEVCAPS_HAS_AUDIO

MCI_AVI.DRV sets the **dwReturn** field to TRUE.

MCI_GETDEVCAPS_HAS_VIDEO

MCI_AVI.DRV sets the **dwReturn** field to TRUE.

MCI_GETDEVCAPS_USES_FILES

MCI_AVI.DRV sets the **dwReturn** field to TRUE.

MCI_DGV_GETDEVCAPS_CAN_FREEZE

MCI_AVI.DRV sets the **dwReturn** field to FALSE.

MCI_DGV_GETDEVCAPS_CAN_LOCK

MCI_AVI.DRV sets the **dwReturn** field to FALSE.

MCI_DGV_GETDEVCAPS_CAN_REVERSE

MCI_AVI.DRV sets the **dwReturn** field to FALSE.

MCI_DGV_GETDEVCAPS_CAN_STRETCH

MCI_AVI.DRV sets the **dwReturn** field to TRUE.

MCI_DGV_GETDEVCAPS_CAN_STR_IN

MCI_AVI.DRV sets the **dwReturn** field to FALSE.

MCI_DGV_GETDEVCAPS_CAN_TEST

MCI_AVI.DRV sets the **dwReturn** field to TRUE.

MCI_DGV_GETDEVCAPS_HAS_STILL

MCI_AVI.DRV sets the **dwReturn** field to FALSE.

MCI_DGV_GETDEVCAPS_PALETTES

MCI_AVI.DRV sets the **dwReturn** field to TRUE.

LPMCI_GETDEVCAPS_PARMS *lParam2*

Specifies a far pointer to the **MCI_GETDEVCAPS_PARMS** data structure.

XE "MCI_GETDEVCAPS command*"§

MCI_INFOXE "MCI_INFO command"§

This message obtains string information from a device.

Parameters

DWORD *lParam1*

The following flags apply to MCI_AVI.DRV:

MCI_INFO_PRODUCT

MCI_AVI.DRV returns Video for Windows.

MCI_DGV_INFO_TEXT

Returns the text string in the title bar of the window associated with the device instance. XE "MCI_INFO command*"§

MCI_INFO_FILE XE "MCI_INFO_FILE command"§

Obtains the path and filename of the last file specified with the **MCI_OPEN** command.

MCI_INFO_VERSION

Returns the release level of the device driver and hardware.

LPMCI_DGV_INFO_PARMS *lParam2*

Specifies a far pointer to the **MCI_DGV_INFO_PARMS** data structure.

XE "MCI_INFO_FILE command*"§

MCI_OPEN XE "MCI_OPEN command"§

This message initializes an instance of the device or device element.

Parameters

DWORD *lParam1*

The following flags apply to MCI_AVI.DRV:

MCI_OPEN_ALIAS

Specifies that an alias is referenced in the **lpstrAlias** field of the data structure identified by *lParam2*.

MCI_OPEN_TYPE

Specifies that a device-type constant or a pointer to a device-type name is included in the **lpstrDeviceType** field of the data structure identified by *lParam2*.

MCI_OPEN_TYPE_ID

Specifies that the **lpstrDeviceType** field of the data structure identified by *lParam2* contains a standard MCI device-type ID and the optional ordinal index for the device.

MCI_OPEN_ELEMENT

Specifies that an element name is included in the **lpstrElementName** field of the data structure identified by *lParam2*.

MCI_OPEN_ELEMENT_ID

Specifies that the **lpstrElementName** field of the data structure identified by *lParam2* has meaning defined by the device.

MCI_DGV_OPEN_PARENT

Indicates the parent window handle is specified in the **hWndParent** field of the data structure identified by *lParam2*.

MCI_DGV_OPEN_WS

Indicates a window style is specified in the **dwStyle** field of the data structure identified by *lParam2*.

LPMCI_DGV_OPEN_PARMS *lParam2*

Specifies a far pointer to the **MCI_DGV_OPEN_PARMS** data structure.

MCI_PAUSE "MCI_PAUSE command"§

This message pauses the current action.

Parameters

DWORD *lParam1*

Specifies the MCI_NOTIFY, MCI_TEST, and MCI_WAIT flags.

LPMCI_DGV_PAUSE_PARMS *lParam2*

Specifies a far pointer to the **MCI_DGV_PAUSE_PARMS** data structure.

XE "MCI_PAUSE command"§

MCI_PLAY "MCI_PLAY command"§

This message begin play of the audio and video.

Parameters

DWORD *lParam1*

The following flags apply to MCI_AVI.DRV:

MCI_FROM

Specifies that a starting position is included in the **dwFrom** field of the data structure identified by *lParam2*.

MCI_TO

Specifies that an ending position is included in the **dwTo** field of the data structure identified by *lParam2*. MCI_AVI.DRV returns an error if the "to" position is less than the "from" position.

MCI_DGV_PLAY_WINDOW

Specifies that playing should occur in the window associated with a device instance (the default). (This flag is specific to the MCI_AVI.DRV.)

MCI_MCI_AVI_PLAY_FULLSCREEN

Specifies that playing should use a full-screen display, typically, with a 320-by-200 resolution. The full-screen display takes over the entire desktop. (This flag is specific to MCI_AVI.DRV.)

LPMCI_DGV_PLAY_PARMS *lParam2*

Specifies a far pointer to an **MCI_DGV_PLAY_PARMS** data structure.

XE "MCI_PLAY command"§

MCI_PUTXE "MCI_PUT command"§

This message specifies a rectangular region that describes a cropping or scaling option.

Parameters

DWORD *lParam1*

The following flags apply to MCI_AVI.DRV:

MCI_DGV_RECT

Specifies that the **rc** field of the data structure identified by *lParam2* contains a valid rectangle.

MCI_DGV_PUT_DESTINATION

Indicates the rectangle defined for MCI_DGV_RECT specifies a destination rectangle. The destination rectangle specifies the portion of the client window associated with this device driver instance that shows the image or video.

MCI_DGV_PUT_SOURCE

Indicates the rectangle defined for MCI_DGV_RECT specifies a source rectangle. The source rectangle specifies which portion of the frame buffer is to be scaled to fit into the destination rectangle.

MCI_DGV_PUT_WINDOW

Indicates that the rectangle defined for MCI_DGV_RECT applies to the display window. This rectangle is relative to the parent window of the display window (usually the desktop). If the window is not specified, it defaults to the initial window size and position.

LPMCI_DGV_PUT_PARMS *lParam2*

Specifies a far pointer to a **MCI_DGV_PUT_PARMS** data structure.

XE "MCI_PUT command"§

MCI_REALIZEXE "MCI_REALIZE command"§

This message tells MCI_AVI to select and realize its palette into a display context of the displayed window. You should use this message when your application receives the WM_QUERYNEWPALETTE message from Windows.

Parameters

DWORD *lParam1*

The following flags apply to MCI_AVI.DRV:

MCI_DGV_REALIZE_BKGD

Realizes the palette as a background palette.

MCI_DGV_REALIZE_NORM

Realizes the palette normally (the default).

LPMCI_GENERIC_PARMS *lParam2*

Specifies a far pointer to a **MCI_GENERIC_PARMS** data structure.

XE "MCI_REALIZE command*"§

MCI_RESUMEXE "MCI_RESUME command"§

This message resumes MCIavi operation when it is paused .

Parameters

DWORD *lParam1*

Specifies the MCI_NOTIFY, MCI_WAIT, and MCI_TEST flags.

LPMCI_GENERIC_PARMS *lParam2*

Specifies a far pointer to the **MCI_GENERIC_PARMS** data structure.

XE "MCI_RESUME command*"§

MCI_SEEKXE "MCI_SEEK command"§

This message positions and cues the workspace to the specified position showing the specified frame.

Parameters

DWORD *lParam1*

The following flags apply to MCIavi.DRV:

MCI_SEEK_TO_END

Specifies the seek should move to the end of the workspace.

MCI_SEEK_TO_START

Specifies the seek should move to the beginning of the workspace.

MCI_TO

Specifies a position is included in the **dwTo** field of the

MCI_SEEK_PARMS data structure.

LPMCI_SEEK_PARMS *lParam2*

Specifies a far pointer to the **MCI_SEEK_PARMS** data structure.

XE "MCI_SEEK command*"§

MCI_SETXE "MCI_SET command"§

This message sets device information.

Parameters

DWORD *lParam1*

The following flags apply to MCIavi.DRV:

MCI_SET_AUDIO

Specifies an audio-channel number is included in the **dwAudio** field of the data structure identified by *lParam2*. This flag must be used with MCI_SET_ON or MCI_SET_OFF. Specify the constant

MCI_SET_AUDIO_ALL for the channel number.

MCI_SET_TIME_FORMAT

Specifies a time-format parameter is included in the **dwTimeFormat** field of the data structure identified by *lParam2*. Constants defined for time formats include:

MCI_FORMAT_FRAMES

Specifies frames.

MCI_FORMAT_MILLISECONDS

Specifies milliseconds.

MCI_SET_VIDEO

Sets the video signal on or off. This flag must be used with either MCI_SET_ON or MCI_SET_OFF.

MCI_SET_ON

Enables the video or audio channel, or enables the seek-exactly mode.

MCI_SET_OFF

Disables the video or audio channel, or disables the seek-exactly mode.

MCI_DGV_SET_SEEK_EXACTLY

Sets the format used for positioning. This flag must be used with MCI_SET_ON or MCI_SET_OFF.

MCI_DGV_SET_SPEED

Specifies that a speed parameter is included in the **dwSpeed** field of the data structure identified by *lParam2*.

LPMCI_DGV_SET_PARMS *lParam2*

Specifies a far pointer to the **MCI_DGV_SET_PARMS** data structure.

MCI_SETAUDIOX "MCI_SETAUDIO command"§

This message gets various values associated with audio playback and capture.

Parameters

DWORD *lParam1*

The following flags apply to MCI_AVI.DRV:

MCI_DGV_SETAUDIO_ITEM

Indicates an audio constant is specified in the **dwAdjustParm** field of the data structure identified by *lParam2*. The following constant is supported by MCI_AVI.DRV:

MCI_DGV_SETAUDIO_VOLUME

Indicates that the audio level is specified as a factor in the **dwValue** field of the data structure identified by *lParam2*. The volume level ranges between 0 and 1000.

MCI_DGV_SETAUDIO_VALUE

Indicates that an audio value is specified in the **dwValue** field of the data structure identified by *lParam2*.

MCI_SET_ON

Enables the audio channel.

MCI_SET_OFF

Disables the audio channel.

LPMCI_DGV_SETAUDIO_PARMS *lParam2*

Specifies a far pointer to the **MCI_DGV_SETAUDIO_PARMS** data structure.

XE "MCI_SETAUDIO command"§

MCI_SETVIDEOXE "MCI_SETVIDEO command"§

This message sets various values associated with video playback.

Parameters

DWORD *lParam1*

The following flags apply to MCI.avi.DRV:

MCI_DGV_SETVIDEO_ITEM

Indicates a video constant is specified in the **dwAdjustParm** field of the data structure identified by *lParam2*. MCI.avi.DRV supports the following constant:

MCI_DGV_SETVIDEO_PALHANDLE

Indicates that a palette-handle value is specified in the **dwValue** field of the data structure identified by *lParam2*.

MCI_DGV_SETVIDEO_SRC_VALUE

Specifies a value is included in the **dwValue** field of the data structure identified by *lParam2*.

MCI_SET_ON

Enables video output.

MCI_SET_OFF

Disables video output.

LPMCI_DGV_SETVIDEO_PARMS *lParam2*

Specifies a far pointer to the **MCI_DGV_SETVIDEO_PARMS** data structure.

XE "MCI_SETVIDEO command"§

MCI_SIGNALXE "MCI_SIGNAL command"§

This message sets a specified position in the workspace. MCI.avi.DRV supports only one active signal at a time.

Parameters

DWORD *lParam1*

The following flags apply to all devices supporting **MCI_SIGNAL** :

MCI_DGV_SIGNAL_AT

Specifies a signal position is included in the **dwPosition** field of the data structure identified by *lParam2*.

MCI_DGV_SIGNAL_EVERY

Specifies a signal-period value is included in the **dwEvery** field of the data structure identified by *lParam2*.

MCI_DGV_SIGNAL_USERVAL

Specifies a data value is included in the **dwUserParm** field of the data structure identified by *lParam2*. The data value associated with this request is reported back with the Windows message.

MCI_DGV_SIGNAL_CANCEL

Removes the signal position specified by the value associated with the **MCI_DGV_SIGNAL_USERVAL** flag.

MCI_DGV_SIGNAL_POSITION

Specifies that the device should send the position value with the Windows message instead of the user-specified value.

LPMCI_DGV_SIGNAL_PARMS *lParam2*

Specifies a far pointer to the **MCI_DGV_SIGNAL_PARMS** structure.

XE "MCI_SIGNAL command"§

MCI_STATUSXE "MCI_STATUS command"§

This message obtains information about an instance of an MCI device.

Parameters

DWORD *lParam1*

The following flags apply to MCI_AVI.DRV:

MCI_STATUS_ITEM

Specifies that the **dwItem** field of the data structure identified by *lParam2* contains a constant specifying which status item to obtain.

MCI_AVI.DRV supports the following constants:

MCI_STATUS_LENGTH

MCI_AVI.DRV sets the **dwReturn** field to the media length. (It returns an error for any track but 1.)

MCI_STATUS_MODE

MCI_AVI.DRV sets the **dwReturn** field to the current mode of the device.

MCI_STATUS_NUMBER_OF_TRACKS

MCI_AVI.DRV sets the **dwReturn** field to 1.

MCI_STATUS_POSITION

MCIAVI.DRV sets the **dwReturn** field to the position of the track.
(It returns an error for any track but 1.)

MCI_STATUS_READY

MCIAVI.DRV sets the **dwReturn** field to TRUE if the device is ready to accept another command.

MCI_STATUS_TIME_FORMAT

MCIAVI.DRV sets the **dwReturn** to the current time format.

MCI_DGV_STATUS_AUDIO

MCIAVI.DRV sets the **dwReturn** field to MCI_ON or MCI_OFF, depending on the most recent MCI_SET_AUDIO option for the **MCI_SET** command.

MCI_DGV_STATUS_FILEFORMAT

MCIAVI.DRV returns the constant for AVI RIFF in the **dwReturn** field.

MCI_DGV_STATUS_FORWARD

MCIAVI.DRV sets the **dwReturn** field to TRUE.

MCI_DGV_STATUS_MEDIA_PRESENT

MCIAVI.DRV sets the **dwReturn** field to TRUE.

MCI_DGV_STATUS_MONITOR

MCIAVI.DRV sets the **dwReturn** field to MCI_DGV_MONITOR_FILE.

MCI_DGV_STATUS_HPAL

MCIAVI.DRV sets the **dwReturn** field to the current palette handle.

MCI_DGV_STATUS_HWND

MCIAVI.DRV sets the **dwReturn** field to the window handle.

MCI_DGV_STATUS_NOMINAL_RATE

MCIAVI.DRV sets the **dwReturn** field to the nominal frame rate associated with the file.

MCI_DGV_STATUS_SIZE

MCIAVI.DRV sets the **dwReturn** field to zero.

MCI_DGV_STATUS_SEEK_EXACTLY

MCIAVI.DRV sets the **dwReturn** field to TRUE or FALSE indicating whether or not seek exactly is set.

MCI_DGV_STATUS_SPEED

MCIAVI.DRV sets the **dwReturn** field to the current playback speed.

MCI_DGV_STATUS_UNSAVED

MCIAVI.DRV sets the **dwReturn** field to FALSE.

MCI_DGV_STATUS_VIDEO

MCI_AVI.DRV indicates whether the video is enabled or disabled in the **dwReturn** field.

MCI_DGV_STATUS_WINDOW_VISIBLE

MCI_AVI.DRV sets the **dwReturn** field to TRUE if the window is not hidden.

MCI_DGV_STATUS_WINDOW_MINIMIZED

MCI_AVI.DRV sets the **dwReturn** field to TRUE if the window is minimized.

MCI_DGV_STATUS_WINDOW_MAXIMIZED

MCI_AVI.DRV sets the **dwReturn** field to TRUE if the window is maximized.

MCI_STATUS_START

Obtains the starting position of the media. To get the starting position, combine this flag with MCI_STATUS_ITEM and set the **dwItem** field of the data structure, identified by *lParam2*, to MCI_STATUS_POSITION.

MCI_DGV_STATUS_REFERENCE

The **dwReference** field returns the nearest previous keyframe.

LPMCI_DGV_STATUS_PARMS *lParam2*

Specifies a far pointer to the **MCI_DGV_STATUS_PARMS** data structure.

XE "MCI_STATUS command*"§

MCI_STEPXE "MCI_STEP command"§

This message steps the player one or more frames.

Parameters

DWORD *lParam1*

The following flags apply to MCI_AVI.DRV:

MCI_DGV_STEP_FRAMES

Indicates that the **dwFrames** field of the data structure identified by *lParam2* specifies the number of frames to advance before displaying another image.

MCI_DGV_STEP_REVERSE

Steps in reverse.

LPMCI_DGV_STEP_PARMS *lParam2*

Specifies a far pointer to the **MCI_DGV_STEP_PARMS** data structure.

XE "MCI_STEP command*"§

MCI_STOPXE "MCI_STOP command"§

This message stops all play sequences and ceases display of video images.

ParametersDWORD *lParam1*

Specifies the MCI_NOTIFY, MCI_WAIT, and MCI_TEST flags.

LPMCI_DGV_STOP_PARDS *lParam2*Specifies a far pointer to the **MCI_DGV_STOP_PARDS** data structure.

MCI_UPDATEEXE "MCI_UPDATE command"§

This message repaints the current frame into the specified display context.

ParametersDWORD *lParam1*

The following flags apply to MCI_AVI.DRV:

MCI_UPDATE_HDCSpecifies that the **hDC** field of the data structure identified by *lParam2* contains a valid window of the display context to paint.**MCI_DGV_UPDATE_PAINT**

An application uses this flag when it receives a WM_PAINT message that is intended for a display DC. A frame-buffer device will usually paint the key color. If the display device does not have a frame buffer, it might ignore the MCI_UPDATE message when the MCI_DGV_UPDATE_PAINT flag is used, because the display will be repainted during the playback operation.

LPMCI_DGV_UPDATE_PARDS *lParam2*Specifies a far pointer to a **MCI_DGV_UPDATE_PARDS** data structure.

XE "MCI_UPDATE command"§

MCI_WHEREEXE "MCI_WHERE command"§

This message returns the rectangular region that has been specified with the **MCI_PUT** command.**Parameters**DWORD *lParam1*

The following flags apply to MCI_AVI.DRV:

MCI_DGV_WHERE_DESTINATION

Obtains a description of the rectangular region used to display video and images in the client area of the current window.

MCI_DGV_WHERE_SOURCE

Obtains a description of the rectangular region (cropped from the frame buffer) that is stretched to fit the destination rectangle on the display.

MCI_DGV_WHERE_MAX

When used with MCI_DGV_WHERE_DESTINATION or

MCI_DGV_WHERE_SOURCE, the rectangle returned indicates the maximum width and height of the specified region.

MCI_DGV_WHERE_WINDOW

Obtains a description of the display window frame.

LPMCI_DGV_RECT_PARMS *lParam2*

Specifies a far pointer to a **MCI_DGV_RECT_PARMS** data structure.

MCI_WINDOWXE "MCI_WINDOW command"§

This message specifies the window and the window characteristics for graphic devices.

Parameters

DWORD *lParam1*

The following flags apply to MCI_AVI.DRV:

MCI_DGV_WINDOW_HWND

Indicates that the handle of the window needed for use as the destination is included in the **hWnd** field of the data structure identified by *lParam2*.

MCI_DGV_WINDOW_STATE

Indicates the **nCmdShow** field of the **MCI_DGV_WINDOW_PARMS** data structure contains parameters for setting the window state.

MCI_DGV_WINDOW_TEXT

Indicates the **lpstrText** field of the **MCI_DGV_WINDOW_PARMS** data structure contains a pointer to a buffer containing the caption used in the window title bar.

LPMCI_DGV_WINDOW_PARMS *lParam2*

Specifies a far pointer to a **MCI_DGV_WINDOW_PARMS** data structure.

XE "MCI_WINDOW command"§

MM_MCISIGNALXE "MM_MCISIGNAL command"§

This message is sent to a window to notify an application that an MCI device has reached a position defined in a previous **MCI_SIGNAL** to the device.

Parameters

WORD *wParam*

Contains the ID of the device initiating the signal message.

LONG *lParam*

Normally this contains the value passed in **dwUserParm** when the **MCI_SIGNAL** message has defined this callback. Alternatively, it might contain the position value.

MM_MCI SIGNAL command**§

Data Structures for MCI Command Messages

XE "MCI devices:command message data structures"§XE "AVI video devices:command message data structures"§XE "MCI:command message data structures"§XE "MCI\AVI.DRV:command message data structures"§XE "Command message data structures, MCI"§The following data structures are used by the MCI command messages for MCI\AVI.DRV.

MCI_DGV_CUE_PARMSXE "MCI_DGV_CUE_PARMS structure"§

The **MCI_DGV_CUE_PARMS** structure contains parameters used by the **MCI_CUE** message for digital video devices. When assigning data to the fields in the following data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field:

```
typedef struct {
    DWORD dwCallback;
    DWORD dwTo;
} MCI_DGV_CUE_PARMS;
```

Fields

The **MCI_DGV_CUE_PARMS** structure has the following fields:

dwCallback

The low-order word specifies a window handle used for the MCI_NOTIFY flag.

dwTo

Specifies the cue position.

XE "MCI_DGV_CUE_PARMS structure**§

MCI_DGV_INFO_PARMSXE "MCI_DGV_INFO_PARMS structure"§

The **MCI_DGV_INFO_PARMS** structure contains parameters used by the **MCI_INFO** message for digital video devices. When assigning data to the fields in the following data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field:

```
typedef struct {
    DWORD dwCallback;
    LPSTR lpstrReturn;
    DWORD dwRetSize;
} MCI_DGV_INFO_PARMS;
```

Fields

The **MCI_DGV_INFO_PARMS** structure has the following fields:

dwCallback

The low-order word specifies a window handle used for the MCI_NOTIFY flag.

lpstrReturn

Specifies a long pointer to a user-supplied buffer for the return string.

dwRetSize

Specifies the size in bytes of the buffer for the return string.

MCI_DGV_OPEN_PARMSXE "MCI_DGV_OPEN_PARMS structure"§

The **MCI_DGV_OPEN_PARMS** structure contains information used by **MCI_OPEN** message for digital video devices. When assigning data to the fields in the following data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field:

```
typedef struct {  
    DWORD dwCallback;  
    WORD  wDeviceID;  
    WORD  wReserved0;  
    LPSTR lpstrDeviceType;  
    LPSTR lpstrElementName;  
    LPSTR lpstrAlias;  
    DWORD dwStyle;  
    WORD  hWndParent;  
    WORD  wReserved1;  
} MCI_DGV_OPEN_PARMS;
```

Fields

The **MCI_DGV_OPEN_PARMS** structure has the following fields:

dwCallback

The low-order word specifies a window handle used for the MCI_NOTIFY flag.

wDeviceID

Contains the device ID returned to user.

wReserved0

Reserved.

lpstrDeviceType

Specifies the name or constant ID of the device type.

lpstrElementName

Specifies the device-element name (usually a path).

lpstrAlias

Specifies an optional device alias.

dwStyle

Specifies the window style.

MCI_DGV_INFO_PARMS
ure**§

hWndParent

Specifies the handle to use as the window parent.

wReserved1

Reserved.

```
XE "MCI_DGV_OPEN_PARMS structure"§
```

```
MCI_DGV_PAUSE_PARMSXE "MCI_DGV_PAUSE_PARMS structure"§
```

The **MCI_DGV_PAUSE_PARMS** structure contains information used by the **MCI_PAUSE** command. When assigning data to the fields in the following data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field:

```
typedef struct {
    DWORD dwCallback;
} MCI_DGV_PAUSE_PARMS;
```

Fields

The **MCI_DGV_PAUSE_PARMS** structure has the following field:

dwCallback

The low-order word specifies a window handle used for the MCI_NOTIFY flag.

```
XE "MCI_DGV_PAUSE_PARMS structure"§
```

```
MCI_DGV_PLAY_PARMSXE "MCI_DGV_PLAY_PARMS structure"§
```

The **MCI_DGV_PLAY_PARMS** structure contains parameters use by the **MCI_PLAY** message for digital video devices. When assigning data to the fields in the following data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field:

```
typedef struct {
    DWORD dwCallback;
    DWORD dwFrom;
    DWORD dwTo;
} MCI_DGV_PLAY_PARMS;
```

Fields

The **MCI_DGV_PLAY_PARMS** structure has the following fields:

dwCallback

The low-order word specifies a window handle used for the MCI_NOTIFY flag.

dwFrom

Specifies the position to play from.

dwTo

Specifies the position to play to. `MCI_DGV_PLAY_PARMS` structure*"

`MCI_DGV_PUT_PARMS` `MCI_DGV_PUT_PARMS` structure"

The **MCI_DGV_PUT_PARMS** structure contains parameters used by the **MCI_PUT** message for digital video devices. When assigning data to the fields in the following data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field:

```
typedef struct {
    DWORD dwCallback;
    RECT rc;
} MCI_DGV_PUT_PARMS;
```

Fields

The **MCI_DGV_PUT_PARMS** structure has the following fields:

dwCallback

The low-order word specifies a window handle used for the **MCI_NOTIFY** flag.

rc

Specifies a rectangle.

`MCI_DGV_SIGNAL_PARMS` `MCI_DGV_SIGNAL_PARMS` structure"

The **MCI_DGV_SIGNAL_PARMS** structure contains parameters for the **MCI_SIGNAL** message used by digital video devices. When assigning data to the fields in this data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field.

```
typedef struct {
    DWORD dwCallback;
    DWORD dwPosition;
    DWORD dwPeriod;
    DWORD dwUserParam;
} MCI_DGV_SIGNAL_PARMS;
```

Fields

The **MCI_DGV_SIGNAL_PARMS** structure has the following fields:

dwCallback

The low-order word specifies a window handle used for the **MCI_NOTIFY** flag.

dwPosition

Specifies the position to be marked.

`MCI_DGV_PUT_PARMS`
structure*"

dwPeriod

Specifies the interval of the position marks.

dwUserParm

Specifies a value associated with the signals being set.

XE "MCI_DGV_SIGNAL_PARMS structure"§

MCI_DGV_RECT_PARMSXE "MCI_DGV_RECT_PARMS structure"§

The **MCI_DGV_RECT_PARMS** structure contains parameters used by the **MCI_FREEZE**, **MCI_PUT**, **MCI_UNFREEZE**, and **MCI_WHERE** messages for digital video devices. When assigning data to the fields in the following data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field:

```
typedef struct {
    DWORD dwCallback;
    RECT rc;
} MCI_DGV_RECT_PARMS;
```

Fields

The **MCI_DGV_RECT_PARMS** structure has the following fields:

dwCallback

The low-order word specifies a window handle used for the MCI_NOTIFY flag.

rc

Specifies a rectangle.

MCI_DGV_RECT_PARMS structure"§

MCI_DGV_SET_PARMSXE "MCI_DGV_SET_PARMS structure"§

The **MCI_DGV_SET_PARMS** structure contains parameters used by the **MCI_SET** message for digital video devices. When assigning data to the fields in the following data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field:

```
typedef struct {
    DWORD dwCallback;
    DWORD dwTimeFormat;
    DWORD dwAudio;
    DWORD dwFileFormat;
    DWORD dwSpeed;
} MCI_DGV_SET_PARMS;
```

Fields

The **MCI_DGV_SET_PARMS** structure has the following fields:

dwCallback

The low-order word specifies a window handle used for the MCI_NOTIFY flag.

dwTimeFormat

Specifies the time format used by the device.

dwAudio

Specifies the channel used for audio output.

dwFileFormat

Specifies the file format.

dwSpeed

Specifies the playback speed.

XE "MCI_DGV_SET_PARMS structure"§

MCI_DGV_SETAUDIO_PARMSXE "MCI_DGV_SETAUDIO_PARMS structure"§

The **MCI_DGV_SETAUDIO_PARMS** structure contains parameters used by the **MCI_SETAUDIO** message for digital video devices. When assigning data to the fields in the following data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field:

```
typedef struct {  
    DWORD dwCallback;  
    DWORD dwItem;  
    DWORD dwValue;  
    DWORD dwOver;  
    LPSTR lpstrAlgorithm;  
    LPSTR lpstrQuality;  
} MCI_DGV_SETAUDIO_PARMS;
```

Fields

The **MCI_DGV_SETAUDIO_PARMS** structure has the following fields:

dwCallback

The low-order word specifies a window handle used for the MCI_NOTIFY flag.

dwItem

Specifies the constant indicating the target adjustment.

dwValue

Specifies the adjustment level.

dwOver

Specifies the transition-length parameter.

lpstrAlgorithm

Specifies a long pointer to a null-terminated string containing the name of the audio-compression algorithm.

lpstrQuality

Specifies a long pointer to a null-terminated string containing a descriptor of the audio-compression algorithm.

XE "MCI_DGV_SETAUDIO_PARMS structure**§

MCI_DGV_SETVIDEO_PARMS XE "MCI_DGV_SETVIDEO_PARMS structure"§

The **MCI_DGV_SETVIDEO_PARMS** structure contains parameters used by the **MCI_SETVIDEO** message for digital video devices. When assigning data to the fields in the following data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field:

```
typedef struct {
    DWORD dwCallback;
    DWORD dwItem;
    DWORD dwValue;
    DWORD dwOver;
    LPSTR lpstrQuality;
    LPSTR lpstrAlgorithm;
    DWORD dwSourceNumber;
} MCI_DGV_SETVIDEO_PARMS;
```

Fields

The **MCI_DGV_SETVIDEO_PARMS** structure has the following fields:

dwCallback

The low-order word specifies a window handle used for the MCI_NOTIFY flag.

dwItem

Specifies the constant indicating the target adjustment.

dwValue

Specifies the adjustment level.

dwOver

Specifies the transition-length parameter.

lpstrQuality

Specifies a long pointer to a null-terminated string containing a descriptor of the video-compression algorithm.

lpstrAlgorithm

Specifies a long pointer to a null-terminated string containing the name of the video-compression algorithm.

dwSourceNumber

Specifies the index of input source.

XE "MCI_DGV_SETVIDEO_PARDS structure"§

MCI_DGV_STATUS_PARDSXE "MCI_DGV_STATUS_PARDS structure"§

The **MCI_DGV_STATUS_PARDS** structure contains parameters used by the **MCI_STATUS** message for digital video devices. When assigning data to the fields in the following data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field:

```
typedef struct {  
    DWORD dwCallback;  
    DWORD dwReturn;  
    DWORD dwItem;  
    DWORD dwTrack;  
    LPSTR lpstrDrive;  
    DWORD dwReference;  
} MCI_DGV_STATUS_PARDS;
```

Fields

The **MCI_DGV_STATUS_PARDS** structure has the following fields:

dwCallback

The low-order word specifies a window handle used for the MCI_NOTIFY flag.

dwReturn

Contains the return information on exit.

dwItem

Identifies the capability being queried.

dwTrack

Specifies the length or number of tracks.

lpstrDrive

Specifies the approximate amount of disk space that can be obtained by a **MCI_RESERVE** command.

dwReference

Specifies the approximate location of nearest, previous intraframe-encoded image.

XE "MCI_DGV_STATUS_PARMS structure*"§

MCI_DGV_STEP_PARMSXE "MCI_DGV_STEP_PARMS structure"§

The **MCI_DGV_STEP_PARMS** structure contains parameters used by the **MCI_STEP** message for digital video devices. When assigning data the fields in the following data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field:

```
typedef struct {
    DWORD dwCallback;
    DWORD dwFrames;
} MCI_DGV_STEP_PARMS;
```

Fields

The **MCI_DGV_STEP_PARMS** structure has the following fields:

dwCallback

The low-order word specifies a window handle used for the MCI_NOTIFY flag.

dwFrames

Specifies the number of frames to step.

XE "MCI_DGV_STOP_PARMS structure*"§

MCI_DGV_STOP_PARMSXE "MCI_DGV_STOP_PARMS structure"§

The **MCI_DGV_STOP_PARMS** structure contains the information used by **MCI_STOP** command message for digital video devices. When assigning data to the fields in the following data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field:

```
typedef struct {
    DWORD dwCallback;
} MCI_DGV_STOP_PARMS;
```

Fields

The **MCI_DGV_STOP_PARMS** structure has the following field:

dwCallback

The low-order word specifies a window handle used for the MCI_NOTIFY flag.

XE "MCI_DGV_UPDATE_PARMS structure*"§

MCI_DGV_UPDATE_PARMSXE "MCI_DGV_UPDATE_PARMS structure"§

The **MCI_DGV_UPDATE_PARMS** structure contains parameters used by the **MCI_UPDATE** message. When assigning data to the fields in the following data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field:

```
typedef struct {
    DWORD dwCallback;
    RECT rc;
    HDC hDC;
    WORD wReserved0;
} MCI_DGV_UPDATE_PARMS;
```

Fields

The **MCI_DGV_UPDATE_PARMS** structure has the following fields:

dwCallback

The low-order word specifies a window handle used for the MCI_NOTIFY flag.

rc

Specifies a rectangle.

hDC

Specifies a handle to a display context.

wReserved0

Reserved.

MCI_DGV_UPDATE_PARMS
structure"§

MCI_DGV_WINDOW_PARMSXE "MCI_DGV_WINDOW_PARMS
structure"§

The **MCI_DGV_WINDOW_PARMS** structure contains parameters used by the **MCI_WINDOW** message for digital video devices. When assigning data to the fields in the following data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field:

```
typedef struct {
    DWORD dwCallback;
    WORD hWnd;
    WORD wReserved1;
    WORD nCmdShow;
    WORD wReserved2;
    LPSTR lpstrText;
} MCI_DGV_WINDOW_PARMS;
```

Fields

The **MCI_DGV_WINDOW_PARMS** structure has the following fields:

dwCallback

The low-order word specifies a window handle used for the MCI_NOTIFY flag.

hWnd

Specifies a handle to the display window.

wReserved1

Reserved.

nCmdShow

Specifies how the window is displayed.

wReserved2

Reserved.

lpstrText

Specifies a long pointer to a null-terminated string containing the window caption.XE "MCI_DGV_WINDOW_PARMS structure*"§

MCI_GENERIC_PARMS XE "MCI_GENERIC_PARMS structure"§

The **MCI_GENERIC_PARMS** structure contains the information used by MCI command messages that have empty parameter lists. When assigning data to the fields in the following data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field:

```
typedef struct {
    DWORD dwCallback;
} MCI_GENERIC_PARMS;
```

Fields

The **MCI_GENERIC_PARMS** structure has the following field:

dwCallback

The low-order word specifies a window handle used for the MCI_NOTIFY flag.

MCI_GETDEVCAPS_PARMS XE "MCI_GETDEVCAPS_PARMS structure"§

The **MCI_GETDEVCAPS_PARMS** structure contains parameters for the **MCI_GETDEVCAPS** message. When assigning data to the fields in the following data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field:

```
typedef struct {
    DWORD dwCallback;
    DWORD dwReturn;
    DWORD dwItem;
} MCI_GETDEVCAPS_PARMS;
```

Fields

The **MCI_GETDEVCAPS_PARMS** structure has the following fields:

dwCallback

The low-order word specifies a window handle used for the MCI_NOTIFY flag.

dwReturn

Contains the return information on exit.

dwItem

Identifies the capability being queried.XE "MCI_GETDEVCAPS_PARMS structure*"§

MCI_SEEK_PARMSXE "MCI_SEEK_PARMS structure"§

The **MCI_SEEK_PARMS** structure contains parameters used by the **MCI_SEEK** message. When assigning data to the fields in the following data structure, set the corresponding MCI flags in the *lParam1* parameter of **mciSendCommand** to validate each field:

```
typedef struct {  
    DWORD dwCallback;  
    DWORD dwTo;  
} MCI_SEEK_PARMS;
```

Fields

The **MCI_SEEK_PARMS** structure has the following fields:

dwCallback

The low-order word specifies a window handle used for the MCI_NOTIFY flag.

dwTo

Specifies the seek position.XE "MCI_SEEK_PARMS structure*"§XE "Command message data structures, MCI*"§XE "MCI\AVI.DRV:command message data structures*"§XE "MCI:command message data structures*"§XE "AVI video devices:command message data structures*"§XE "MCI devices:command message data structures*"§XE "Command messages, MCI*"§XE "MCI:command messages|described*"§XE "MCI\AVI.DRV:command messages*"§XE "AVI video devices:command messages*"§XE "MCI devices:command messages*"§