DVD Playback Framework Reference

Graphics & Imaging > Video

2007-06-28

Contents

Tables

DVD Playback Framework Reference

Framework:	DVDPlayback
Companion guide	DVD Playback Services Programming Guide
Declared in	DVDPlayback.h

Overview

DVD Playback is a core technology introduced in Mac OS X version 10.3. Mac OS X applications can use DVD Playback to display a DVD-Video recording located on an optical disc or a mass storage device such as a hard drive. DVD Playback makes it easy for applications to incorporate basic video playback features such as selecting a title from a menu and playing the title, as well as advanced features such as bookmarks, video clips, and multiple viewing angles.

Functions by Task

Starting and Ending Playback Sessions

`DVDInitialize` (page 45)
>   Starts a new DVD-Video playback session.

`DVDSetFatalErrorCallBack` (page 61)
>   Registers a callback function to handle unrecoverable errors.

`DVDDispose` (page 18)
>   Ends the current DVD-Video playback session.

Opening and Closing DVD Media

`DVDIsValidMediaRef` (page 49)
>   Determines whether a VIDEO_TS media folder has a valid layout.

`DVDOpenMediaFile` (page 51)
>   Opens a VIDEO_TS media folder located on a mass storage device.

`DVDOpenMediaVolume` (page 52)
>   Opens a VIDEO_TS media folder located on a DVD-Video disc.

`DVDHasMedia` (page 43)
>   Determines whether DVD-Video media is open for playback.

`DVDCloseMediaFile` (page 17)

> Closes a previously opened VIDEO_TS media folder on a mass storage device.

`DVDCloseMediaVolume` (page 17)

> Closes a previously opened VIDEO_TS media folder on a DVD-Video disc.

Accessing the Playback Device Using a GDevice Record

`DVDGetVideoDevice` (page 101) Deprecated in Mac OS X v10.5

> Gets the graphics device for the current DVD-Video playback session. (Deprecated. Use `DVDGetVideoDisplay` (page 40) instead.)

`DVDIsSupportedDevice` (page 102) Deprecated in Mac OS X v10.5

> Determines whether a graphics device supports DVD-Video playback. (Deprecated. Use `DVDIsSupportedDisplay` (page 49) instead.)

`DVDSetVideoDevice` (page 104) Deprecated in Mac OS X v10.5

> Sets the graphics device for the current DVD-Video playback session. (Deprecated. Use `DVDSetVideoDisplay` (page 65) instead.)

`DVDSwitchToDevice` (page 105) Deprecated in Mac OS X v10.5

> If a graphics device supports DVD-Video playback, sets the device for the current playback session. (Deprecated. Use `DVDSwitchToDisplay` (page 68) instead.)

Accessing the Playback Device Using a Quartz Display ID

`DVDIsSupportedDisplay` (page 49)

> Determines whether a graphics display supports DVD-Video playback.

`DVDSetVideoDisplay` (page 65)

> Sets the graphics display for the current DVD-Video playback session.

`DVDSwitchToDisplay` (page 68)

> If a specified graphics display supports DVD-Video playback, sets the display for the current playback session.

`DVDGetVideoDisplay` (page 40)

> Gets the graphics display for the current DVD-Video playback session.

Accessing Video Information

`DVDSetVideoWindowID` (page 65)

> Enables or disables DVD-Video playback in a Cocoa window.

`DVDGetVideoWindowID` (page 41)

> Gets the window ID of the Cocoa window in use for DVD-Video playback.

`DVDGetNativeVideoSize` (page 34)

> Gets the native width and height in pixels of the current title.

`DVDGetAspectRatio` (page 20)

> Gets the aspect ratio of the current title or menu.

`DVDGetFormatStandard` (page 30)

> Gets the video broadcast format of current DVD-Video media.

`DVDGetVideoBounds` (page 100) Deprecated in Mac OS X v10.5

    Gets the bounds in which to display the video. (Deprecated. Use `DVDGetVideoCGBounds` instead.)

`DVDGetVideoPort` (page 102) Deprecated in Mac OS X v10.5

    Gets the graphics port of the Carbon window in use for DVD-Video playback. (Deprecated. Use `DVDGetVideoWindowRef` instead.)

`DVDSetVideoBounds` (page 103) Deprecated in Mac OS X v10.5

    Sets the bounds in which to display the video. (Deprecated. Use `DVDSetVideoCGBounds` instead.)

`DVDSetVideoPort` (page 104) Deprecated in Mac OS X v10.5

    Enables or disables DVD-Video playback in a Carbon window. (Deprecated. Use `DVDSetVideoWindowRef` instead.)

Accessing Audio Streams

These functions are used to access audio streams in the current title.

`DVDSetAudioStream` (page 57)

    Sets the audio stream to use.

`DVDGetAudioStream` (page 22)

    Gets the audio stream currently being used.

`DVDGetNumAudioStreams` (page 35)

    Gets the number of audio streams currently available.

`DVDGetAudioStreamFormat` (page 23)

    Gets the current audio stream format.

`DVDGetAudioStreamFormatByStream` (page 23)

    Gets the audio format for a specified stream.

Accessing the Playback Time Position

You can use these functions to access the current video playback position relative to the beginning or end of the title.

`DVDSetTime` (page 63)

    Sets the video playback position relative to the specified time code (elapsed, remaining).

`DVDGetTime` (page 39)

    Gets the current video playback position in the current title or menu, relative to the specified time code (elapsed, remaining).

Controlling Playback

`DVDEnableWebAccess` (page 19)

    Turns DVD@ccess support on or off.

`DVDIsPlaying` (page 48)

    Determines if media is open and playing (even if paused).

`DVDIsPaused` (page 47)

    Determines if media is open and playing, and video playback is paused.

DVDPlay (page 53)

> Starts playing the media, if currently stopped.

DVDPause (page 52)

> Pauses the media, if currently playing.

DVDResume (page 55)

> Resumes playing the media, if currently paused.

DVDStop (page 67)

> Stops playing the current title. If video playback is already stopped, the current video playback position is cleared.

DVDScan (page 55)

> Fast forwards or rewinds at the specified scan rate.

DVDGetScanRate (page 37)

> Gets the current scan direction and scan rate.

DVDGetState (page 37)

> Gets the state of the current DVD-Video playback session.

DVDStepFrame (page 66)

> Steps one frame in the forward direction.

DVDSleep (page 66)

> Saves video playback state information and suspends playback.

DVDWakeUp (page 69)

> Restores video playback state information and resumes playback.

Accessing the Audio Volume

These functions are used to adjust the playback audio volume, relative to the current system volume.

DVDIsMuted (page 46)

> Determines whether the video playback volume is currently muted.

DVDMute (page 50)

> Toggles the mute setting on or off.

DVDSetAudioVolume (page 58)

> Sets the playback audio volume setting (0 - 255).

DVDGetAudioVolume (page 24)

> Gets the current playback audio volume setting (0 - 255) .

DVDGetAudioVolumeInfo (page 25)

> Gets video playback volume information (minimum, maximum, and current volume).

Menu Navigation

These functions are used to access menus, navigate between menu buttons in response to keyboard or mouse events, and perform the actions that buttons represent.

DVDHasMenu (page 44)

> Determines whether a specified input menu is available.

`DVDIsOnMenu` (page 47)

    Determines whether a menu is being displayed, and if so, which one.

`DVDGoToMenu` (page 42)

    Jumps to a specified menu.

`DVDReturnToTitle` (page 55)

    Returns from a menu to the current position within the title.

`DVDGoBackOneLevel` (page 41)

    If a submenu is active, moves up one level in the same domain.

`DVDDoUserNavigation` (page 19)

    Allows the user to navigate between menu buttons.

`DVDDoButtonActivate` (page 18)

    Selects and activates a menu button by index.

`DVDGetButtoninfo` (page 26)

    Gets information about all available menu buttons in the current title, domain, or content.

`DVDGetButtonPosition` (page 27)

    Gets the position and default action of a menu button.

`DVDDoMenuClick` (page 99) Deprecated in Mac OS X v10.5

    If a menu button contains a specified point, the button's action is executed and the button's index is passed back. (Deprecated. Use `DVDDoMenuCGClick` instead.)

`DVDDoMenuMouseOver` (page 100) Deprecated in Mac OS X v10.5

    If a specified point lies inside a menu button, focuses the button and passes back its index. (Deprecated. Use `DVDDoMenuCGMouseOver` instead.)

Accessing Media Information

`DVDGetGPRMValue` (page 30)

    Returns the current value of a general parameter (GPRM) register.

`DVDGetMediaUniqueID` (page 31)

    Gets a unique identifier for a media folder.

`DVDGetMediaVolumeCFName` (page 32)

    Gets the volume name of the current media as a Core Foundation string.

Accessing Titles

`DVDSetTitle` (page 64)

    Sets the title to play.

`DVDGetTitle` (page 40)

    Gets the number of the current title.

`DVDGetNumTitles` (page 36)

    Gets the number of titles available on the media.

Accessing Chapters

These functions are used to access chapters in the current title.

DVDSetChapter (page 58)

> Sets the chapter to play.

DVDGetChapter (page 28)

> Gets the current chapter.

DVDGetNumChapters (page 35)

> Gets the number of chapters in the specified title.

DVDPreviousChapter (page 53)

> Plays the previous chapter in the current title.

DVDNextChapter (page 50)

> Plays the next chapter in the current title.

Accessing Camera Angles

DVDSetAngle (page 56)

> Sets the camera angle to display.

DVDGetAngle (page 20)

> Gets the camera angle displayed.

DVDGetNumAngles (page 34)

> Gets the number of camera angles currently available.

Controlling Subpictures

These functions are used to access subpictures in the current title.

DVDDisplaySubPicture (page 18)

> Turns the display of subpictures on or off.

DVDIsDisplayingSubPicture (page 46)

> Determines if subpictures are currently being displayed.

DVDSetSubPictureStream (page 62)

> Sets the current subpicture stream.

DVDGetSubPictureStream (page 39)

> Gets the current subpicture stream.

DVDGetNumSubPictureStreams (page 36)

> Gets the number of subpicture streams currently available.

Accessing Language Codes

These functions are used to access language codes that determine which subpicture, audio stream, or menu is used.

DVDSetDefaultSubPictureLanguageCode (page 60)

> Sets the default subpicture language and extension when subpictures are enabled.

DVDGetSubPictureLanguageCode (page 37)

> Gets the subpicture language code and extension for the current subpicture stream.

DVDGetSubPictureLanguageCodeByStream (page 38)
> Gets the subpicture language code and extension for a specified subpicture stream.

DVDSetDefaultAudioLanguageCode (page 59)
> Sets the default audio language code and extension.

DVDGetAudioLanguageCode (page 21)
> Gets the current audio language code and extension.

DVDGetAudioLanguageCodeByStream (page 22)
> Gets the audio language code and extension for a specified stream.

DVDSetDefaultMenuLanguageCode (page 59)
> Sets the default menu language code.

DVDGetMenuLanguageCode (page 33)
> Gets the menu language code for the current title.

Accessing Bookmarks

DVDGetBookmark (page 25)
> Requests a bookmark to the current play position.

DVDGotoBookmark (page 42)
> Sets the video playback position using a bookmark, and resumes play.

DVDGetLastPlayBookmark (page 30)
> Requests a bookmark to the last play position.

DVDSetLastPlayBookmark (page 62)
> Sets the video playback position using a bookmark.

DVDClearLastPlayBookmark (page 16)
> Sets the video playback position to the beginning of the disc.

Accessing Region Codes

These functions are used to access region codes on a DVD-Video disc or a DVD drive.

DVDGetDiscRegionCode (page 28)
> Gets the region codes available on a DVD-Video disc.

DVDGetDriveRegionCode (page 29)
> Gets the region code assigned to a DVD drive, and finds how many additional region code changes are permitted.

DVDSetDriveRegionCode (page 60)
> Sets the drive region code (requires user authentication).

Using Event Callbacks

DVDRegisterEventCallBack (page 54)
> Registers a callback function to get notified when state changes in DVD Playback occur.

DVDUnregisterEventCallBack (page 68)
> Unregisters an event callback.

`DVDIsRegisteredEventCallBack` (page 48)
>   Determines if an event callback is already registered.

`DVDSetTimeEventRate` (page 64)
>   Sets the rate of the DVD Playback time event.

`DVDGetTimeEventRate` (page 40)
>   Gets the rate of the time event.

Obsolete Functions

`DVDGetMediaVolumeName` (page 33)
>   Gets the volume name of the current media as a C string.

`DVDHasNextChapter` (page 44)
>   Indicates whether there is a chapter after the current chapter.

`DVDHasPreviousChapter` (page 44)
>   Indicates whether there is a chapter before the current chapter.

`DVDIdle` (page 45)
>   Not recommended.

`DVDSetAspectRatio` (page 57)
>   Sets the aspect ratio for the current title.

`DVDUpdateVideo` (page 69)
>   Not recommended.

`DVDGetVideoKeyColor` (page 101) Deprecated in Mac OS X v10.5
>   Gets the key color used by the video driver. (Deprecated. There is no replacement function.)

Functions

DVDClearLastPlayBookmark

Sets the video playback position to the beginning of the disc.

```
OSStatus DVDClearLastPlayBookmark (
    void
);
```

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
This function sets the video playback position to the beginning of the disc, but does not begin playing the media. Calling this function is equivalent to calling the function `DVDStop` (page 67) twice in succession. Before calling this function, media needs to be open.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`DVDPlayback.h`

DVDCloseMediaFile

Closes a previously opened VIDEO_TS media folder on a mass storage device.

```
OSStatus DVDCloseMediaFile (
    void
);
```

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
This function closes a VIDEO_TS media folder that was previously opened using the function `DVDOpenMediaFile` (page 51). If necessary, this function stops playback before closing the folder. You must call this function before attempting to open another media folder.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
`DVDPlayback.h`

DVDCloseMediaVolume

Closes a previously opened VIDEO_TS media folder on a DVD-Video disc.

```
OSStatus DVDCloseMediaVolume (
    void
);
```

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
This function closes a VIDEO_TS media folder that was previously opened using the function `DVDOpenMediaVolume` (page 52). If necessary, this function stops playback before closing the folder. You must call this function before attempting to open another media folder.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
`DVDPlayback.h`

DVDDisplaySubPicture

Turns the display of subpictures on or off.

```
OSStatus DVDDisplaySubPicture (
    Boolean inDisplay
);
```

Parameters

inDisplay

A Boolean value that specifies whether to display subpictures in the current title. To display subpictures, specify `TRUE`. To hide subpictures, specify `FALSE`.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
Before calling this function, DVD-Video media needs to be open and video playback started. You can use this function to display or hide additional bitmap graphic elements such as subtitles.

Availability
Available in Mac OS X v10.3 and later.

Declared In
DVDPlayback.h

DVDDispose

Ends the current DVD-Video playback session.

```
OSStatus DVDDispose (
    void
);
```

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
You call this function when you're finished using DVD Playback, usually when your application quits.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
DVDPlayback.h

DVDDoButtonActivate

Selects and activates a menu button by index.

```
OSStatus DVDDoButtonActivate (
    SInt32 inIndex
);
```

Parameters

inIndex

> The 1-based index of the menu button the user has selected.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

If the index coincides with an active button, this function executes the action corresponding to the button.

Availability

Available in Mac OS X v10.4 and later.

Declared In

DVDPlayback.h

DVDDoUserNavigation

Allows the user to navigate between menu buttons.

```
OSStatus DVDDoUserNavigation (
    DVDUserNavigation inNavigation
);
```

Parameters

inNavigation

> A user-navigation constant that specifies the direction the user is navigating in the menu. For a list of possible values, see "User Navigation Directions" (page 75).

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

You should call this function whenever the user navigates between buttons in the menu using the keyboard. This function moves the focus to the button. If you pass in kDVDUserNavigationEnter, the button action is executed.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

CocoaDVDPlayer

Declared In

DVDPlayback.h

DVDEnableWebAccess

Turns DVD@ccess support on or off.

```
OSStatus DVDEnableWebAccess (
    Boolean inEnable
);
```

Parameters

inEnable

To turn on support for DVD@ccess, specify `TRUE`. To turn off DVD@ccess support, specify `FALSE`.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

DVD@ccess is a feature in Apple's DVD Studio Pro that allows the DVD-Video author to embed external links to web-based resources and activate these links during video playback. Some users may find this feature to be intrusive, so DVD@ccess support is turned off by default.

Availability

Available in Mac OS X v10.3 and later.

Declared In

DVDPlayback.h

DVDGetAngle

Gets the camera angle displayed.

```
OSStatus DVDGetAngle (
    UInt16 *outAngleNum
);
```

Parameters

outAngleNum

A pointer to an integer. On return, the integer contains the 1-based index of the current camera angle.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

Before calling this function, DVD-Video media needs to be open and video playback started.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

CocoaDVDPlayer

Declared In

DVDPlayback.h

DVDGetAspectRatio

Gets the aspect ratio of the current title or menu.

```
OSStatus DVDGetAspectRatio (
    DVDAspectRatio *outRatio
);
```

Parameters

outRatio

> A pointer to a `DVDAspectRatio` variable. On return, the variable specifies the aspect ratio of the current title. For a list of possible aspect ratios, see "Aspect Ratios" (page 78).

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

The aspect ratio of the video can change whenever a different title or menu is displayed. You can use this function together with `DVDGetNativeVideoSize` (page 34) to calculate the bounds of the video area in a window.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

CocoaDVDPlayer

Declared In

`DVDPlayback.h`

DVDGetAudioLanguageCode

Gets the current audio language code and extension.

```
OSStatus DVDGetAudioLanguageCode (
    DVDLanguageCode *outCode,
    DVDAudioExtensionCode *outExtension
);
```

Parameters

outCode

> A pointer to a language code variable. On return, the variable contains the language code for the current audio stream. For a list of possible values, see "Language Codes" (page 81).

outExtension

> A pointer to an audio extension code variable. On return, the variable contains the audio extension code for the current audio stream. For a list of possible values, see "Audio Extension Codes" (page 84).

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Availability

Available in Mac OS X v10.3 and later.

Declared In

`DVDPlayback.h`

DVDGetAudioLanguageCodeByStream

Gets the audio language code and extension for a specified stream.

```
OSStatus DVDGetAudioLanguageCodeByStream (
    UInt16 inStreamNum,
    DVDLanguageCode *outCode,
    DVDAudioExtensionCode *outExtension
);
```

Parameters

inStreamNum

> An integer in the range 1 to 8 that specifies an audio stream in the current title.

outCode

> A pointer to a language code variable. On return, the variable contains the language code for the specified audio stream. For a list of possible values, see "Language Codes" (page 81).

outExtension

> A pointer to an audio extension code variable. On return, the variable contains the audio extension code for the specified audio stream. For a list of possible values, see "Audio Extension Codes" (page 84).

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Availability
Available in Mac OS X v10.3 and later.

Declared In
DVDPlayback.h

DVDGetAudioStream

Gets the audio stream currently being used.

```
OSStatus DVDGetAudioStream (
    UInt16 *outStreamNum
);
```

Parameters

outStreamNum

> A pointer to an integer. On return, the integer contains the number of the current audio stream, or 0 if the current title has no audio streams.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
Before calling this function, DVD-Video media needs to be open and video playback started.

Availability
Available in Mac OS X v10.3 and later.

Declared In
DVDPlayback.h

DVDGetAudioStreamFormat

Gets the current audio stream format.

```
OSStatus DVDGetAudioStreamFormat (
    DVDAudioFormat *outFormat,
    UInt32 *outBitsPerSample,
    UInt32 *outSamplesPerSecond,
    UInt32 *outChannels
);
```

Parameters

outFormat

> A pointer to an audio format variable. On return, the variable contains the audio format of the audio stream for the current title. For a list of possible formats, see "Audio Stream Formats" (page 80).

outBitsPerSample

> A pointer to an integer. On return, the integer contains the number of bits per sample in the current audio stream.

outSamplesPerSecond

> A pointer to an integer. On return, the integer contains the number of samples per second in the current audio stream.

outChannels

> A pointer to an integer. On return, the integer contains the number of audio channels in the current audio stream.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
Before calling this function, DVD-Video media needs to be open and video playback started.

Availability
Available in Mac OS X v10.3 and later.

Declared In
DVDPlayback.h

DVDGetAudioStreamFormatByStream

Gets the audio format for a specified stream.

```
OSStatus DVDGetAudioStreamFormatByStream (
    UInt32 inStreamNum,
    DVDAudioFormat *outFormat,
    UInt32 *outBitsPerSample,
    UInt32 *outSamplesPerSecond,
    UInt32 *outChannels
);
```

Parameters

inStreamNum

> An integer that specifies an audio stream in the current title.

outFormat

> A pointer to an audio format variable. On return, the variable contains the audio format of the audio stream for the current title. For a list of possible formats, see "Audio Stream Formats" (page 80).

outBitsPerSample

> A pointer to an integer. On return, the integer contains the number of bits per sample in the current audio stream.

outSamplesPerSecond

> A pointer to an integer. On return, the integer contains the number of samples per second in the current audio stream.

outChannels

> A pointer to an integer. On return, the integer contains the number of audio channels in the current audio stream.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

Before calling this function, DVD-Video media needs to be open.

Availability

Available in Mac OS X v10.4 and later.

Declared In

DVDPlayback.h

DVDGetAudioVolume

Gets the current playback audio volume setting (0 - 255) .

```
OSStatus DVDGetAudioVolume (
    UInt16 *outVolume
);
```

Parameters

outVolume

> A pointer to an integer. On return, the integer contains the current playback volume in the range 0 to 255.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function passes back the current playback audio volume setting. For more information, see DVDSetAudioVolume (page 58).

Availability

Available in Mac OS X v10.3 and later.

Declared In

DVDPlayback.h

DVDGetAudioVolumeInfo

Gets video playback volume information (minimum, maximum, and current volume).

```
OSStatus DVDGetAudioVolumeInfo (
    UInt16 *outMinVolume,
    UInt16 *outCurVolume,
    UInt16 *outMaxVolume
);
```

Parameters

outMinVolume

> A pointer to an integer. On return, the integer contains the minimum video playback volume. This parameter is optional; you may specify NULL.

outCurVolume

> A pointer to an integer. On return, the integer contains the current video playback volume. This parameter is optional; you may specify NULL.

outMaxVolume

> A pointer to an integer. On return, the integer contains the maximum video playback volume. This parameter is optional; you may specify NULL.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

Currently the minimum and maximum volumes are always 0 and 255.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

CocoaDVDPlayer

Declared In

DVDPlayback.h

DVDGetBookmark

Requests a bookmark to the current play position.

```
OSStatus DVDGetBookmark (
    void *outBookMarkData,
    UInt32 *ioBookMarkDataSize
);
```

Parameters

outBookmarkData

> A generic pointer to memory for a bookmark in your calling program, or NULL. On return, this memory contains the new bookmark.
>
> The size of this memory should be equal to or greater than the actual size of a bookmark in bytes. To determine the actual size of a bookmark, see the Discussion below.

ioBookmarkDataSize

On entry, a pointer to an integer that contains the size of your bookmark memory. If you haven't allocated memory for a bookmark, set this size to zero. On return, the integer's value is the actual size of a new bookmark. See the Discussion below.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function is used when playing media on a DVD-Video disc. Before calling this function, the media needs to be open and playing. This function passes back a bookmark to the current play position in the current DVD-Video playback session.

To request a play bookmark, you need to call this function twice. The first call determines the minimum required size for the bookmark:

```
UInt32 size = 0;
DVDGetBookmark (NULL, &size);
```

After you allocate sufficient memory for the bookmark, call this function again to create the bookmark, passing it the location and size of your bookmark memory:

```
void *bookmark = malloc (size);
OSStatus err = DVDGetBookmark (bookmark, &size);
```

For information about requesting a bookmark when the media is not playing, see `DVDGetLastPlayBookmark` (page 30).

Availability

Available in Mac OS X v10.4 and later.

Related Sample Code

CocoaDVDPlayer

Declared In

DVDPlayback.h

DVDGetButtoninfo

Gets information about all available menu buttons in the current title, domain, or content.

```
OSStatus DVDGetButtoninfo (
    UInt32 *numberOfButtons,
    UInt32 *selectedButton,
    UInt32 *forcedActivateButton,
    UInt32 *userButtonOffset,
    UInt32 *numberOfUserButtons
);
```

Parameters

numberOfButtons

A pointer to an integer. On return, the integer contains the number of buttons in the current menu.

selectedButton

A pointer to an integer. On return, the integer contains the 1-based index of the selected button.

forcedActivateButton

>A pointer to an integer. On return, the integer contains the index of the button whose action is performed when a specified period of time elapses after the menu is first displayed.

userButtonOffset

>A pointer to an integer. On return, the integer contains the index of the first user-selectable button. If the number of user-selectable buttons in a menu is smaller than the total number of buttons, this index may be greater than zero.

numberOfUserButtons

>A pointer to an integer. On return, the integer contains the number of user-selectable buttons in the current menu.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Availability

Available in Mac OS X v10.4 and later.

Declared In

DVDPlayback.h

DVDGetButtonPosition

Gets the position and default action of a menu button.

```
OSStatus DVDGetButtonPosition (
    UInt32 index,
    CGRect *outRect,
    UInt32 *autoAction
);
```

Parameters

index

>The zero-based index of a menu button. If your button index is 1-based, you should decrement the index before you pass it to this function.

outRect

>A pointer to a Quartz rectangle. On return, the rectangle contains the position and dimensions of the specified button in window local coordinates.

autoAction

>A pointer to an integer flag. On return, a value of 1 indicates the button is a forced activate button—that is, the button's action is executed immediately when the button is selected. A value of 0 indicates the button is not a forced activate button.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Availability

Available in Mac OS X v10.4 and later.

Declared In

DVDPlayback.h

DVDGetChapter

Gets the current chapter.

```
OSStatus DVDGetChapter (
    UInt16 *outChapterNum
);
```

Parameters

outChapterNum

A pointer to an integer. On return, the integer contains the current chapter number.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Availability

Available in Mac OS X v10.3 and later.

Declared In

DVDPlayback.h

DVDGetDiscRegionCode

Gets the region codes available on a DVD-Video disc.

```
OSStatus DVDGetDiscRegionCode (
    DVDRegionCode *outCode
);
```

Parameters

outCode

A pointer to a DVDRegionCode variable. On return, the variable contains a bitfield that specifies one or more region codes. For more information about region codes, see "Region Codes" (page 87).

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function passes back the region codes assigned to the DVD-Video disc that's currently in use. Before calling this function, you need to call the function DVDOpenMediaVolume (page 52) to open the media folder on the disc.

To test whether a disc is authorized for playback in region *n*, you need to compute the bitwise AND of outCode with region code *n*. If the result is equal to region code *n*, the disc is authorized for playback in region *n*. For example, this source code shows how to test whether a disc is authorized for playback in region 1:

```
const DVDRegionCode myRegionCode1 = kDVDRegionCode1;
DVDRegionCode outCode;
DVDGetDiscRegionCode (&outCode);
if ((~outCode & ~myRegionCode1) == ~myRegionCode1) {
    // media supports region 1
}
```

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
`DVDPlayback.h`

DVDGetDriveRegionCode

Gets the region code assigned to a DVD drive, and finds how many additional region code changes are permitted.

```
OSStatus DVDGetDriveRegionCode (
    DVDRegionCode *outCode,
    SInt16 *outNumberChangesLeft
);
```

Parameters

outCode

> A pointer to a `DVDRegionCode` variable. On return, the variable specifies the region code that's assigned to the DVD drive currently in use. If the drive was never initialized, the region code is `kDVDRegionCodeUninitialized`. For more information about region codes, see "Region Codes" (page 87).

outNumberChangesLeft

> A pointer to an integer. On return, the integer contains the number of region code changes remaining for this drive.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
This function passes back the region code assigned to the DVD drive that's currently in use. Before calling this function, you need to call the function `DVDOpenMediaVolume` (page 52) to open the media folder on a disc that's mounted in the drive.

To test whether region *n* is assigned to a drive, you need to compute the bitwise `AND` of `outCode` with region code *n*. If the result is equal to region code *n*, the drive can play discs that are authorized for playback in region *n*. For example, this source code shows how to test whether region 1 is assigned to a drive:

```
const DVDRegionCode myRegionCode1 = kDVDRegionCode1;
DVDRegionCode outCode;
SInt16 outNumberChangesLeft;
DVDGetDriveRegionCode (&outCode, &outNumberChangesLeft);
if ((~outCode & ~myRegionCode1) == ~myRegionCode1) {
    // media supports region 1
}
```

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
`DVDPlayback.h`

DVDGetFormatStandard

Gets the video broadcast format of current DVD-Video media.

```
OSStatus DVDGetFormatStandard (
    DVDFormat *outFormat
);
```

Parameters

outFormat

      A pointer to a `DVDFormat` variable. On return, the variable contains the video broadcast format of the current media. For a list of possible values, see " Video Broadcast Formats" (page 79).

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function determines which video broadcast format (NTSC or PAL) is used in the current media. Before calling this function, DVD-Video media needs to be open.

Availability

Available in Mac OS X v10.3 and later.

Declared In

DVDPlayback.h

DVDGetGPRMValue

Returns the current value of a general parameter (GPRM) register.

```
OSStatus DVDGetGPRMValue (
    UInt32 index,
    UInt32 *value
);
```

Parameters

index

      An integer index that specifies the desired GPRM register. The valid range is 1 to 16.

value

      A pointer to an integer. On return, the integer contains the current value of the specified GPRM register.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Availability

Available in Mac OS X v10.4 and later.

Declared In

DVDPlayback.h

DVDGetLastPlayBookmark

Requests a bookmark to the last play position.

```
OSStatus DVDGetLastPlayBookmark (
   void *outBookMarkData,
   UInt32 *ioBookMarkDataSize
);
```

Parameters

outBookmarkData

A generic pointer to memory for a bookmark in your calling program, or `NULL`. On return, this memory contains the last bookmark after the first Stop command.

The size of this memory should be equal to or greater than the actual size of a bookmark in bytes. To determine the actual size of a bookmark, see the discussion below.

ioBookmarkDataSize

On entry, a pointer to an integer that contains the size of your bookmark memory. If you haven't allocated memory for a bookmark, set this size to zero. On return, the integer's value is the actual size of a new bookmark. See the Discussion below.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function is used when playing media on a DVD-Video disc. Before calling this function, the media needs to be open.

This function passes back a bookmark to the last play position in the current DVD-Video playback session. A last play bookmark represents the exact stop position after calling the function `DVDStop` (page 67).

To request the last play bookmark, you need to call this function twice. The first call determines the minimum required size for the bookmark:

```
UInt32 size = 0;
DVDGetBookmark (NULL, &size);
```

After you allocate sufficient memory for the bookmark, call this function again to create the bookmark, passing it the location and size of your bookmark memory:

```
void *bookmark = malloc (size);
OSStatus err = DVDGetBookmark (bookmark, &size);
```

For information about requesting a bookmark when the media is playing, see `DVDGetBookmark` (page 25).

Availability

Available in Mac OS X v10.4 and later.

Declared In

`DVDPlayback.h`

DVDGetMediaUniqueID

Gets a unique identifier for a media folder.

```
OSStatus DVDGetMediaUniqueID (
    DVDDiscID outDiscID
);
```

Parameters

outDiscID

An 8-byte array of type `DVDDiscID`. On return, this array contains the unique identifier of the current media.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function passes back a 64-bit identifier that can help you distinguish between different media folders. While the identifiers generated by this function are not guaranteed to be unique (see below), duplicate identifiers are extremely rare.

There are two known limitations of this function:

- Two copies of the same DVD that are produced at different times do not necessarily have the same identifier. This can happen when the two media folders have minor differences.

- Two different DVDs with similar content—collections of episodes from a television series, for example—do not always have different identifiers. This can happen when the two media folders are identical.

Availability

Available in Mac OS X v10.4 and later.

Related Sample Code

CocoaDVDPlayer

Declared In

DVDPlayback.h

DVDGetMediaVolumeCFName

Gets the volume name of the current media as a Core Foundation string.

```
OSStatus DVDGetMediaVolumeCFName (
    CFStringRef *outDiscVolumeCFName
);
```

Parameters

outDiscVolumeCFName

On return, a pointer to the volume name of the current media. If you need to save this Core Foundation string for later use, you should retain it.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function passes back the media volume name as a Core Foundation string. This is the name seen on the desktop when Mac OS X mounts a DVD-Video disc. For information about using Core Foundation strings, see *CFString Reference*.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`DVDPlayback.h`

DVDGetMediaVolumeName

Gets the volume name of the current media as a C string.

```
OSStatus DVDGetMediaVolumeName (
    char **outDiscVolumeName
);
```

Parameters

`outDiscVolumeName`

A pointer to a C string. On return, the C string is the volume name of the current media. You are not responsible for allocating a buffer for the string.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
This function passes back the volume name of the current media as a C string. This is the name seen on the desktop when Mac OS X mounts a DVD-Video disc.

Special Considerations

This function has a serious limitation: it does not support volume names with multiple byte characters. For this reason, applications running in Mac OS X version 10.4 or later should use the replacement function `DVDGetMediaVolumeCFName` (page 32) instead.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`DVDPlayback.h`

DVDGetMenuLanguageCode

Gets the menu language code for the current title.

```
OSStatus DVDGetMenuLanguageCode (
    DVDLanguageCode *outCode
);
```

Parameters

`outCode`

A pointer to a language code variable. On return, the variable contains the menu language code for the current title. For a list of possible values, see "Language Codes" (page 81).

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Availability
Available in Mac OS X v10.3 and later.

Declared In
DVDPlayback.h

DVDGetNativeVideoSize

Gets the native width and height in pixels of the current title.

```
OSStatus DVDGetNativeVideoSize (
    UInt16 *outWidth,
    UInt16 *outHeight
);
```

Parameters

outWidth
> A pointer to an integer. On return, the integer contains the native width of the current title.

outHeight
> A pointer to an integer. On return, the integer contains the native height of the current title.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
The native dimensions of a title are specified by the author of the DVD-Video media. The width and height are typically 720 x 480 pixels for the NTSC video format and 720 x 576 pixels for the PAL video format.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
DVDPlayback.h

DVDGetNumAngles

Gets the number of camera angles currently available.

```
OSStatus DVDGetNumAngles (
    UInt16 *outNumAngles
);
```

Parameters

outNumAngles
> On return, a pointer to the number of different camera angles in the current title. DVD-Video media supports up to 9 camera angles.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
`DVDPlayback.h`

DVDGetNumAudioStreams

Gets the number of audio streams currently available.

```
OSStatus DVDGetNumAudioStreams (
    UInt16 *outNumStreams
);
```

Parameters

outNumStreams

> A pointer to an integer. On return, the integer contains the number of audio streams available in the current title.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
Before calling this function, DVD-Video media needs to be open.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`DVDPlayback.h`

DVDGetNumChapters

Gets the number of chapters in the specified title.

```
OSStatus DVDGetNumChapters (
    UInt16 inTitleNum,
    UInt16 *outNumChapters
);
```

Parameters

inTitleNum

> An integer in the range 1 to 99 that specifies a title on the current media.

outNumChapters

> A pointer to an integer. On return, the integer contains the number of chapters in the specified title.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
If the specified chapter does not exist, this function does nothing and returns an error.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`DVDPlayback.h`

DVDGetNumSubPictureStreams

Gets the number of subpicture streams currently available.

```
OSStatus DVDGetNumSubPictureStreams (
    UInt16 *outNumStreams
);
```

Parameters

outNumStreams

On return, a pointer to the number of subpicture streams available in the current title. DVD-Video media supports up to 32 subpicture streams per title.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

Before calling this function, DVD-Video media needs to be open and video playback started.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`DVDPlayback.h`

DVDGetNumTitles

Gets the number of titles available on the media.

```
OSStatus DVDGetNumTitles (
    UInt16 *outNumTitles
);
```

Parameters

outNumTitles

A pointer to an integer. On return, the integer contains the number of titles on the current media in the range 0 to 99.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Availability

Available in Mac OS X v10.3 and later.

Declared In

`DVDPlayback.h`

DVDGetScanRate

Gets the current scan direction and scan rate.

```
OSStatus DVDGetScanRate (
    DVDScanRate *outRate,
    DVDScanDirection *outDirection
);
```

Parameters

outRate

> A pointer to a `DVDScanRate` variable. On return, the variable contains the current scan rate. For a list of possible values, see "Scan Rates" (page 78).

outDirection

> A pointer to a `DVDScanDirection` variable. On return, the variable contains the current scan direction. For a list of possible values, see "Scan Directions" (page 77).

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Availability
Available in Mac OS X v10.3 and later.

Declared In
`DVDPlayback.h`

DVDGetState

Gets the state of the current DVD-Video playback session.

```
OSStatus DVDGetState (
    DVDState *outState
);
```

Parameters

outState

> A pointer to a `DVDState` variable. On return, the variable contains the state of the current playback session. For a list of possible values, see "Playback States" (page 73).

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Availability
Available in Mac OS X v10.3 and later.

Declared In
`DVDPlayback.h`

DVDGetSubPictureLanguageCode

Gets the subpicture language code and extension for the current subpicture stream.

```
OSStatus DVDGetSubPictureLanguageCode (
    DVDLanguageCode *outCode,
    DVDSubpictureExtensionCode *outExtension
);
```

Parameters

outCode

> A pointer to a language code variable. On return, the variable contains the language code for the current subpicture. For a list of possible values, see "Language Codes" (page 81).

outExtension

> A pointer to a subpicture extension variable. On return, the variable contains the extension code for the current subpicture. For a list of possible values, see "Subpicture Extension Codes" (page 85).

Return Value

A result code. If subpictures are not available, this function does nothing and returns `noErr`. For a list of other possible result codes, see "DVD Playback Result Codes" (page 95).

Availability

Available in Mac OS X v10.3 and later.

Declared In

DVDPlayback.h

DVDGetSubPictureLanguageCodeByStream

Gets the subpicture language code and extension for a specified subpicture stream.

```
OSStatus DVDGetSubPictureLanguageCodeByStream (
    UInt16 inStreamNum,
    DVDLanguageCode *outCode,
    DVDSubpictureExtensionCode *outExtension
);
```

Parameters

inStreamNum

> A subpicture stream number in the current title. Subpicture stream numbers range from 1 to 32.

outCode

> A pointer to a language code variable. On return, the variable contains the language code for the subpicture stream. For a list of possible values, see "Language Codes" (page 81).

outExtension

> A pointer to a subpicture extension variable. On return, the variable contains the extension code for the subpicture stream. For a list of possible values, see "Subpicture Extension Codes" (page 85).

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

If subpictures are not available, this function does nothing and returns a non-zero result code.

Availability

Available in Mac OS X v10.3 and later.

Declared In

DVDPlayback.h

DVDGetSubPictureStream

Gets the current subpicture stream.

```
OSStatus DVDGetSubPictureStream (
    UInt16 *outStreamNum
);
```

Parameters

outStreamNum

> A pointer to an integer. On return, the integer contains the identifier of the current subpicture stream, or 0 if the current title has no subpictures. DVD-Video media supports up to 32 subpicture streams per title.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
Before calling this function, DVD-Video media needs to be open and video playback started.

Availability
Available in Mac OS X v10.3 and later.

Declared In
DVDPlayback.h

DVDGetTime

Gets the current video playback position in the current title or menu, relative to the specified time code (elapsed, remaining).

```
OSStatus DVDGetTime (
    DVDTimeCode inTimeCode,
    DVDTimePosition *outTime,
    UInt16 *outFrames
);
```

Parameters

inTimeCode

> A time code constant. You must specify one of two values: kDVDTimeCodeElapsedSeconds or kDVDTimeCodeRemainingSeconds.

outTime

> A pointer to an integer. On return, the integer contains the current video playback position in seconds.

outFrames

> A pointer to an integer. On return, the integer contains the current frame number.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
Before calling this function, DVD-Video media needs to be open and video playback started. The video playback position in a menu refers to animated menus.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`DVDPlayback.h`

DVDGetTimeEventRate

Gets the rate of the time event.

```
OSStatus DVDGetTimeEventRate (
   UInt32 *outMilliseconds
);
```

Parameters

outMilliseconds

A pointer to an integer variable. On return, the integer contains the current interval between time events in milliseconds.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Availability
Available in Mac OS X v10.3 and later.

Declared In
`DVDPlayback.h`

DVDGetTitle

Gets the number of the current title.

```
OSStatus DVDGetTitle (
   UInt16 *outTitleNum
);
```

Parameters

outTitleNum

A pointer to an integer. On return, the integer contains the current title number, or 0 if a menu is active. DVD-Video media can contain up to 99 titles.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
(discussion forthcoming)

Availability
Available in Mac OS X v10.3 and later.

Declared In
`DVDPlayback.h`

DVDGetVideoDisplay

Gets the graphics display for the current DVD-Video playback session.

```
OSStatus DVDGetVideoDisplay (
   CGDirectDisplayID *outDisplay
);
```

Parameters

outDisplay

A pointer to a Quartz display ID. On return, the display ID is set to the current video playback display.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

Before calling this function, you need to set the video playback display using `DVDSetVideoDisplay` (page 65) or `DVDSwitchToDisplay` (page 68).

Availability

Available in Mac OS X v10.3 and later.

Declared In

`DVDPlayback.h`

DVDGetVideoWindowID

Gets the window ID of the Cocoa window in use for DVD-Video playback.

```
OSStatus DVDGetVideoWindowID (
   UInt32 *outVidWindowID
);
```

Parameters

outVidWindowID

A pointer to a Cocoa window ID. On return, the window ID of the Cocoa window currently in use for DVD-Video playback, or zero if there is no Cocoa window.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Availability

Available in Mac OS X v10.3 and later.

Declared In

`DVDPlayback.h`

DVDGoBackOneLevel

If a submenu is active, moves up one level in the same domain.

```
OSStatus DVDGoBackOneLevel (
    void
);
```

Return Value

A result code. If no submenu is active, this function does nothing and returns . For other possible result codes, see "DVD Playback Result Codes" (page 95).

Discussion

This function is used to navigate one level up in a hierarchical structure in the same domain—for example, from a scene selection menu back to the main menu. This action is comparable to navigating upwards in the directory or folder hierarchy of a file system.

Availability

Available in Mac OS X v10.3 and later.

Declared In

DVDPlayback.h

DVDGotoBookmark

Sets the video playback position using a bookmark, and resumes play.

```
OSStatus DVDGotoBookmark (
    void *inBookMarkData,
    UInt32 inBookMarkDataSize
);
```

Parameters

inBookmarkData

> A generic pointer to memory for a bookmark. To learn how to obtain a bookmark, see DVDGetBookmark (page 25) or DVDGetLastPlayBookmark (page 30).

inBookmarkDataSize

> The actual size of the specified bookmark in bytes.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function is used when playing media on a DVD-Video disc. Before calling this function, the media needs to be open. This function uses the specified bookmark to set the video playback position, and begins playing the media. See also DVDSetLastPlayBookmark (page 62).

Availability

Available in Mac OS X v10.4 and later.

Related Sample Code

CocoaDVDPlayer

Declared In

DVDPlayback.h

DVDGoToMenu

Jumps to a specified menu.

```
OSStatus DVDGoToMenu (
    DVDMenu inMenu
);
```

Parameters

inMenu

> A constant that specifies the desired menu. For a list of possible values, see "Playback Menus" (page 74).

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

Before calling this function, DVD-Video media needs to be open and video playback started. Not every DVD has every menu type. If you attempt to jump to a non-existent menu, this function returns an error.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

CocoaDVDPlayer

Declared In

DVDPlayback.h

DVDHasMedia

Determines whether DVD-Video media is open for playback.

```
OSStatus DVDHasMedia (
    Boolean *outHasMedia
);
```

Parameters

outHasMedia

> A pointer to a Boolean variable. On return, a value of TRUE indicates that DVD-Video media is open for playback. A value of FALSE indicates no media is currently open.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function checks to see if DVD-Video media is currently open as a result of successfully calling DVDOpenMediaFile (page 51) or DVDOpenMediaVolume (page 52).

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

CocoaDVDPlayer

Declared In

DVDPlayback.h

DVDHasMenu

Determines whether a specified input menu is available.

```
OSStatus DVDHasMenu (
    DVDMenu inMenu,
    Boolean *outHasMenu
);
```

Parameters

inMenu

A constant that specifies an input menu. For a list of possible values, see "Playback Menus" (page 74).

outHasMenu

A pointer to a Boolean variable. On return, a value of `TRUE` indicates the specified input menu is available. A value of `FALSE` indicates the menu is not available.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

Before calling this function, DVD-Video media needs to be open.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`DVDPlayback.h`

DVDHasNextChapter

Indicates whether there is a chapter after the current chapter.

```
OSStatus DVDHasNextChapter (
    Boolean *outHasChapter
);
```

Parameters

outHasChapter

A pointer to a Boolean variable. On return, its value is `TRUE`.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function does nothing and passes back `TRUE`.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`DVDPlayback.h`

DVDHasPreviousChapter

Indicates whether there is a chapter before the current chapter.

```
OSStatus DVDHasPreviousChapter (
    Boolean *outHasChapter
);
```

Parameters

outHasChapter

A pointer to a Boolean variable. On return, its value is `TRUE`.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function does nothing and passes back `TRUE`.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`DVDPlayback.h`

DVDIdle

Not recommended.

```
OSStatus DVDIdle (
    void
);
```

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function does nothing and returns `noErr`.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`DVDPlayback.h`

DVDInitialize

Starts a new DVD-Video playback session.

```
OSStatus DVDInitialize (
    void
);
```

Return Value

A result code. If a new DVD-Video playback session is successfully initiated, this function returns `noErr`. If a session already exists, this function returns `kDVDErrorInitializingLib`. For other possible result codes, see "DVD Playback Result Codes" (page 95).

Discussion

When preparing to play a DVD, you must call this function to initialize a new DVD-Video playback session. Only one process at a time can use DVD Playback. When you are finished using DVD Playback, you should call the function `DVDDispose` (page 18) to end the session.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

CocoaDVDPlayer

Declared In

`DVDPlayback.h`

DVDIsDisplayingSubPicture

Determines if subpictures are currently being displayed.

```
OSStatus DVDIsDisplayingSubPicture (
    Boolean *outDisplayingSubPicture
);
```

Parameters

outDisplayingSubPicture

>A pointer to a Boolean variable. On return, a value of `TRUE` indicates subpictures are being displayed. A value of `FALSE` indicates subpictures are hidden.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

Before calling this function, DVD-Video media needs to be open and video playback started.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`DVDPlayback.h`

DVDIsMuted

Determines whether the video playback volume is currently muted.

```
OSStatus DVDIsMuted (
    Boolean *outIsMuted
);
```

Parameters

outIsMuted

>A pointer to a Boolean variable. On return, a value of `TRUE` indicates the video playback volume is currently muted. A value of `FALSE` indicates the volume is not muted.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
`DVDPlayback.h`

DVDIsOnMenu

Determines whether a menu is being displayed, and if so, which one.

```
OSStatus DVDIsOnMenu (
    Boolean *outOnMenu,
    DVDMenu *outMenu
);
```

Parameters

outOnMenu

A pointer to a `Boolean` variable. On return, a value of `TRUE` indicates a menu with one or more buttons is displayed. A value of `FALSE` indicates no menu is being displayed.

outMenu

A pointer to a menu code. On return, the code's value is the current menu or `kDVDMenuNone`. For a list of all possible values, see "Playback Menus" (page 74).

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
Before calling this function, DVD-Video media needs to be open and video playback started. There can be only one menu displayed at a time.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
`DVDPlayback.h`

DVDIsPaused

Determines if media is open and playing, and video playback is paused.

```
OSStatus DVDIsPaused (
    Boolean *outIsPaused
);
```

Parameters

outIsPaused

A pointer to a Boolean variable. On return, a value of `TRUE` indicates that media is open and paused. A value of `FALSE` indicates that media is open and not paused, or not open.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Availability

Available in Mac OS X v10.3 and later.

Declared In

`DVDPlayback.h`

DVDIsPlaying

Determines if media is open and playing (even if paused).

```
OSStatus DVDIsPlaying (
    Boolean *outIsPlaying
);
```

Parameters

outIsPlaying

A pointer to a Boolean variable. On return, a value of `TRUE` indicates that media is open and playing, even if paused. A value of `FALSE` indicates that media is open and stopped, or not open.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Availability

Available in Mac OS X v10.3 and later.

Declared In

`DVDPlayback.h`

DVDIsRegisteredEventCallBack

Determines if an event callback is already registered.

```
Boolean DVDIsRegisteredEventCallBack (
    DVDEventCallBackRef inCallBackID
);
```

Parameters

inCallBackID

The unique identifier for an event callback registration. See `DVDRegisterEventCallBack` (page 54).

Return Value

A Boolean value that indicates whether the corresponding event callback has been registered.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`DVDPlayback.h`

DVDIsSupportedDisplay

Determines whether a graphics display supports DVD-Video playback.

```
OSStatus DVDIsSupportedDisplay (
    CGDirectDisplayID inDisplay,
    Boolean *outSupported
);
```

Parameters

inDisplay

> The Quartz display ID for the graphics display you want to use for video playback. For information about how to get a display ID, see *Quartz Display Services Reference*.

outSupported

> A pointer to a Boolean variable. On return, a value of `TRUE` indicates that the specified graphics display can be used for DVD-Video playback. A value of `FALSE` indicates the display cannot be used for this purpose. If the value is `FALSE`, the display may require a different video driver than the one currently in use.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
For this function to work properly, you must first call the function `DVDSetVideoWindowID` (page 65) with a valid window.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`DVDPlayback.h`

DVDIsValidMediaRef

Determines whether a VIDEO_TS media folder has a valid layout.

```
OSStatus DVDIsValidMediaRef (
    FSRef *inRef,
    Boolean *outIsValid
);
```

Parameters

inFile

> A pointer to a VIDEO_TS media folder on a DVD-Video disc or another mass storage device. To convert a path into an `FSRef` structure, use the function `FSPathMakeRef`.

outIsValid

> A pointer to a Boolean variable. On return, a value of `TRUE` indicates the folder layout is valid. A value of `FALSE` indicates the layout is not valid and the folder cannot be opened for playback.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

Before attempting to open a DVD-Video media folder, you should call this function to verify that the folder is valid. If another media folder is open, before calling this function you should call `DVDCloseMediaFile` (page 17) or `DVDCloseMediaVolume` (page 17) to close the other folder.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

CocoaDVDPlayer

Declared In

DVDPlayback.h

DVDMute

Toggles the mute setting on or off.

```
OSStatus DVDMute (
    Boolean inMute
);
```

Parameters

inMute

> A Boolean value that specifies whether to turn the mute setting on or off. To mute the video playback volume, pass `TRUE`. To restore the previous volume setting, pass `FALSE`.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

CocoaDVDPlayer

Declared In

DVDPlayback.h

DVDNextChapter

Plays the next chapter in the current title.

```
OSStatus DVDNextChapter (
    void
);
```

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
Before calling this function, DVD-Video media needs to be open and video playback started. This function finds and begins to play the next chapter. If there is no next chapter, this function continues playing the current chapter.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
DVDPlayback.h

DVDOpenMediaFile

Opens a VIDEO_TS media folder located on a mass storage device.

```
OSStatus DVDOpenMediaFile (
    FSRef *inFile
);
```

Parameters
inFile

A pointer to a VIDEO_TS media folder on a mass storage device other than a DVD-Video disc. To convert a path into an FSRef structure, use the function FSPathMakeRef.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
This function opens a VIDEO_TS media folder on a mass storage device such as a hard drive. To open the media folder on a DVD-Video disc, you should use the function DVDOpenMediaVolume (page 52) instead.

Before calling this function:

■ If another media folder is open, you should close the other folder by calling DVDCloseMediaFile (page 17) or DVDCloseMediaVolume (page 17).

■ You should call the function DVDIsValidMediaRef (page 49) to verify that this media folder is valid.

After you are finished playing the DVD-Video media, you should call the function DVDCloseMediaFile (page 17) to close the folder.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
`DVDPlayback.h`

DVDOpenMediaVolume

Opens a VIDEO_TS media folder located on a DVD-Video disc.

```
OSStatus DVDOpenMediaVolume (
    FSRef *inVolume
);
```

Parameters

inVolume

> A pointer to the VIDEO_TS media folder located at the root level of a DVD-Video disc. To convert a path into an `FSRef` structure, use the function `FSPathMakeRef`.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
This function opens the VIDEO_TS media folder on a DVD-Video disc. To open a media folder on a mass storage device such as a hard drive, you should use the function `DVDOpenMediaFile` (page 51) instead.

Before calling this function:

■ If another media folder is open, you should close the other folder by calling `DVDCloseMediaFile` (page 17) or `DVDCloseMediaVolume` (page 17).

■ You should call the function `DVDIsValidMediaRef` (page 49) to verify that this media folder is valid.

After you are finished playing the media on the disc, you should call the function `DVDCloseMediaVolume` (page 17) to close the folder.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
`DVDPlayback.h`

DVDPause

Pauses the media, if currently playing.

```
OSStatus DVDPause (
    void
);
```

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
This function pauses the video playback and freezes the video. Before calling this function, DVD-Video media needs to be open and video playback started. After calling this function, the media is still considered to be in video playback mode—that is, you may call DVDStop (page 67) without first calling DVDResume (page 55).

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
DVDPlayback.h

DVDPlay

Starts playing the media, if currently stopped.

```
OSStatus DVDPlay
```

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
Before calling this function, DVD-Video media needs to be open. If you call this function after calling DVDStop (page 67), video playback resumes at the stop position.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
DVDPlayback.h

DVDPreviousChapter

Plays the previous chapter in the current title.

```
OSStatus DVDPreviousChapter (
    void
);
```

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion

Before calling this function, DVD-Video media needs to be open and video playback started. This function finds and begins to play the previous chapter. If there is no previous chapter, this function rewinds to the beginning of the current chapter and begins playing.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

CocoaDVDPlayer

Declared In

DVDPlayback.h

DVDRegisterEventCallBack

Registers a callback function to get notified when state changes in DVD Playback occur.

```
OSStatus DVDRegisterEventCallBack (
    DVDEventCallBackFunctionPtr inCallBackProc,
    DVDEventCode *inCode,
    UInt32 inCodeCount,
    void *inRefCon,
    DVDEventCallBackRef *outCallBackID
);
```

Parameters

inCallBackProc

>   A callback function that handles one or more DVD Playback events.

inCode

>   An array of one or more events for which you are registering this callback. For a list of possible events, see "Event Codes" (page 93).

inCodeCount

>   The number of events for which you are registering.

inRefCon

>   Application-defined data that's passed to your callback. If you decide not to use this parameter, you should pass NULL.

outCallBackID

>   A pointer to an integer. On return, the integer contains the unique identifier for this registration.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function is optional. You may register more than one callback, and you may register multiple events for a single callback. For every call to this function, there should be a corresponding call to the function DVDUnregisterEventCallBack (page 68) using the identifier for this registration. You should unregister your callbacks before calling the function DVDDispose (page 18) to end the current DVD-Video playback session.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
`DVDPlayback.h`

DVDResume

Resumes playing the media, if currently paused.

```
OSStatus DVDResume (
    void
);
```

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Availability
Available in Mac OS X v10.3 and later.

Declared In
`DVDPlayback.h`

DVDReturnToTitle

Returns from a menu to the current position within the title.

```
OSStatus DVDReturnToTitle (
    void
);
```

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
This function returns to the last played position in the current title.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
`DVDPlayback.h`

DVDScan

Fast forwards or rewinds at the specified scan rate.

```
OSStatus DVDScan (
    DVDScanRate inRate,
    DVDScanDirection inDirection
);
```

Parameters

inRate

> An integer that specifies the rate at which to scan. For a list of possible values, see "Scan Rates" (page 78).

inDirection

> An integer that specifies the direction in which to scan. For a list of possible values, see "Scan Directions" (page 77).

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
This function advances the media in the specified direction at the specified rate. When you scan at a rate other than `kDVDScanRate1x` (normal speed), the audio channel is muted and subtitles are not displayed. If the scan rate is normal and the direction is forward, this function is equivalent to `DVDPlay` (page 53). This function does not support video playback at normal speed in the reverse direction.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
`DVDPlayback.h`

DVDSetAngle

Sets the camera angle to display.

```
OSStatus DVDSetAngle (
    UInt16 inAngleNum
);
```

Parameters

inAngleNum

> A 1-based index that specifies the camera angle to display. The angle number of the main angle is 1. DVD-Video media supports up to nine camera angles.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
Before calling this function, DVD-Video media needs to be open and video playback started. The DVD-Video author decides how many camera angles are used. If the specified angle does not exist, this function does nothing.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
`DVDPlayback.h`

DVDSetAspectRatio

Sets the aspect ratio for the current title.

```
OSStatus DVDSetAspectRatio (
    DVDAspectRatio inRatio
);
```

Parameters

inRatio

A constant that specifies the desired aspect ratio for the current title. For a list of possible values, see "Aspect Ratios" (page 78).

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function does nothing and returns `noErr`.

Availability

Available in Mac OS X v10.3 and later.

Declared In
`DVDPlayback.h`

DVDSetAudioStream

Sets the audio stream to use.

```
OSStatus DVDSetAudioStream (
    UInt16 inStreamNum
);
```

Parameters

inStreamNum

An integer from 1 to 8 that specifies an audio stream in the current title. DVD-Video media supports up to 8 audio streams per title.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

Before calling this function, DVD-Video media needs to be open and video playback started.

Availability

Available in Mac OS X v10.3 and later.

Declared In
`DVDPlayback.h`

DVDSetAudioVolume

Sets the playback audio volume setting (0 - 255).

```
OSStatus DVDSetAudioVolume (
    UInt16 inVolume
);
```

Parameters

inVolume

An integer in the range 0 to 255 that specifies the new video playback volume setting. Larger values are clamped to 255.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function changes the current playback audio volume setting, not the system audio volume setting. The actual audio volume experienced by the user is a function of these two settings:

```
actual audio volume = system volume * playback audio volume / 255
```

For example, if you set the playback volume to 127, the actual volume is approximately one-half the current system volume.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

CocoaDVDPlayer

Declared In

DVDPlayback.h

DVDSetChapter

Sets the chapter to play.

```
OSStatus DVDSetChapter (
    UInt16 inChapterNum
);
```

Parameters

inChapterNum

An integer that specifies the chapter to play.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function defines the specified chapter to be the current video playback chapter. If video playback is paused, this function starts playing the chapter immediately. Note that some discs do not allow jumping directly to a chapter.

Availability

Available in Mac OS X v10.3 and later.

Declared In
DVDPlayback.h

DVDSetDefaultAudioLanguageCode

Sets the default audio language code and extension.

```
OSStatus DVDSetDefaultAudioLanguageCode (
    DVDLanguageCode inCode,
    DVDAudioExtensionCode inExtension
);
```

Parameters

inCode

> A constant that specifies the default audio language for the specified audio extension.. For a list of possible values, see "Language Codes" (page 81). If no language is specified—that is, if you pass in kDVDLanguageNoPreference or kDVDLanguageCodeNone—the default audio language is matched to the language setting in the International Preferences Panel.

inExtension

> A constant that specifies an audio extension. For a list of possible values, see "Audio Extension Codes" (page 84).

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
This function selects an audio language other than the DVD's default language for a specified audio extension. Before calling this function, DVD-Video media needs to be open.

Availability
Available in Mac OS X v10.3 and later.

Declared In
DVDPlayback.h

DVDSetDefaultMenuLanguageCode

Sets the default menu language code.

```
OSStatus DVDSetDefaultMenuLanguageCode (
    DVDLanguageCode inCode
);
```

Parameters

inCode

> An integer code that specifies the default menu language for the current title. For a list of possible values, see "Language Codes" (page 81). If no language is specified—that is, if you pass in kDVDLanguageNoPreference or kDVDLanguageCodeNone—the default menu language is matched to the language setting in the International Preferences Panel.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
This function selects a language other than the DVD's default language for menus in the current title. Before calling this function, DVD-Video media needs to be open and video playback stopped.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`DVDPlayback.h`

DVDSetDefaultSubPictureLanguageCode

Sets the default subpicture language and extension when subpictures are enabled.

```
OSStatus DVDSetDefaultSubPictureLanguageCode (
    DVDLanguageCode inCode,
    DVDSubpictureExtensionCode inExtension
);
```

Parameters

inCode

> A constant that specifies the default language for the specified subpicture extension. For a list of possible values, see "Language Codes" (page 81). If no language is specified—that is, if you pass in `kDVDLanguageNoPreference` or `kDVDLanguageCodeNone`—the default subpicture language code is matched to the language setting in the International Preferences Panel.

inExtension

> A constant that specifies a subpicture extension. For a list of possible values, see "Subpicture Extension Codes" (page 85).

Return Value
A result code. If the current title is playing, this function does nothing and returns `kDVDErrorAlreadyPlaying`. For a list of other possible result codes, see "DVD Playback Result Codes" (page 95).

Discussion
This function is used to select a subpicture language other than the system default language. Before calling this function, DVD-Video media needs to be open and video playback stopped.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`DVDPlayback.h`

DVDSetDriveRegionCode

Sets the drive region code (requires user authentication).

```
OSStatus DVDSetDriveRegionCode (
   DVDRegionCode inCode,
   AuthorizationRef inAuthorization
);
```

Parameters

inCode

> A region code. For a list of possible values, see "Region Codes" (page 87).

inAuthorization

> An authorization object. For information on how to obtain a valid user authorization, see *Authorization Services Programming Guide*.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

In order to play a DVD-Video disc in a DVD drive, the region code assigned to the drive must match one of the region codes assigned to the disc. The region code assigned to a DVD drive may be changed a total of five times, after which the drive's region is permanently locked. The initial region code assigned to a new drive is set automatically when the user inserts a disc for the first time.

You can use the function DVDGetDiscRegionCode (page 28) to retrieve the region codes on a disc; you can use the function DVDGetDriveRegionCode (page 29) to retrieve the region code assigned to a drive and the number of remaining changes. If there is a mismatch and the number of remaining changes is greater than zero, you should display a dialog that asks whether the user wants to reset the drive region to match one of the disc regions. If the user wants to proceed, you should obtain an authorization object and call this function.

Availability

Available in Mac OS X v10.3 and later.

Declared In

DVDPlayback.h

DVDSetFatalErrorCallBack

Registers a callback function to handle unrecoverable errors.

```
OSStatus DVDSetFatalErrorCallBack (
   DVDFatalErrCallBackFunctionPtr inCallBackProc,
   void *inRefCon
);
```

Parameters

inCallBackProc

> A pointer to the callback function you provide to handle unrecoverable errors. For information about the requirements for such a function, see DVDFatalErrCallBackFunctionPtr (page 71).

inRefCon

> Application-defined data that is passed to your callback function when it is called to handle a fatal error.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion
This function registers a callback that you write to handle unrecoverable errors during video playback. You should call this function immediately after calling the function `DVDInitialize` (page 45). If you don't register a callback to handle unrecoverable errors, your application may crash when a fatal error occurs. For information about how to write such a callback function, see `DVDFatalErrCallBackFunctionPtr` (page 71).

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
`DVDPlayback.h`

DVDSetLastPlayBookmark

Sets the video playback position using a bookmark.

```
OSStatus DVDSetLastPlayBookmark (
    void *inBookMarkData,
    UInt32 inBookMarkDataSize
);
```

Parameters

inBookmarkData
> A generic pointer to memory that contains a bookmark.

inBookmarkDataSize
> The actual size of the specified bookmark in bytes.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
This function is used when playing media on a DVD-Video disc. Before calling this function, the media needs to be open. This function uses the specified bookmark to set the video playback position, but does not begin playing the media. See also `DVDGotoBookmark` (page 42).

Availability
Available in Mac OS X v10.4 and later.

Declared In
`DVDPlayback.h`

DVDSetSubPictureStream

Sets the current subpicture stream.

```
OSStatus DVDSetSubPictureStream (
   UInt16 inStreamNum
);
```

Parameters

inStreamNum

> A positive integer that specifies the subpicture stream to display. Subpicture stream numbers range from 1 to 32.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

Before calling this function, DVD-Video media needs to be open and video playback started.

Availability

Available in Mac OS X v10.3 and later.

Declared In

DVDPlayback.h

DVDSetTime

Sets the video playback position relative to the specified time code (elapsed, remaining).

```
OSStatus DVDSetTime (
   DVDTimeCode inTimeCode,
   DVDTimePosition inTime,
   UInt16 inFrames
);
```

Parameters

inTimeCode

> A time code constant. You must specify one of two values: kDVDTimeCodeElapsedSeconds or kDVDTimeCodeRemainingSeconds.

inTime

> An integer that specifies the position in seconds.

inFrames

> An integer that specifies the frame number. For the NTSC video format, the range of frame numbers is 0 to 29. For the PAL format, the range is 0 to 24.

Return Value

A result code. If the media does not allow this action, the result code is kDVDErrorUserActionNoOp. For other possible result codes, see "DVD Playback Result Codes" (page 95).

Discussion

This function sets the video playback position in the current title. Not all titles permit this operation, so it's important to check the result code returned by this function.

Before calling this function, DVD-Video media needs to be open and video playback started.

Availability

Available in Mac OS X v10.3 and later.

Declared In
DVDPlayback.h

DVDSetTimeEventRate

Sets the rate of the DVD Playback time event.

```
OSStatus DVDSetTimeEventRate (
   UInt32 inMilliseconds
);
```

Parameters

inMilliseconds

 The new interval between time events, specified in milliseconds.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
The default interval between DVD Playback time events is 900 milliseconds. You can use this function to lengthen or shorten the interval between time events. This function is relevant when you have used the function DVDRegisterEventCallBack (page 54) to register a callback for one or both of the time events: kDVDEventTitleTime and kDVDEventChapterTime . Time event callbacks are typically used for tasks such as updating the elapsed time or remaining time displayed in your user interface.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
DVDPlayback.h

DVDSetTitle

Sets the title to play.

```
OSStatus DVDSetTitle (
   UInt16 inTitleNum
);
```

Parameters

inTitleNum

 An integer in the range 1 to 99 that specifies the title to play.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
If the specified title exists on the current media and the media permits this operation, this function begins playing the title.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`DVDPlayback.h`

DVDSetVideoDisplay

Sets the graphics display for the current DVD-Video playback session.

```
OSStatus DVDSetVideoDisplay (
   CGDirectDisplayID inDisplay
);
```

Parameters

inDisplay

> The Quartz display ID for the graphics display you want to use for video playback. For information about how to get a display ID, see *Quartz Display Services Reference*.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
You need to call this function or `DVDSetVideoDevice` (page 104) each time you move the video playback window to a new graphics display. To avoid degrading video performance, you should not attempt to draw a video playback window that spans two different displays.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`DVDPlayback.h`

DVDSetVideoWindowID

Enables or disables DVD-Video playback in a Cocoa window.

```
OSStatus DVDSetVideoWindowID (
   UInt32 inVidWindowID
);
```

Parameters

inVidWindowID

> The window ID associated with a Cocoa window, or `NULL`. For information about when to pass `NULL`, see the Discussion below.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
If you're using a Cocoa window for DVD-Video playback, you need to use this function in two different situations:

- Immediately after calling the function `DVDInitialize` (page 45) to start a new DVD-Video playback session, you should call this function and pass the window ID associated with the Cocoa window.

■ When video is not playing and you want to draw into the area of the window you set by calling the function `DVDSetVideoBounds` (page 103), you should call this function and pass `NULL` in the `inVidWindowID` parameter.

To learn how to obtain the window number for a Cocoa window, see the description of the `windowNumber` method in the `NSWindow` class.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
`DVDPlayback.h`

DVDSleep

Saves video playback state information and suspends playback.

```
OSStatus DVDSleep (
    void
);
```

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
You should call this function when the operating system is putting the machine to sleep, to save video playback information. To find out when the machine is about to sleep:

■ Cocoa applications can add an observer to the workspace notification center for the notification `NSWorkspaceWillSleepNotification`.

■ Carbon applications can use the procedure described in Technical Q&A QA1340.

You should also call this function if the user changes the display configuration or resolution during video playback. To learn how to receive a notification that the display configuration is about to change, see *Quartz Display Services Reference*.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
`DVDPlayback.h`

DVDStepFrame

Steps one frame in the forward direction.

```
OSStatus DVDStepFrame (
    DVDScanDirection inDirection
);
```

Parameters

inDirection

> An integer that specifies the direction in which to step. This function does not support frame-stepping in the backward direction, so you should pass the constant `kDVDScanDirectionForward`.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
Before calling this function, the media needs to be open and video playback started. If the media is not paused when you call this function, this function pauses the media before stepping the frame in the forward direction.

Availability
Available in Mac OS X v10.3 and later.

Declared In
DVDPlayback.h

DVDStop

Stops playing the current title. If video playback is already stopped, the current video playback position is cleared.

```
OSStatus DVDStop (
    void
);
```

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
If you call this function while a title is playing, video playback is stopped, the current video playback position is saved, and the video area is cleared to the color black.

If you call this function a second time with no intervening play commands, the video playback position is cleared and the playback position is now the beginning of the disc. This is equivalent to calling the function `DVDClearLastPlayBookmark` (page 16).

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
DVDPlayback.h

DVDSwitchToDisplay

If a specified graphics display supports DVD-Video playback, sets the display for the current playback session.

```
OSStatus DVDSwitchToDisplay (
    CGDirectDisplayID newDisplay,
    Boolean *outSupported
);
```

Parameters

newDisplay

The Quartz display ID for the graphics display you want to use for video playback. For information about how to get a display ID, see *Quartz Display Services Reference*.

outSupported

A pointer to a Boolean variable. On return, a value of TRUE indicates that the specified graphics display can be used for DVD-Video playback. A value of FALSE indicates the display cannot be used for this purpose. If the value is FALSE, the current video playback display remains unchanged.

Return Value

A result code. If the new display is not supported, returns an error and maintains the current display. See "DVD Playback Result Codes" (page 95).

Discussion

This convenience function is equivalent to calling the function DVDIsSupportedDisplay (page 49) followed by a call to DVDSetVideoDisplay (page 65). This function works correctly even if there is no current video playback display.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

CocoaDVDPlayer

Declared In

DVDPlayback.h

DVDUnregisterEventCallBack

Unregisters an event callback.

```
OSStatus DVDUnregisterEventCallBack (
    DVDEventCallBackRef inCallBackID
);
```

Parameters

inCallBackID

The unique identifier for an event callback registration. See DVDRegisterEventCallBack (page 54).

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion
This function should be called once for each call to the function `DVDRegisterEventCallBack` (page 54).
You should unregister all event callbacks before calling the function `DVDDispose` (page 18) to end the
current DVD-Video playback session.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
`DVDPlayback.h`

DVDUpdateVideo

Not recommended.

```
OSStatus DVDUpdateVideo (
    void
);
```

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
This function does nothing and returns `noErr`.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`DVDPlayback.h`

DVDWakeUp

Restores video playback state information and resumes playback.

```
OSStatus DVDWakeUp (
    void
);
```

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
This function restores the video playback information saved by a previous call to the function `DVDSleep` (page
66). You should call this function whenever the operating system wakes up the machine and you have already
called `DVDSleep`. To find out when the machine is waking up from sleep:

■ Cocoa applications can add an observer to the workspace notification center for the notification
`NSWorkspaceDidWakeNotification`.

■ Carbon applications can use the procedure described in Technical Q&A QA1340.

You should also call this function if the user changes the display configuration or resolution during video playback. To learn how to receive a notification that the display configuration has changed, see *Quartz Display Services Reference*.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
CocoaDVDPlayer

Declared In
`DVDPlayback.h`

Callbacks

DVDEventCallBackFunctionPtr

Defines a pointer to an application-defined callback function that's called when a registered event occurs.

```
typedef void (*DVDEventCallBackFunctionPtr) (
    DVDEventCode inEventCode,
    UInt32 inEventValue1,
    UInt32 inEventValue2,
    UInt32 inRefCon
);
```

If you name your function `MyDVDEventCallBack`, you would declare it like this:

```
void MyDVDEventCallBack (
    DVDEventCode inEventCode,
    UInt32 inEventValue1,
    UInt32 inEventValue2,
    UInt32 inRefCon
);
```

Parameters

inEventCode

> A code that identifies the event. For a list of event codes, see "Event Codes" (page 93).

inEventValue1

> The first event-specific parameter value. For a list of one or both parameters associated with each event, see "Event Codes" (page 93).

inEventValue2

> The second event-specific parameter value. For a list of one or both parameters associated with each event, see "Event Codes" (page 93).

inRefCon

> Optional application-defined data. This is the same data you pass in when you register the callback using the function `DVDRegisterEventCallBack` (page 54).

Discussion

Your callback function is called from a different thread than the main application thread. You should not attempt to draw inside this function. Instead, cache any necessary information and do the actual drawing when you are back in the main thread for your application.

If your callback function is registered for multiple events, the function is called separately for each different event.

Availability

Available in Mac OS X version 10.3 and later.

Declared In

DVDPlayback.h

DVDFatalErrCallBackFunctionPtr

Defines a pointer to an application-defined callback function that's called when an unrecoverable error occurs during a DVD-Video playback session.

```
typedef void (*DVDFatalErrCallBackFunctionPtr) (
    DVDErrorCode inError,
    UInt32 inRefCon
);
```

If you name your function `MyDVDFatalErrCallBack`, you would declare it like this:

```
void MyDVDFatalErrCallBack (
    DVDErrorCode inError,
    UInt32 inRefCon
);
```

Parameters

inError

A code that identifies the fatal error. See "DVD Playback Result Codes" (page 95) for a list of possible values.

inRefCon

Optional application-defined data. This is the same data that you pass in when you register this callback using the function `DVDSetFatalErrorCallBack` (page 61).

Discussion

When an unrecoverable error occurs, the DVD-Video playback session cannot continue and your callback function is invoked to handle the error. Your function should report the fatal error to the user, perform any necessary cleanup, and call the function `DVDDispose` (page 18).

Availability

Available in Mac OS X version 10.3 and later.

Declared In

DVDPlayback.h

Data Types

DVDDiscID

Defines a unique identifier for a DVD-Video media folder.

```
typedef UInt8 DVDDiscID[8];
```

Discussion
The purpose of this 64-bit identifier is to help you distinguish between different media folders. You can obtain the identifier for a media folder by calling the function `DVDGetMediaUniqueID` (page 31).

Availability
Available in Mac OS X v10.4 and later.

Declared In
`DVDPlayback.h`

DVDTimePosition

Defines the number of seconds with respect to a time basis specified with a DVD Playback Services time code.

```
typedef UInt32 DVDTimePosition;
```

Discussion
This data type is used in conjunction with a time code to find an exact temporal location within the current title and chapter. See "Time Codes" (page 76).

Availability
Available in Mac OS X v10.3 and later.

Declared In
`DVDPlayback.h`

DVDErrorCode

Defines a non-zero result code passed to a callback function.

```
typedef OSStatus DVDErrorCode;
```

Discussion
This data type is used to pass a result code to the following types of callback functions:

- An event callback function that's registered for the `kDVDEventError` event. See "Event Codes" (page 93).

- A fatal error callback function. See DVDFatalErrCallBackFunctionPtr (page 71).

For information about the result codes defined for DVD Playback Services, see "DVD Playback Result Codes" (page 95).

Availability
Available in Mac OS X v10.3 and later.

Declared In
`DVDPlayback.h`

Constants

Playback States

Constants that specify the current play state of the framework.

```
enum {
    kDVDStateUnknown,
    kDVDStatePlaying,
    kDVDStatePlayingStill,
    kDVDStatePaused,
    kDVDStateStopped,
    kDVDStateScanning,
    kDVDStateIdle,
    kDVDStatePlayingSlow
};
typedef OSStatus DVDState;
```

Constants
`kDVDStateUnknown`

  The initial state of a new DVD-Video playback session.

  Available in Mac OS X v10.3 and later.

  Declared in `DVDPlayback.h`.

`kDVDStatePlaying`

  DVD-Video media is open and playing at normal rate (1x scan rate).

  Available in Mac OS X v10.3 and later.

  Declared in `DVDPlayback.h`.

`kDVDStatePlayingStill`

  DVD-Video media is open and playing a single frame.

  Available in Mac OS X v10.3 and later.

  Declared in `DVDPlayback.h`.

`kDVDStatePaused`

  DVD-Video media is open and paused or frame-stepping.

  Available in Mac OS X v10.3 and later.

  Declared in `DVDPlayback.h`.

`kDVDStateStopped`

  DVD-Video media is open and stopped (not playing).

  Available in Mac OS X v10.3 and later.

  Declared in `DVDPlayback.h`.

`kDVDStateScanning`

> DVD-Video media is open and playing in fast motion (greater than 1x scan rate).
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDStateIdle`

> Not used.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDStatePlayingSlow`

> DVD-Video media is open and playing in slow motion (less than 1x scan rate).
>
> Available in Mac OS X v10.4 and later.
>
> Declared in `DVDPlayback.h`.

Discussion

These constants are passed back to your application when calling the function `DVDGetState` (page 37) or when handling the `kDVDEventPlayback` event.

Declared In

`DVDPlayback.h`

Playback Menus

Constants that specify input menus for a DVD-Video volume.

```
enum {
    kDVDMenuTitle,
    kDVDMenuRoot,
    kDVDMenuSubPicture,
    kDVDMenuAudio,
    kDVDMenuAngle,
    kDVDMenuPTT,
    kDVDMenuNone
};
typedef UInt32 DVDMenu;
```

Constants

`kDVDMenuTitle`

> A title or top menu.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDMenuRoot`

> A root menu.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDMenuSubPicture`

> A menu used to select subpictures.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

```
kDVDMenuAudio
```
>   An audio menu.
>
>   Available in Mac OS X v10.3 and later.
>
>   Declared in `DVDPlayback.h`.

```
kDVDMenuAngle
```
>   A viewing angle menu.
>
>   Available in Mac OS X v10.3 and later.
>
>   Declared in `DVDPlayback.h`.

```
kDVDMenuPTT
```
>   A part-of-title (PTT) or chapter menu.
>
>   Available in Mac OS X v10.3 and later.
>
>   Declared in `DVDPlayback.h`.

```
kDVDMenuNone
```
>   Menu not defined.
>
>   Available in Mac OS X v10.3 and later.
>
>   Declared in `DVDPlayback.h`.

Discussion

These constants are used when calling `DVDGoToMenu` (page 42) and other menu-related functions, or when handling the `kDVDEventMenuCalled` event.

Declared In

`DVDPlayback.h`

User Navigation Directions

Constants that specify keyboard operations used to select a menu button and execute the action or command associated with a button.

```
enum {
    kDVDUserNavigationMoveUp = 1,
    kDVDUserNavigationMoveDown,
    kDVDUserNavigationMoveLeft,
    kDVDUserNavigationMoveRight,
    kDVDUserNavigationEnter
};
typedef UInt32 DVDUserNavigation;
```

Constants

```
kDVDUserNavigationMoveUp
```
>   Move to and select the button above the current location.
>
>   Available in Mac OS X v10.3 and later.
>
>   Declared in `DVDPlayback.h`.

```
kDVDUserNavigationMoveDown
```
>   Move to and select the button below the current location.
>
>   Available in Mac OS X v10.3 and later.
>
>   Declared in `DVDPlayback.h`.

`kDVDUserNavigationMoveLeft`

  Move to and select the button to the left of the current location.

  Available in Mac OS X v10.3 and later.

  Declared in `DVDPlayback.h`.

`kDVDUserNavigationMoveRight`

  Move to and select the button to the right of the current location.

  Available in Mac OS X v10.3 and later.

  Declared in `DVDPlayback.h`.

`kDVDUserNavigationEnter`

  Execute the action of the selected button.

  Available in Mac OS X v10.3 and later.

  Declared in `DVDPlayback.h`.

Discussion

These constants are used when calling the function `DVDDoUserNavigation` (page 19).

Declared In

`DVDPlayback.h`

Time Codes

Constants that specify a temporal location in the video stream of the current title or chapter.

```
enum {
    kDVDTimeCodeUninitialized,
    kDVDTimeCodeElapsedSeconds,
    kDVDTimeCodeRemainingSeconds,
    kDVDTimeCodeTitleDurationSeconds,
    kDVDTimeCodeChapterElapsedSeconds,
    kDVDTimeCodeChapterRemainingSeconds,
    kDVDTimeCodeChapterDurationSeconds
};
typedef SInt16 DVDTimeCode;
```

Constants

`kDVDTimeCodeUninitialized`

  Not used.

  Available in Mac OS X v10.3 and later.

  Declared in `DVDPlayback.h`.

`kDVDTimeCodeElapsedSeconds`

  Number of seconds after a specified position.

  Available in Mac OS X v10.3 and later.

  Declared in `DVDPlayback.h`.

`kDVDTimeCodeRemainingSeconds`

  Number of seconds before a specified position.

  Available in Mac OS X v10.3 and later.

  Declared in `DVDPlayback.h`.

`kDVDTimeCodeTitleDurationSeconds`
> Number of seconds in the current title.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDTimeCodeChapterElapsedSeconds`
> Number of seconds elapsed since the beginning of the current chapter.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDTimeCodeChapterRemainingSeconds`
> Number of seconds remaining before the end of the current chapter.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDTimeCodeChapterDurationSeconds`
> Number of seconds in the current chapter.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

Discussion
These constants are used, along with the DVD time position, to find an exact temporal position within the current title or chapter. See the functions DVDSetTime (page 63) and DVDGetTime (page 39).

Declared In
`DVDPlayback.h`

Scan Directions

Constants that specify the direction of play.

```
enum {
    kDVDScanDirectionForward,
    kDVDScanDirectionBackward
};
typedef SInt8 DVDScanDirection;
```

Constants
`kDVDScanDirectionForward`
> Scan in the forward direction.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDScanDirectionBackward`
> Scan in the reverse direction.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

Discussion
These constants are used along with the scan rate to specify the direction and speed of play. See the functions DVDScan (page 55), DVDGetScanRate (page 37), and DVDStepFrame (page 66).

Declared In
`DVDPlayback.h`

Scan Rates

Constants that specify the rate at which to scan.

```
enum {
    kDVDScanRateOneEigth    = -8,
    kDVDScanRateOneFourth   = -4,
    kDVDScanRateOneHalf     = -2,
    kDVDScanRate1x          = 1,
    kDVDScanRate2x          = 2,
    kDVDScanRate4x          = 4,
    kDVDScanRate8x          = 8,
    kDVDScanRate16x         = 16,
    kDVDScanRate32x         = 32
};
typedef SInt16 DVDScanRate;
```

Discussion
These constants are used along with the scan direction to specify the speed and direction of play. The constant `kDVDScanRate1x` represents the normal playback speed; the slower and faster playback speeds are multiples of the normal speed. See the functions `DVDScan` (page 55) and `DVDGetScanRate` (page 37).

Availability
Available in Mac OS X version 10.3 and later.

Declared In
`DVDPlayback.h`

Aspect Ratios

Constants that specify the current aspect ratio.

```
enum {
    kDVDAspectRatioUninitialized,
    kDVDAspectRatio4x3,
    kDVDAspectRatio4x3PanAndScan,
    kDVDAspectRatio16x9,
    kDVDAspectRatioLetterBox
};
typedef SInt16 DVDAspectRatio;
```

Constants
`kDVDAspectRatioUninitialized`

An unspecified aspect ratio.

Available in Mac OS X v10.3 and later.

Declared in `DVDPlayback.h`.

`kDVDAspectRatio4x3`

> The process of displaying a 4:3 video frame on a screen with the same aspect ratio.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDAspectRatio4x3PanAndScan`

> The process of choosing a sample area of a 16:9 video frame for display on a 4:3 screen, using center-of-interest information stored on the DVD.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDAspectRatio16x9`

> The process of displaying a 16:9 video frame on a screen with the same aspect ratio.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDAspectRatioLetterBox`

> The process of displaying the entire 16:9 video frame on a 4:3 screen by shrinking the frame and adding mattes or black bars above and below the frame.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

Discussion

These constants are used to specify an aspect ratio when calling the function `DVDGetAspectRatio` (page 20) and when handling the `kDVDEventDisplayMode` event.

Declared In

`DVDPlayback.h`

Video Broadcast Formats

Constants that specify video broadcast formats used in DVD media.

```
enum {
    kDVDFormatUninitialized,
    kDVDFormatNTSC,
    kDVDFormatPAL
};
typedef SInt16 DVDFormat;
```

Constants

`kDVDFormatUninitialized`

> An unspecified video format.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDFormatNTSC`

> National Television Standards Committee (NTSC) video format used in North America and Japan.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDFormatPAL`

> Phase Alternating Line (PAL) video format widely used in Europe, South Asia, Africa, and South America.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

Discussion

These constants are passed back when calling the function `DVDGetFormatStandard` (page 30) to indicate the video format used in a DVD.

Declared In

`DVDPlayback.h`

Audio Stream Formats

Constants that specify the audio stream format.

```
enum {
    kDVDAudioUnknownFormat,
    kDVDAudioAC3Format,
    kDVDAudioMPEG1Format,
    kDVDAudioMPEG2Format,
    kDVDAudioPCMFormat,
    kDVDAudioDTSFormat,
    kDVDAudioSDDSFormat
};
typedef SInt16 DVDAudioFormat;
```

Constants

`kDVDAudioUnknownFormat`

> An unspecified audio format.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDAudioAC3Format`

> The Dolby Digital AC-3 format.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDAudioMPEG1Format`

> The MPEG-1 Layer II digital format.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDAudioMPEG2Format`

> The MPEG-2 digital format.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDAudioPCMFormat`

> The linear pulse-code modulation (PCM) format used on CDs.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDAudioDTSFormat`

    The Digital Theater Systems (DTS) format. This format is not supported.

    Available in Mac OS X v10.3 and later.

    Declared in `DVDPlayback.h`.

`kDVDAudioSDDSFormat`

    The Sony Dynamic Digital Sound (SDDS) format. This format is not supported.

    Available in Mac OS X v10.3 and later.

    Declared in `DVDPlayback.h`.

Discussion

These constants are used when calling the functions `DVDGetAudioStreamFormat` (page 23) and `DVDGetAudioStreamFormatByStream` (page 23).

Declared In

`DVDPlayback.h`

Language Codes

Constants that specify languages used in subpictures, audio, and menus.

```
enum {
    kDVDLanguageCodeUninitialized    = FOUR_CHAR_CODE('??  '),
    kDVDLanguageNoPreference         = FOUR_CHAR_CODE('**  '),
    kDVDLanguageCodeNone             = FOUR_CHAR_CODE('OO  '),
    kDVDLanguageCodeAfar             = FOUR_CHAR_CODE('aa  '),
    kDVDLanguageCodeAbkhazian        = FOUR_CHAR_CODE('ab  '),
    kDVDLanguageCodeAfrikaans        = FOUR_CHAR_CODE('af  '),
    kDVDLanguageCodeAmharic          = FOUR_CHAR_CODE('am  '),
    kDVDLanguageCodeArabic           = FOUR_CHAR_CODE('ar  '),
    kDVDLanguageCodeAssamese         = FOUR_CHAR_CODE('as  '),
    kDVDLanguageCodeAymara           = FOUR_CHAR_CODE('ay  '),
    kDVDLanguageCodeAzerbaijani      = FOUR_CHAR_CODE('az  '),
    kDVDLanguageCodeBashkir          = FOUR_CHAR_CODE('ba  '),
    kDVDLanguageCodeByelorussian     = FOUR_CHAR_CODE('be  '),
    kDVDLanguageCodeBulgarian        = FOUR_CHAR_CODE('bg  '),
    kDVDLanguageCodeBihari           = FOUR_CHAR_CODE('bh  '),
    kDVDLanguageCodeBislama          = FOUR_CHAR_CODE('bi  '),
    kDVDLanguageCodeBengali          = FOUR_CHAR_CODE('bn  '),
    kDVDLanguageCodeTibetan          = FOUR_CHAR_CODE('bo  '),
    kDVDLanguageCodeBreton           = FOUR_CHAR_CODE('br  '),
    kDVDLanguageCodeCatalan          = FOUR_CHAR_CODE('ca  '),
    kDVDLanguageCodeCorsican         = FOUR_CHAR_CODE('co  '),
    kDVDLanguageCodeCzech            = FOUR_CHAR_CODE('cs  '),
    kDVDLanguageCodeWelsh            = FOUR_CHAR_CODE('cy  '),
    kDVDLanguageCodeDanish           = FOUR_CHAR_CODE('da  '),
    kDVDLanguageCodeGerman           = FOUR_CHAR_CODE('de  '),
    kDVDLanguageCodeBhutani          = FOUR_CHAR_CODE('dz  '),
    kDVDLanguageCodeGreek            = FOUR_CHAR_CODE('el  '),
    kDVDLanguageCodeEnglish          = FOUR_CHAR_CODE('en  '),
    kDVDLanguageCodeEsperanto        = FOUR_CHAR_CODE('eo  '),
    kDVDLanguageCodeSpanish          = FOUR_CHAR_CODE('es  '),
    kDVDLanguageCodeEstonian         = FOUR_CHAR_CODE('et  '),
    kDVDLanguageCodeBasque           = FOUR_CHAR_CODE('eu  '),
    kDVDLanguageCodePersian          = FOUR_CHAR_CODE('fa  '),
    kDVDLanguageCodeFinnish          = FOUR_CHAR_CODE('fi  '),
    kDVDLanguageCodeFiji             = FOUR_CHAR_CODE('fj  '),
    kDVDLanguageCodeFaeroese         = FOUR_CHAR_CODE('fo  '),
    kDVDLanguageCodeFrench           = FOUR_CHAR_CODE('fr  '),
    kDVDLanguageCodeFrisian          = FOUR_CHAR_CODE('fy  '),
    kDVDLanguageCodeIrish            = FOUR_CHAR_CODE('ga  '),
    kDVDLanguageCodeScotsGaelic      = FOUR_CHAR_CODE('gd  '),
    kDVDLanguageCodeGalician         = FOUR_CHAR_CODE('gl  '),
    kDVDLanguageCodeGuarani          = FOUR_CHAR_CODE('gn  '),
    kDVDLanguageCodeGujarati         = FOUR_CHAR_CODE('gu  '),
    kDVDLanguageCodeHausa            = FOUR_CHAR_CODE('ha  '),
    kDVDLanguageCodeHindi            = FOUR_CHAR_CODE('hi  '),
    kDVDLanguageCodeCroatian         = FOUR_CHAR_CODE('hr  '),
    kDVDLanguageCodeHungarian        = FOUR_CHAR_CODE('hu  '),
    kDVDLanguageCodeArmenian         = FOUR_CHAR_CODE('hy  '),
    kDVDLanguageCodeInterlingua      = FOUR_CHAR_CODE('ia  '),
    kDVDLanguageCodeInterlingue      = FOUR_CHAR_CODE('ie  '),
    kDVDLanguageCodeInupiak          = FOUR_CHAR_CODE('ik  '),
    kDVDLanguageCodeIndonesian       = FOUR_CHAR_CODE('in  '),
    kDVDLanguageCodeIcelandic        = FOUR_CHAR_CODE('is  '),
    kDVDLanguageCodeItalian          = FOUR_CHAR_CODE('it  '),
    kDVDLanguageCodeHebrew           = FOUR_CHAR_CODE('iw  '),
    kDVDLanguageCodeJapanese         = FOUR_CHAR_CODE('ja  '),
    kDVDLanguageCodeYiddish          = FOUR_CHAR_CODE('ji  '),
```

```
kDVDLanguageCodeJavanese          = FOUR_CHAR_CODE('jw  '),
kDVDLanguageCodeGeorgian          = FOUR_CHAR_CODE('ka  '),
kDVDLanguageCodeKazakh            = FOUR_CHAR_CODE('kk  '),
kDVDLanguageCodeGreenlandic       = FOUR_CHAR_CODE('kl  '),
kDVDLanguageCodeCambodian         = FOUR_CHAR_CODE('km  '),
kDVDLanguageCodeKannada           = FOUR_CHAR_CODE('kn  '),
kDVDLanguageCodeKorean            = FOUR_CHAR_CODE('ko  '),
kDVDLanguageCodeKashmiri          = FOUR_CHAR_CODE('ks  '),
kDVDLanguageCodeKurdish           = FOUR_CHAR_CODE('ku  '),
kDVDLanguageCodeKirghiz           = FOUR_CHAR_CODE('ky  '),
kDVDLanguageCodeLatin             = FOUR_CHAR_CODE('la  '),
kDVDLanguageCodeLingala           = FOUR_CHAR_CODE('ln  '),
kDVDLanguageCodeLaothian          = FOUR_CHAR_CODE('lo  '),
kDVDLanguageCodeLithuanian        = FOUR_CHAR_CODE('lt  '),
kDVDLanguageCodeLatvian           = FOUR_CHAR_CODE('lv  '),
kDVDLanguageCodeMalagasy          = FOUR_CHAR_CODE('mg  '),
kDVDLanguageCodeMaori             = FOUR_CHAR_CODE('mi  '),
kDVDLanguageCodeMacedonian        = FOUR_CHAR_CODE('mk  '),
kDVDLanguageCodeMalayalam         = FOUR_CHAR_CODE('ml  '),
kDVDLanguageCodeMongolian         = FOUR_CHAR_CODE('mn  '),
kDVDLanguageCodeMoldavian         = FOUR_CHAR_CODE('mo  '),
kDVDLanguageCodeMarathi           = FOUR_CHAR_CODE('mr  '),
kDVDLanguageCodeMalay             = FOUR_CHAR_CODE('ms  '),
kDVDLanguageCodeMaltese           = FOUR_CHAR_CODE('mt  '),
kDVDLanguageCodeBurmese           = FOUR_CHAR_CODE('my  '),
kDVDLanguageCodeNauru             = FOUR_CHAR_CODE('na  '),
kDVDLanguageCodeNepali            = FOUR_CHAR_CODE('ne  '),
kDVDLanguageCodeDutch             = FOUR_CHAR_CODE('nl  '),
kDVDLanguageCodeNorwegian         = FOUR_CHAR_CODE('no  '),
kDVDLanguageCodeOccitan           = FOUR_CHAR_CODE('oc  '),
kDVDLanguageCodeOromo             = FOUR_CHAR_CODE('om  '),
kDVDLanguageCodeOriya             = FOUR_CHAR_CODE('or  '),
kDVDLanguageCodePunjabi           = FOUR_CHAR_CODE('pa  '),
kDVDLanguageCodePolish            = FOUR_CHAR_CODE('pl  '),
kDVDLanguageCodePashto            = FOUR_CHAR_CODE('ps  '),
kDVDLanguageCodePortugese         = FOUR_CHAR_CODE('pt  '),
kDVDLanguageCodeQuechua           = FOUR_CHAR_CODE('qu  '),
kDVDLanguageCodeRhaetoRomance     = FOUR_CHAR_CODE('rm  '),
kDVDLanguageCodeKirundi           = FOUR_CHAR_CODE('rn  '),
kDVDLanguageCodeRomanian          = FOUR_CHAR_CODE('ro  '),
kDVDLanguageCodeRussian           = FOUR_CHAR_CODE('ru  '),
kDVDLanguageCodeKinyarwanda       = FOUR_CHAR_CODE('rw  '),
kDVDLanguageCodeSanskrit          = FOUR_CHAR_CODE('sa  '),
kDVDLanguageCodeSindhi            = FOUR_CHAR_CODE('sd  '),
kDVDLanguageCodeSangro            = FOUR_CHAR_CODE('sg  '),
kDVDLanguageCodeSerboCroatian     = FOUR_CHAR_CODE('sh  '),
kDVDLanguageCodeSinghalese        = FOUR_CHAR_CODE('si  '),
kDVDLanguageCodeSlovak            = FOUR_CHAR_CODE('sk  '),
kDVDLanguageCodeSlovenian         = FOUR_CHAR_CODE('sl  '),
kDVDLanguageCodeSamoan            = FOUR_CHAR_CODE('sm  '),
kDVDLanguageCodeShona             = FOUR_CHAR_CODE('sn  '),
kDVDLanguageCodeSomali            = FOUR_CHAR_CODE('so  '),
kDVDLanguageCodeAlbanian          = FOUR_CHAR_CODE('sq  '),
kDVDLanguageCodeSerbian           = FOUR_CHAR_CODE('sr  '),
kDVDLanguageCodeSiswati           = FOUR_CHAR_CODE('ss  '),
kDVDLanguageCodeSesotho           = FOUR_CHAR_CODE('st  '),
kDVDLanguageCodeSudanese          = FOUR_CHAR_CODE('su  '),
kDVDLanguageCodeSwedish           = FOUR_CHAR_CODE('sv  '),
```

```
    kDVDLanguageCodeSwahili         = FOUR_CHAR_CODE('sw  '),
    kDVDLanguageCodeTamil           = FOUR_CHAR_CODE('ta  '),
    kDVDLanguageCodeTelugu          = FOUR_CHAR_CODE('te  '),
    kDVDLanguageCodeTajik           = FOUR_CHAR_CODE('tg  '),
    kDVDLanguageCodeThai            = FOUR_CHAR_CODE('th  '),
    kDVDLanguageCodeTigrinya        = FOUR_CHAR_CODE('ti  '),
    kDVDLanguageCodeTurkmen         = FOUR_CHAR_CODE('tk  '),
    kDVDLanguageCodeTagalog         = FOUR_CHAR_CODE('tl  '),
    kDVDLanguageCodeSetswana        = FOUR_CHAR_CODE('tn  '),
    kDVDLanguageCodeTonga           = FOUR_CHAR_CODE('to  '),
    kDVDLanguageCodeTurkish         = FOUR_CHAR_CODE('tr  '),
    kDVDLanguageCodeTsonga          = FOUR_CHAR_CODE('ts  '),
    kDVDLanguageCodeTatar           = FOUR_CHAR_CODE('tt  '),
    kDVDLanguageCodeTwi             = FOUR_CHAR_CODE('tw  '),
    kDVDLanguageCodeUkranian        = FOUR_CHAR_CODE('uk  '),
    kDVDLanguageCodeUrdu            = FOUR_CHAR_CODE('ur  '),
    kDVDLanguageCodeUzbek           = FOUR_CHAR_CODE('uz  '),
    kDVDLanguageCodeVietnamese      = FOUR_CHAR_CODE('vi  '),
    kDVDLanguageCodeVolapuk         = FOUR_CHAR_CODE('vo  '),
    kDVDLanguageCodeWolof           = FOUR_CHAR_CODE('wo  '),
    kDVDLanguageCodeXhosa           = FOUR_CHAR_CODE('xh  '),
    kDVDLanguageCodeYoruba          = FOUR_CHAR_CODE('yo  '),
    kDVDLanguageCodeChinese         = FOUR_CHAR_CODE('zh  '),
    kDVDLanguageCodeZulu            = FOUR_CHAR_CODE('zu  ')
};
typedef OSType DVDLanguageCode;
```

Discussion
These constants are used when calling the functions described in "Accessing Language Codes" (page 14).
By convention, ISO 639 language codes are used to define these constants.

Availability
Available in Mac OS X version 10.3 and later.

Declared In
`DVDPlayback.h`

Audio Extension Codes

Constants that specify the purpose of an audio channel.

```
enum {
    kDVDAudioExtensionCodeNotSpecified = 0,
    kDVDAudioExtensionCodeNormalCaptions = 1,
    kDVDAudioExtensionCodeNVisualImpaired= 2,
    kDVDAudioExtensionCodeDirectorsComment1= 3,
    kDVDAudioExtensionCodeDirectorsComment2= 4
};
typedef OSType DVDAudioExtensionCode;
```

Constants
`kDVDAudioExtensionCodeNotSpecified`

> Audio extension not specified.

> Available in Mac OS X v10.3 and later.

> Declared in `DVDPlayback.h`.

`kDVDAudioExtensionCodeNormalCaptions`
> Audio channel for extended information about the title.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDAudioExtensionCodeNVisualImpaired`
> Audio channel for the visually impaired.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDAudioExtensionCodeDirectorsComment1`
> First audio channel for the artistic director's comments.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDAudioExtensionCodeDirectorsComment2`
> Second audio channel for the artistic director's comments.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

Discussion

These constants are used along with a language code to specify the characteristics of an audio channel. See the audio-related functions described in "Accessing Language Codes" (page 14).

Declared In

`DVDPlayback.h`

Subpicture Extension Codes

Constants that specify the purpose of a subpicture stream.

```
enum {
    kDVDSubpictureExtensionCodeNotSpecified       = 0,
    kDVDSubpictureExtensionCodeCaptionNormalSize  = 1,
    kDVDSubpictureExtensionCodeCaptionBiggerSize  = 2,
    kDVDSubpictureExtensionCodeCaption4Children   = 3,
    kDVDSubpictureExtensionCodeClosedCaptionNormalSize  = 5,
    kDVDSubpictureExtensionCodeClosedCaptionBiggerSize  = 6,
    kDVDSubpictureExtensionCodeClosedCaption4Children   = 7,
    kDVDSubpictureExtensionCodeForcedCaption      = 9,
    kDVDSubpictureExtensionDirectorsCommentNormalSize  = 13,
    kDVDSubpictureExtensionDirectorsCommentBiggerSize  = 14,
    kDVDSubpictureExtensionCodeDirectorsComment4Children  = 15
};
typedef OSType DVDSubpictureExtensionCode;
```

Constants

`kDVDSubpictureExtensionCodeNotSpecified`
> Subpicture not specified.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDSubpictureExtensionCodeCaptionNormalSize`
> A language-caption subpicture.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDSubpictureExtensionCodeCaptionBiggerSize`
> A large language-caption subpicture.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDSubpictureExtensionCodeCaption4Children`
> A language-caption subpicture for children.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDSubpictureExtensionCodeClosedCaptionNormalSize`
> A closed-caption subpicture.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDSubpictureExtensionCodeClosedCaptionBiggerSize`
> A large closed-caption subpicture.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDSubpictureExtensionCodeClosedCaption4Children`
> A closed-caption subpicture for children.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDSubpictureExtensionCodeForcedCaption`
> A subpicture which is always visible, regardless of the viewer's preferences.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDSubpictureExtensionDirectorsCommentNormalSize`
> A normal subpicture for director's comments.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDSubpictureExtensionDirectorsCommentBiggerSize`
> A large subpicture for director's comments.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDSubpictureExtensionDirectorsComment4Children`
> A subpicture for director's comments for children.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

Discussion

These constants are used along with a language code to specify the characteristics of a subpicture. See the subpicture functions described in "Accessing Language Codes" (page 14).

Declared In
DVDPlayback.h

Region Codes

Constants that specify marketing regions for DVD discs and playback devices.

```
enum {
    kDVDRegionCodeUninitialized = 0xff,
    kDVDRegionCode1 = 0xfe,
    kDVDRegionCode2 = 0xfd,
    kDVDRegionCode3 = 0xfb,
    kDVDRegionCode4 = 0xf7,
    kDVDRegionCode5 = 0xef,
    kDVDRegionCode6 = 0xdf,
    kDVDRegionCode7 = 0xbf,
    kDVDRegionCode8 = 0x7f

};
typedef UInt32 DVDRegionCode;
```

Constants
kDVDRegionCodeUninitialized
> The default region code for an uninitialized DVD drive.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in DVDPlayback.h.

kDVDRegionCode1
> Region 1: North America.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in DVDPlayback.h.

kDVDRegionCode2
> Region 2: Japan, Europe, South Africa.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in DVDPlayback.h.

kDVDRegionCode3
> Region 3: Southeast Asia.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in DVDPlayback.h.

kDVDRegionCode4
> Region 4: Australia, New Zealand, Central & South America.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in DVDPlayback.h.

kDVDRegionCode5
> Region 5: Northwest Asia and North Africa.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in DVDPlayback.h.

`kDVDRegionCode6`

> Region 6: China.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDRegionCode7`

> Region 7: Unassigned.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDRegionCode8`

> Region 8: Special venues (airplanes, hotels, cruise ships).
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

Discussion

A DVD Playback region code is a bitfield in which a zero in bit *n* is used to specify region *n*. The bits are numbered from right to left, and only the least significant 8 bits are used. For example, the region code 11101111 or `0xef` specifies region 5. A region code can also specify more than one region; the region code 11110110 or `0xf6`, for example, specifies regions 1 and 4.

Most DVD drives are designed to play only those discs that are authorized for use in a specific region. Region codes are used to:

- Assign a new region to a DVD drive when calling the function `DVDSetDriveRegionCode` (page 60).

- Specify the region currently assigned to a DVD drive when calling the function `DVDGetDriveRegionCode` (page 29).

- Specify one or more regions assigned to a DVD disc when calling the function `DVDGetDiscRegionCode` (page 28).

Declared In

`DVDPlayback.h`

Domain Codes

Constants that specify playback domains or modes that define a set of possible actions.

```
enum {
    kDVDFPDomain                  = 0,
    kDVDVMGMDomain                = 1,
    kDVDVTSMDomain                = 2,
    kDVDTTDomain                  = 3,
    kDVDSTOPDomain                = 4,
    kDVDAMGMDomain                = 5,
    kDVDTTGRDomain                = 6
};
typedef UInt32 DVDDomainCode;
```

Constants

`kDVDFPDomain`

First Play domain.

Available in Mac OS X v10.3 and later.

Declared in `DVDPlayback.h`.

`kDVDVMGMDomain`

Video Manager Menu domain.

Available in Mac OS X v10.3 and later.

Declared in `DVDPlayback.h`.

`kDVDVTSMDomain`

Video Title Set Menu domain.

Available in Mac OS X v10.3 and later.

Declared in `DVDPlayback.h`.

`kDVDTTDomain`

Title domain.

Available in Mac OS X v10.3 and later.

Declared in `DVDPlayback.h`.

`kDVDSTOPDomain`

Stop State domain.

Available in Mac OS X v10.3 and later.

Declared in `DVDPlayback.h`.

`kDVDAMGMDomain`

Audio Manager Menu domain (DVD-Audio only, not used).

Available in Mac OS X v10.3 and later.

Declared in `DVDPlayback.h`.

`kDVDTTGRDomain`

Title Group domain (DVD-Audio only, not used).

Available in Mac OS X v10.3 and later.

Declared in `DVDPlayback.h`.

Discussion

These constants are used to specify the new domain when handling the `kDVDEventDomain` event. See .

Declared In

`DVDPlayback.h`

UOP Codes

Constants that specify user operations (UOP) which are currently disabled.

```
enum {
    kDVDUOPTimePlaySearch            = 0x00000001,
    kDVDUOPPTTPlaySearch             = 0x00000002,
    kDVDUOPTitlePlay                 = 0x00000004,
    kDVDUOPStop                      = 0x00000008,
    kDVDUOPGoUp                      = 0x00000010,
    kDVDUOPTimePTTSearch             = 0x00000020,
    kDVDUOPPrevTopPGSearch           = 0x00000040,
    kDVDUOPNextPGSearch              = 0x00000080,
    kDVDUOPForwardScan               = 0x00000100,
    kDVDUOPBackwardScan              = 0x00000200,
    kDVDUOPMenuCallTitle             = 0x00000400,
    kDVDUOPMenuCallRoot              = 0x00000800,
    kDVDUOPMenuCallSubPicture        = 0x00001000,
    kDVDUOPMenuCallAudio             = 0x00002000,
    kDVDUOPMenuCallAngle             = 0x00004000,
    kDVDUOPMenuCallPTT               = 0x00008000,
    kDVDUOPResume                    = 0x00010000,
    kDVDUOPButton                    = 0x00020000,
    kDVDUOPStillOff                  = 0x00040000,
    kDVDUOPPauseOn                   = 0x00080000,
    kDVDUOPAudioStreamChange         = 0x00100000,
    kDVDUOPSubPictureStreamChange    = 0x00200000,
    kDVDUOPAngleChange               = 0x00400000,
    kDVDUOPKaraokeModeChange         = 0x00800000,
    kDVDUOPVideoModeChange           = 0x01000000,
    kDVDUOPScanOff                   = 0x02000000,
    kDVDUOPPauseOff                  = 0x04000000
};
typedef UInt32 DVDUOPCode;
```

Constants

`kDVDUOPTimePlaySearch`

> The function `DVDSetTime` (page 63) is disabled.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDUOPPTTPlaySearch`

> The function `DVDSetTime` (page 63) is disabled.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDUOPTitlePlay`

> The functions `DVDReturnToTitle` (page 55), `DVDSetTitle` (page 64), and `DVDSetChapter` (page 58) are disabled.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDUOPStop`

> The function `DVDStop` (page 67) is disabled.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDUOPGoUp`

>The function `DVDGoBackOneLevel` (page 41) is disabled.
>
>Available in Mac OS X v10.3 and later.
>
>Declared in `DVDPlayback.h`.

`kDVDUOPTimePTTSearch`

>The function `DVDSetChapter` (page 58) is disabled.
>
>Available in Mac OS X v10.3 and later.
>
>Declared in `DVDPlayback.h`.

`kDVDUOPPrevTopPGSearch`

>The function `DVDPreviousChapter` (page 53) is disabled.
>
>Available in Mac OS X v10.3 and later.
>
>Declared in `DVDPlayback.h`.

`kDVDUOPNextPGSearch`

>The function `DVDNextChapter` (page 50) is disabled.
>
>Available in Mac OS X v10.3 and later.
>
>Declared in `DVDPlayback.h`.

`kDVDUOPForwardScan`

>The function `DVDScan` (page 55) in the forward direction is disabled.
>
>Available in Mac OS X v10.3 and later.
>
>Declared in `DVDPlayback.h`.

`kDVDUOPBackwardScan`

>The function `DVDScan` (page 55) in the backward direction is disabled.
>
>Available in Mac OS X v10.3 and later.
>
>Declared in `DVDPlayback.h`.

`kDVDUOPMenuCallTitle`

>The function `DVDGoToMenu` (page 42) for a title menu is disabled.
>
>Available in Mac OS X v10.3 and later.
>
>Declared in `DVDPlayback.h`.

`kDVDUOPMenuCallRoot`

>The function `DVDGoToMenu` (page 42) for a root menu is disabled.
>
>Available in Mac OS X v10.3 and later.
>
>Declared in `DVDPlayback.h`.

`kDVDUOPMenuCallSubPicture`

>The function `DVDGoToMenu` (page 42) for a subpicture menu is disabled.
>
>Available in Mac OS X v10.3 and later.
>
>Declared in `DVDPlayback.h`.

`kDVDUOPMenuCallAudio`

>The function `DVDGoToMenu` (page 42) for an audio menu is disabled.
>
>Available in Mac OS X v10.3 and later.
>
>Declared in `DVDPlayback.h`.

`kDVDUOPMenuCallAngle`

The function `DVDGoToMenu` (page 42) for an angle menu is disabled.

Available in Mac OS X v10.3 and later.

Declared in `DVDPlayback.h`.

`kDVDUOPMenuCallPTT`

The function `DVDGoToMenu` (page 42) for a chapter menu is disabled.

Available in Mac OS X v10.3 and later.

Declared in `DVDPlayback.h`.

`kDVDUOPResume`

Not used.

Available in Mac OS X v10.3 and later.

Declared in `DVDPlayback.h`.

`kDVDUOPButton`

The functions `DVDDoMenuClick` (page 99), `DVDDoMenuMouseOver` (page 100), and `DVDDoUserNavigation` (page 19) are disabled.

Available in Mac OS X v10.3 and later.

Declared in `DVDPlayback.h`.

`kDVDUOPStillOff`

Not used.

Available in Mac OS X v10.3 and later.

Declared in `DVDPlayback.h`.

`kDVDUOPPauseOn`

The function `DVDPause` (page 52) is disabled.

Available in Mac OS X v10.3 and later.

Declared in `DVDPlayback.h`.

`kDVDUOPAudioStreamChange`

The function `DVDSetAudioStream` (page 57) is disabled.

Available in Mac OS X v10.3 and later.

Declared in `DVDPlayback.h`.

`kDVDUOPAngleChange`

The function `DVDSetAngle` (page 56) is disabled.

Available in Mac OS X v10.3 and later.

Declared in `DVDPlayback.h`.

`kDVDUOPKaraokeModeChange`

Not used.

Available in Mac OS X v10.3 and later.

Declared in `DVDPlayback.h`.

`kDVDUOPVideoModeChange`

Not used.

Available in Mac OS X v10.3 and later.

Declared in `DVDPlayback.h`.

`kDVDUOPScanOff`

> Not used.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

`kDVDUOPPauseOff`

> The function `DVDResume` (page 55) is disabled.
>
> Available in Mac OS X v10.3 and later.
>
> Declared in `DVDPlayback.h`.

Discussion

These constants are used to specify one or more disabled user operations in the mask passed as the first value parameter in the `kDVDEventValidUOP` event. See "Event Codes" (page 93).

Declared In

`DVDPlayback.h`

Event Codes

Constants that specify events for which applications can register to get notified.

```
enum {
    kDVDEventTitle              = 1,
    kDVDEventPTT                = 2,
    kDVDEventValidUOP           = 3,
    kDVDEventAngle              = 4,
    kDVDEventAudioStream        = 5,
    kDVDEventSubpictureStream   = 6,
    kDVDEventDisplayMode        = 7,
    kDVDEventDomain             = 8,
    kDVDEventBitrate            = 9,
    kDVDEventStill              = 10,
    kDVDEventPlayback           = 11,
    kDVDEventVideoStandard      = 12,
    kDVDEventStreams            = 13,
    kDVDEventScanSpeed          = 14,
    kDVDEventMenuCalled         = 15,
    kDVDEventParental           = 16,
    kDVDEventPGC                = 17,
    kDVDEventGPRM               = 18,
    kDVDEventRegionMismatch     = 19,
    kDVDEventTitleTime          = 20,
    kDVDEventSubpictureStreamNumbers = 21,
    kDVDEventAudioStreamNumbers = 22,
    kDVDEventAngleNumbers       = 23,
    kDVDEventError              = 24,
    kDVDEventCCInfo             = 25,
    kDVDEventChapterTime        = 26
};
typedef UInt32 DVDEventCode;
```

Discussion

Table 1 lists the constants you use to register a callback function as an event handler for one or more DVD playback events. To learn how to register an event handler, see the function DVDRegisterEventCallBack (page 54).

Unlike Carbon events, you do not need to call a function to retrieve the data associated with a playback event. When your event callback is invoked, the associated data is contained in one or both *event value* parameters. For more information, see `DVDEventCallBackFunctionPtr` (page 70).

Table 1   DVD Playback events

Event ID	Event description	Event value 1	Event value 2
`kDVDEventTitle`	Title has changed.	Title ID	-
`kDVDEventPTT`	Chapter has changed.	Chapter ID	-
`kDVDEventValidUOP`	User operations that are disabled.	`DVDUOPCode` (page 90)	-
`kDVDEventAngle`	Camera angle has changed.	Stream ID	-
`kDVDEventAudioStream`	Audio stream has changed.	Stream ID	-
`kDVDEventSubpicture-Stream`	Subpicture has changed.	Stream ID	isVisible
`kDVDEventDisplayMode`	Display mode has changed.	`DVDAspectRatio` (page 78)	-
`kDVDEventDomain`	Domain has changed.	`DVDDomainCode` (page 88)	-
`kDVDEventBitrate`	Bit rate has changed.	Bits per second	-
`kDVDEventStill`	Still picture on or off.	On (1) - Off (0)	-
`kDVDEventPlayback`	Playback state has changed.	`DVDState` (page 73)	-
`kDVDEventVideoStandard`	Video format has changed	`DVDFormat` (page 79)	-
`kDVDEventStreams`	Stream has changed (audio, subpicture, or angle).	-	-
`kDVDEventScanSpeed`	Scan rate has changed.	Speed (1x, 2x, 3x, and so on)	-
`kDVDEventMenuCalled`	Menu has changed.	`DVDMenu` (page 74)	-
`kDVDEventParental`	Parental level has changed.	Level number	-
`kDVDEventPGC`	Program chain has changed.	PGC number	-
`kDVDEventGPRM`	GPRM data has changed.	GPRM index	Data
`kDVDEventRegion-Mismatch`	Region mismatch between disc and playback device.	Disc region	-
`kDVDEventTitleTime`	Title time has changed.	Elapsed time in milliseconds	Duration in milliseconds

Event ID	Event description	Event value 1	Event value 2
`kDVDEventSubpicture-StreamNumbers`	The number of subpicture streams has changed.	The number of subpicture streams in title	-
`kDVDEventAudio-StreamNumbers`	The number of audio streams has changed.	The number of audio streams in title	-
`kDVDEventAngleNumbers`	The number of angles has changed.	The number of angles in title	-
`kDVDEventError`	Hardware error.	`DVDErrorCode` (page 72)	-
`kDVDEventCCInfo`	Closed caption has changed.	Opcode	Data
`kDVDEventChapterTime`	Chapter time has changed.	Elapsed time in milliseconds	Duration in milliseconds

Availability
Available in Mac OS X version 10.3 and later.

Declared In
`DVDPlayback.h`

Result Codes

The table below lists the result codes defined for DVD Playback.

Result Code	Value	Description
`kDVDErrorUnknown`	-70001	Unspecified error. Available in Mac OS X v10.3 and later.
`kDVDErrorInitializingLib`	-70002	An error occurred while initializing a DVD-Video playback session. DVD Playback Services is probably being used in another process. Available in Mac OS X v10.3 and later.
`kDVDErrorUninitializedLib`	-70003	A DVD-Video playback session has not been initialized. Any function except `DVDInitialize` (page 45) could return this value. Available in Mac OS X v10.3 and later.
`kDVDErrorNotAllowedDuringPlayback`	-70004	The operation is not permitted while DVD-Video media is playing. Available in Mac OS X v10.3 and later.

Result Code	Value	Description
`kDVDErrorUnassignedGrafPort`	-70005	A graphics port for DVD-Video playback has not been set. Available in Mac OS X v10.3 and later.
`kDVDErrorAlreadyPlaying`	-70006	See `kDVDErrorNotAllowedDuringPlayback`. Available in Mac OS X v10.3 and later.
`kDVDErrorNoFatalErrCallBack`	-70007	No callback has been registered to handle unrecoverable errors during playback. Available in Mac OS X v10.3 and later.
`kDVDErrorIsAlreadySleeping`	-70008	`DVDSleep` (page 66) was called twice without an intervening call to `DVDWakeUp` (page 69). Available in Mac OS X v10.3 and later.
`kDVDErrorDontNeedWakeup`	-70009	`DVDWakeUp` (page 69) was called without first calling `DVDSleep` (page 66). Available in Mac OS X v10.3 and later.
`kDVDErrorTimeOutOfRange`	-70010	The time code is outside the valid range for the current title. Available in Mac OS X v10.3 and later.
`kDVDErrorUserActionNoOp`	-70011	The operation is not permitted at this time. Available in Mac OS X v10.3 and later.
`kDVDErrorMissingDrive`	-70012	A DVD drive is not available. Available in Mac OS X v10.3 and later.
`kDVDErrorNotSupportedConfiguration`	-70013	The current system configuration does not support DVD-Video playback. Available in Mac OS X v10.3 and later.
`kDVDErrorNotSupportedFunction`	-70014	DVD Playback Services does not support this operation. Available in Mac OS X v10.3 and later.
`kDVDErrorNoValidMedia`	-70015	A VIDEO_TS media folder is not open for playback. Available in Mac OS X v10.3 and later.
`kDVDErrorWrongParam`	-70016	An argument is invalid. Available in Mac OS X v10.3 and later.
`kDVDErrorMissingGraphicsDevice`	-70017	A graphics device is not available. Available in Mac OS X v10.3 and later.

Result Code	Value	Description
kDVDErrorGraphicsDevice	-70018	A graphics device error was encountered. Available in Mac OS X v10.3 and later.
kDVDErrorPlaybackOpen	-70019	A VIDEO_TS media folder is already open for playback. Available in Mac OS X v10.3 and later.
kDVDErrorInvalidRegionCode	-70020	Region code is not valid. Available in Mac OS X v10.3 and later.
kDVDErrorRgnMgrInstall	-70021	Not used. Available in Mac OS X v10.3 and later.
kDVDErrorMismatchedRegionCode	-70022	DVD-Video disc region code does not match the region code currently assigned to the DVD drive. Available in Mac OS X v10.3 and later.
kDVDErrorNoMoreRegionSets	-70023	Changing the region code assigned to the DVD drive is no longer permitted. Available in Mac OS X v10.3 and later.
kDVDErrordRegionCodeUninitialized	-70024	No region code has been assigned to the DVD drive. Available in Mac OS X v10.3 and later.
kDVDErrorAuthentification	-70025	An attempt to change the DVD drive region code could not be authenticated. Available in Mac OS X v10.3 and later.
kDVDErrorOutOfVideoMemory	-70026	The video driver does not have enough video memory available for playback. Available in Mac OS X v10.3 and later.
kDVDErrorNoAudioOutputDevice	-70027	An appropriate audio output device could not be found. Available in Mac OS X v10.3 and later.
kDVDErrorSystem	-70028	A system error occurred. Available in Mac OS X v10.3 and later.
kDVDErrorNavigation	-70029	The user has made a selection not supported in the current menu. Available in Mac OS X v10.3 and later.

Deprecated DVD Playback Framework Functions

A function identified as deprecated has been superseded and may become unsupported in the future.

Deprecated in Mac OS X v10.5

DVDDoMenuClick

If a menu button contains a specified point, the button's action is executed and the button's index is passed back. (Deprecated in Mac OS X v10.5. Use `DVDDoMenuCGClick` instead.)

```
OSStatus DVDDoMenuClick (
    Point inPortPt,
    SInt32 *outIndex
);
```

Parameters

inPortPt

> A point specified using window local coordinates.

outIndex

> A pointer to an integer. On return, if the specified point lies inside the clickable area of a menu button, the integer contains the 1-based index of the button.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function checks to see if a specified point lies inside one of the buttons in the current menu. If so, this function executes the button's action and passes back its index. Note that the bounding rectangle of a button is not necessarily the same as the clickable area; not all buttons even have a clickable area. See also `DVDDoButtonActivate` (page 18).

Availability

Available in Mac OS X v10.3 and later.

Deprecated in Mac OS X v10.5.

Not available to 64-bit applications.

Related Sample Code

CocoaDVDPlayer

Declared In

`DVDPlayback.h`

DVDDoMenuMouseOver

If a specified point lies inside a menu button, focuses the button and passes back its index. (Deprecated in Mac OS X v10.5. Use `DVDDoMenuCGMouseOver` instead.)

```
OSStatus DVDDoMenuMouseOver (
    Point inPortPt,
    SInt32 *outIndex
);
```

Parameters

inPortPt

A point specified using window local coordinates.

outIndex

A pointer to an integer. On return, if the point lies inside the button in focus, the integer contains the 1-based index of the button.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

See also `DVDDoUserNavigation` (page 19).

Availability

Available in Mac OS X v10.3 and later.

Deprecated in Mac OS X v10.5.

Not available to 64-bit applications.

Related Sample Code

CocoaDVDPlayer

Declared In

`DVDPlayback.h`

DVDGetVideoBounds

Gets the bounds in which to display the video. (Deprecated in Mac OS X v10.5. Use `DVDGetVideoCGBounds` instead.)

```
OSStatus DVDGetVideoBounds (
    Rect *outPortRect
);
```

Parameters

outPortRect

A pointer to a rectangle. On return, the rectangle contains the current bounds of the video area in window local coordinates.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function gets the video bounding rectangle previously set by calling the function `DVDSetVideoBounds` (page 103).

Availability

Available in Mac OS X v10.3 and later.

Deprecated in Mac OS X v10.5.

Not available to 64-bit applications.

Declared In

DVDPlayback.h

DVDGetVideoDevice

Gets the graphics device for the current DVD-Video playback session. (Deprecated in Mac OS X v10.5. Use DVDGetVideoDisplay (page 40) instead.)

```
OSStatus DVDGetVideoDevice (
    GDHandle *outDevice
);
```

Parameters

outDevice

> A pointer to a graphics device handle. On return, the device handle is set to the current video playback device.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

Before calling this function, you need to set the video playback device using DVDSetVideoDevice (page 104) or DVDSwitchToDevice (page 105).

Availability

Available in Mac OS X v10.3 and later.

Deprecated in Mac OS X v10.5.

Not available to 64-bit applications.

Declared In

DVDPlayback.h

DVDGetVideoKeyColor

Gets the key color used by the video driver. (Deprecated in Mac OS X v10.5. There is no replacement function.)

```
OSStatus DVDGetVideoKeyColor (
    RGBColor *outKeyColor
);
```

Parameters

outKeyColor

> A pointer to an RGB color structure. On return, this structure contains the red, green, and blue color values for the color black.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion
This function does nothing and returns the color black.

Availability
Available in Mac OS X v10.3 and later.

Deprecated in Mac OS X v10.5.

Not available to 64-bit applications.

Declared In
DVDPlayback.h

DVDGetVideoPort

Gets the graphics port of the Carbon window in use for DVD-Video playback. (Deprecated in Mac OS X v10.5. Use `DVDGetVideoWindowRef` instead.)

```
OSStatus DVDGetVideoPort (
    CGrafPtr *outVidPort
);
```

Parameters
outVidPort

> A pointer to a QuickDraw graphics port. On return, the graphics port of the Carbon window currently in use for DVD-Video playback, or NULL if there is no Carbon window.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Availability
Available in Mac OS X v10.3 and later.

Deprecated in Mac OS X v10.5.

Not available to 64-bit applications.

Declared In
DVDPlayback.h

DVDIsSupportedDevice

Determines whether a graphics device supports DVD-Video playback. (Deprecated in Mac OS X v10.5. Use `DVDIsSupportedDisplay` (page 49) instead.)

```
OSStatus DVDIsSupportedDevice (
    GDHandle inDevice,
    Boolean *outSupported
);
```

Parameters
inDevice

> A handle to the GDevice structure for the graphics device you want to use for video playback.

outSupported

> A pointer to a Boolean variable. On return, a value of `TRUE` indicates that the specified graphics device can be used for DVD-Video playback. A value of `FALSE` indicates the device cannot be used for this purpose. If the value is `FALSE`, the device may require a different video driver than the one currently in use.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

For this function to work properly, you must first call the function `DVDSetVideoPort` (page 104) with a valid graphics port.

Availability

Available in Mac OS X v10.3 and later.

Deprecated in Mac OS X v10.5.

Not available to 64-bit applications.

Declared In

`DVDPlayback.h`

DVDSetVideoBounds

Sets the bounds in which to display the video. (Deprecated in Mac OS X v10.5. Use `DVDSetVideoCGBounds` instead.)

```
OSStatus DVDSetVideoBounds (
    Rect *inPortRect
);
```

Parameters

inPortRect

> A pointer to a rectangle that contains the desired bounds in window local coordinates.

Return Value

A result code. See "DVD Playback Result Codes" (page 95).

Discussion

This function is used to set the area inside the current window in which to display the video. The video area is not required to fill the entire window. Generally you should set the video area to be smaller than the window whenever the aspect ratio of the current title and window are different. To find the aspect ratio of the current title, use the function `DVDGetAspectRatio` (page 20).

Because the aspect ratios of the titles in a DVD-Video media folder are not always the same, you may need to call this function repeatedly to reset the video area as the user makes different viewing choices.

Availability

Available in Mac OS X v10.3 and later.

Deprecated in Mac OS X v10.5.

Not available to 64-bit applications.

Related Sample Code

CocoaDVDPlayer

Declared In
`DVDPlayback.h`

DVDSetVideoDevice

Sets the graphics device for the current DVD-Video playback session. (Deprecated in Mac OS X v10.5. Use `DVDSetVideoDisplay` (page 65) instead.)

```
OSStatus DVDSetVideoDevice (
    GDHandle inDevice
);
```

Parameters

inDevice
> A handle to the `GDevice` structure for the graphics device you want to use for video playback.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
You need to call this function or `DVDSetVideoDisplay` (page 65) each time you move the video playback window to a new graphics device. To avoid degrading video performance, you should not attempt to draw a video playback window that spans two different devices.

Availability
Available in Mac OS X v10.3 and later.

Deprecated in Mac OS X v10.5.

Not available to 64-bit applications.

Declared In
`DVDPlayback.h`

DVDSetVideoPort

Enables or disables DVD-Video playback in a Carbon window. (Deprecated in Mac OS X v10.5. Use `DVDSetVideoWindowRef` instead.)

```
OSStatus DVDSetVideoPort (
    CGrafPtr inVidPort
);
```

Parameters

inVidPort
> The QuickDraw graphics port associated with a Carbon window, or `NULL`. For information about when to pass `NULL`, see the discussion below.

Return Value
A result code. See "DVD Playback Result Codes" (page 95).

Discussion
If you're using a Carbon window for DVD-Video playback, you need to use this function in two situations:

■ After calling the function `DVDInitialize` (page 45) to start a new DVD-Video playback session, you should call this function and pass the graphics port for the window.

■ When video is not playing and you want to draw into the area of the window you set by calling the function `DVDSetVideoBounds` (page 103), you should call this function and pass `NULL` in the `inVidPort` parameter. When you are finished drawing into the video area of the window and you are ready to resume video playback, you should call this function again and pass the window port in the `inVidPort` parameter.

Availability

Available in Mac OS X v10.3 and later.

Deprecated in Mac OS X v10.5.

Not available to 64-bit applications.

Declared In

`DVDPlayback.h`

DVDSwitchToDevice

If a graphics device supports DVD-Video playback, sets the device for the current playback session. (Deprecated in Mac OS X v10.5. Use `DVDSwitchToDisplay` (page 68) instead.)

```
OSStatus DVDSwitchToDevice (
    GDHandle newDevice,
    Boolean *outSupported
);
```

Parameters

newDevice

A handle to the `GDevice` structure for the graphics device you want to use for video playback.

outSupported

A pointer to a Boolean variable. On return, a value of `TRUE` indicates that the specified graphics device can be used for DVD-Video playback. A value of `FALSE` indicates the device cannot be used for this purpose. If the value is `FALSE`, the current video playback device remains unchanged.

Return Value

A result code. If the specified device is not supported, this function returns an error and maintains the current device. See "DVD Playback Result Codes" (page 95).

Discussion

This convenience function is equivalent to calling the function `DVDIsSupportedDevice` (page 102) followed by a call to `DVDSetVideoDevice` (page 104). This function works correctly even if there is no current video playback device.

Availability

Available in Mac OS X v10.3 and later.

Deprecated in Mac OS X v10.5.

Not available to 64-bit applications.

Declared In

`DVDPlayback.h`

Document Revision History

This table describes the changes to *DVD Playback Framework Reference*.

Date	Notes
2007-06-28	Added deprecation information. Made format and editorial changes.
2006-07-24	Changed the title from "DVD Playback Services Framework Reference."
2005-08-11	Major content update.
2005-04-29	First public release of this document. While technically accurate, this version is preliminary and not yet complete.

Index

L

Language Codes 81

P

Playback Menus 74
Playback States 73

R

Region Codes 87

S

Scan Directions 77
Scan Rates 78
Subpicture Extension Codes 85

T

Time Codes 76

U

UOP Codes 90
User Navigation Directions 75

V

Video Broadcast Formats 79