iTunes COM Interface Documentation

8.1.0.52

Note that calls to the iTunes COM interface will be suspended when a dialog is being displayed in iTunes.

iTunes COM Interface Changes

Classes

iTunesApp

Interfaces

liTunes

IITObject

IITOperationStatus

IITConvertOperationStatus

IITSource

IITSourceCollection

IITIPodSource

IITPlaylist

IITPlaylistCollection

IITAudioCDPlaylist

IITLibraryPlaylist

IITUserPlaylist

IITTrack

IITTrackCollection

IITFileOrCDTrack

IITURLTrack

IITArtwork

IITArtworkCollection

IITEncoder

IITEncoderCollection

IITEQPreset

IITEQPresetCollection

IITVisual

IITVisualCollection

IITWindow

IITWindowCollection

IITBrowserWindow

IITPlaylistWindow

Outgoing Interfaces

_liTunesEvents

_IITConvertOperationStatusEvents

Enumerations

ITArtworkFormat ITSourceKind

ITErrors ITPlaylistRepeatMode

ITEvent ITTrackKind

ITPlayerState ITPlayButtonState

ITPlayerButton ITPlayerButtonModifierKey

ITVersion ITPlaylistKind

ITWindowKind

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

iTunesAppCOM.idl File Reference

Classes

interface	liTunes
	Defines the top-level iTunes application object. More
interface	IITOperationStatus
	Represents the status of an asynchronous add or convert operation. More
interface	IITConvertOperationStatus
	Represents the status of an asynchronous convert operation. <u>More</u>
interface	_liTunesEvents
	Defines the outgoing event interface of the liTunes interface. <u>More</u>
interface	_IITConvertOperationStatusEvents
	Defines the outgoing event interface of the IITConvertOperationStatus interface. <u>More</u>
class	iTunesApp
	Represents the iTunes application. <u>More</u>

Enumerations

```
enum
       ITPlayerState {
        ITPlayerStateStopped = 0,
        ITPlayerStatePlaying,
        ITPlayerStateFastForward,
        ITPlayerStateRewind
       Specifies the state of the player. More...
       ITVisualSize {
enum
        ITVisualSizeSmall = 0,
        ITVisualSizeMedium,
        ITVisualSizeLarge
       Specifies the size of the visuals. More...
       ITCOMDisabledReason {
enum
        ITCOMDisabledReasonOther = 0,
        ITCOMDisabledReasonDialog.
        ITCOMDisabledReasonQuitting
       Specifies the reason the COM interface is being disabled. More...
enum
       ITPlayButtonState {
        ITPlayButtonStatePlayDisabled = 0,
        ITPlayButtonStatePlayEnabled,
        ITPlayButtonStatePauseEnabled,
        ITPlayButtonStatePauseDisabled,
        ITPlayButtonStateStopEnabled,
        ITPlayButtonStateStopDisabled
       }
       Specifies the state of the play/pause/stop button. More...
       ITPlayerButton {
enum
        ITPlayerButtonPrevious = 0,
        ITPlayerButtonPlay,
        ITPlayerButtonNext
       }
       Specifies a player button. More...
       ITPlayerButtonModifierKey {
enum
        ITPlayerButtonModifierKeyNone = 0,
        ITPlayerButtonModifierKeyShift = 1,
        ITPlayerButtonModifierKeyControl = 2,
        ITPlayerButtonModifierKeyAlt = 4,
        ITPlayerButtonModifierKeyCapsLock = 8
```

```
Player button modifier key flags. More...
       ITEvent {
enum
        ITEventDatabaseChanged = 1,
        ITEventPlayerPlay = 2,
        ITEventPlayerStop = 3,
        ITEventPlayerPlayingTrackChanged = 4,
        ITEventCOMCallsDisabled = 6,
        ITEventCOMCallsEnabled = 7,
        ITEventQuitting = 8,
        ITEventAboutToPromptUserToQuit = 9,
        ITEventSoundVolumeChanged = 10
       }
       Events generated by liTunes. <u>More...</u>
      ITConvertOperationStatusEvent {
enum
        ITConvertOperationStatusChanged = 1,
        ITConvertOperationComplete = 2
       }
       Events generated by IITConvertOperationStatus. More...
```

Enumeration Type Documentation

enum ITCOMDisabledReason

Specifies the reason the COM interface is being disabled.

Enumeration values:

ITCOMDisabledReasonOther (0) COM interface is being

disabled for some other reason.

ITCOMDisabledReasonDialog (1) COM interface is being

disabled because a modal dialog

is being displayed.

ITCOMDisabledReasonQuitting (2) COM interface is being

disabled because iTunes is

quitting.

enum ITConvertOperationStatusEvent

Events generated by **IITConvertOperationStatus**.

Enumeration values:

ITConvertOperationStatusChanged (1) status about the

conversion operation has

changed

ITConvertOperationComplete (2) the conversion operation

has completed

enum ITEvent

Events generated by liTunes.

Enumeration values:

ITEventDatabaseChanged (1) iTunes database has

changed

ITEventPlayerPlay (2) a track has started

playing

ITEventPlayerStop (3) a track has stopped

playing

ITEventPlayerPlayingTrackChanged (4) information about the

currently playing track has

changed

ITEventCOMCallsDisabled (6) calls to the iTunes COM

interface will be deferred (typically because a modal

dialog is displayed)

ITEventCOMCallsEnabled (7) calls to the iTunes COM

interface will no longer be deferred (typically because the last modal dialog has

been dismissed)

ITEventQuitting (8) iTunes is about to quit,

existing iTunes COM objects

will no longer be valid

ITEventAboutToPromptUserToQuit (9) iTunes is about to prompt

the user to quit, you should release iTunes COM objects

if you want to avoid the

dialog

ITEventSoundVolumeChanged (10) the sound output

volume has changed

enum ITPlayButtonState

Specifies the state of the play/pause/stop button.

Enumeration values:

ITPlayButtonStatePlayDisabled

(0) Disabled play button.

ITPlayButtonStatePlayEnabled

(1) Enabled play button.

ITPlayButtonStatePauseEnabled

(2) Enabled pause button.

ITPlayButtonStatePauseDisabled (3) Disabled pause button.

ITPlayButtonStateStopEnabled

(4) Enabled stop button.

ITPlayButtonStateStopDisabled

(5) Disabled stop button.

enum ITPlayerButton

Specifies a player button.

Enumeration values:

ITPlayerButtonPrevious (0) Previous/rewind button.

ITPlayerButtonPlay

(1) Play button.

ITPlayerButtonNext

(2) Next/fast forward button.

enum ITPlayerButtonModifierKey

Player button modifier key flags.

Enumeration values:

ITPlayerButtonModifierKeyNone

(0) No modifiers.

ITPlayerButtonModifierKeyShift

(1) Shift key down.

ITPlayerButtonModifierKeyControl

(2) Control key down.

ITP layer Button Modifier Key Alt

(4) Alt key down.

ITPlayerButtonModifierKeyCapsLock (8) Caps Lock key toggled.

enum ITPlayerState

Specifies the state of the player.

Enumeration values:

ITPlayerStateStopped (0) Player is stopped.ITPlayerStatePlaying (1) Player is playing.

ITPlayerStateFastForward (2) Player is fast forwarding.

ITPlayerStateRewind (3) Player is rewinding.

enum ITVisualSize

Specifies the size of the visuals.

Enumeration values:

ITVisualSizeSmall (0) Small size visuals.

ITVisualSizeMedium (1) Medium size visuals.

ITVisualSizeLarge (2) Large size visuals.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

iTunesArtworkCOM.idl File Reference

Classes

interface	IITArtwork
	Defines a single piece of artwork. More
interface	IITArtworkCollection
	Represents a collection of iTunes artwork objects. More

Enumerations

```
enum ITArtworkFormat {
    ITArtworkFormatUnknown = 0,
    ITArtworkFormatJPEG,
    ITArtworkFormatPNG,
    ITArtworkFormatBMP
    }
    Specifies the artwork format. More...
```

Enumeration Type Documentation

enum ITArtworkFormat

Specifies the artwork format.

Enumeration values:

ITArtworkFormatUnknown (0) Unknown artwork format.

ITArtworkFormatJPEG (1) JPEG image.
ITArtworkFormatPNG (2) PNG image.

ITArtworkFormatBMP (3) BMP image.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 Apple Computer, Inc.

iTunesCOMInterface.idl File Reference

Namespaces

namespace iTunesLib

Enumerations

```
enum ITVersion {
    kITTypeLibrary_MajorVersion = 1,
    kITTypeLibrary_MinorVersion = 12
  }

enum ITErrors {
    ITUNES_E_USERCANCEL = 0xA0040201,
    ITUNES_E_OBJECTDELETED = 0xA0040202,
    ITUNES_E_OBJECTLOCKED = 0xA0040203,
    ITUNES_E_CONVERSIONINPROGRESS = 0xA0040204,
    ITUNES_E_MUSICSTOREDISABLED = 0xA0040205,
    ITUNES_E_OBJECTEXISTS = 0xA0040206,
    ITUNES_E_OBJECTEXISTS = 0xA0040207
    }

iTunes-specific HRESULT error codes. More...
```

Functions

importlib ("stdole2.tlb")

Enumeration Type Documentation

enum ITErrors

iTunes-specific HRESULT error codes.

Enumeration values:

ITUNES_E_USERCANCEL (0xA0040201) user

canceled the operation

Note that Microsoft

defines

ERROR_CANCELLED, but this is a positive value. We need to use a negative value to force this to be treated

as an exception.

ITUNES_E_OBJECTDELETED (0xA0040202) the

entity referenced by this COM object has

been deleted

ITUNES_E_OBJECTLOCKED (0xA0040203) attempt

to modify a locked

property

ITUNES_E_CONVERSIONINPROGRESS (0xA0040204) attempt

to start a conversion while a previous conversion is still in

progress

ITUNES_E_MUSICSTOREDISABLED (0xA0040205) access

to the iTunes Store is

disabled in the

ITUNES E OBJECTEXISTS

preferences (0xA0040206) an object with the same name already exists (added in iTunes type library 1.2)

ITUNES E PODCASTSDISABLED

(0xA0040207) access to podcasts is disabled in the preferences (added in iTunes type library 1.5)

enum ITVersion

The major and minor version of the iTunes type library. Use these constants with CheckVersion().

Enumeration values:

kITTypeLibrary_MajorVersion (1) iTunes type library major version kITTypeLibrary MinorVersion (4) iTunes type library minor version (0 for iTunes 4.5, 1 for iTunes 4.6, 2 for iTunes 4.7 and 4.7.1, 3 for iTunes 4.8 and 4.8.1, 4 for iTunes 4.9, 5 for iTunes 5.0, 6 for iTunes 6.0, 7 for iTunes 6.0.2, 8 for iTunes 7.0, 9 for iTunes 7.1, 10 for iTunes 7.4, 11 for iTunes 7.7, 12 for iTunes 8.1)

Function Documentation

importlib ("stdole2.tlb")

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

iTunesEncoderCOM.idl File Reference

Classes

interface	IITEncoder
	Represents an iTunes encoder. More
interface	IITEncoderCollection
	Represents a collection of encoder objects. More

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

iTunesEQPresetCOM.idl File Reference

Classes

interface	IITEQPreset
	Represents an equalizer preset. More
interface	IITEQPresetCollection
	Represents a collection of EQ preset objects. More

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

iTunesObjectCOM.idl File Reference

Classes

interface IITObject

Defines a source, playlist or track. More...

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

iTunesPlaylistCOM.idl File Reference

Classes

interface	IITPlaylist
	Represents a playlist. <u>More</u>
interface	IITLibraryPlaylist
	Represents a library playlist. <u>More</u>
interface	IITUserPlaylist
	Represents a user-defined playlist. More
interface	IITAudioCDPlaylist
	Represents an audio CD playlist. More
interface	IITPlaylistCollection
	Represents a collection of playlist objects. More

Enumerations

```
enum
       ITPlaylistKind {
        ITPlaylistKindUnknown = 0,
        ITPlaylistKindLibrary,
        ITPlaylistKindUser.
        ITPlaylistKindCD.
        ITPlaylistKindDevice.
        ITPlaylistKindRadioTuner
       }
       Specifies the playlist kind. More...
       ITPlaylistRepeatMode {
enum
        ITPlaylistRepeatModeOff = 0.
        ITPlaylistRepeatModeOne.
        ITPlaylistRepeatModeAll
        Specifies the playlist playback repeat mode. More...
       ITPlaylistPrintKind {
enum
        ITPlaylistPrintKindPlaylist = 0,
        ITPlaylistPrintKindAlbumlist,
         ITPlaylistPrintKindInsert
        Specifies the kind of playlist printout. More...
enum
       ITPlaylistSearchField {
        ITPlaylistSearchFieldAll = 0,
        ITPlaylistSearchFieldVisible,
        ITPlaylistSearchFieldArtists,
        ITPlaylistSearchFieldAlbums
        ITPlaylistSearchFieldComposers,
        ITPlaylistSearchFieldSongNames
       Specifies the fields in each track that will be searched by IITPlaylist::Search(). More...
       ITUserPlaylistSpecialKind {
enum
        ITUserPlaylistSpecialKindNone = 0,
        ITUserPlaylistSpecialKindPurchasedMusic,
        ITUserPlaylistSpecialKindPartyShuffle,
        ITUserPlaylistSpecialKindPodcasts.
        ITUserPlaylistSpecialKindFolder,
        ITUserPlaylistSpecialKindVideos,
        ITUserPlaylistSpecialKindMusic,
        ITUserPlaylistSpecialKindMovies,
        ITUserPlaylistSpecialKindTVShows.
        ITUserPlaylistSpecialKindAudiobooks
```

Specifies the user playlist special kind. More...

Enumeration Type Documentation

enum ITPlaylistKind

Specifies the playlist kind.

Enumeration values:

ITPlaylistKindUnknown (0) Unknown playlist kind.

ITPlaylistKindLibrary (1) Library playlist (IITLibraryPlaylist).

ITPlaylistKindUser (2) User playlist (IITUserPlaylist).

ITPlaylistKindCD (3) CD playlist (IITAudioCDPlaylist).

ITPlaylistKindDevice (4) Device playlist.

ITPlaylistKindRadioTuner (5) Radio tuner playlist.

enum ITPlaylistPrintKind

Specifies the kind of playlist printout.

Enumeration values:

ITPlaylistPrintKindPlaylist (0) Print a list of tracks in the playlist.

ITPlaylistPrintKindAlbumlist (1) Print a list of albums in the

playlist.

ITPlaylistPrintKindInsert (2) Print a CD jewel case insert.

enum ITPlaylistRepeatMode

Specifies the playlist playback repeat mode.

Enumeration values:

ITPlaylistRepeatModeOff (0) Play playlist once.

ITPlaylistRepeatModeOne (1) Repeat song.

ITPlaylistRepeatModeAll (2) Repeat playlist.

enum ITPlaylistSearchField

Specifies the fields in each track that will be searched by **IITPlaylist::Search()**.

Enumeration values:

ITPlaylistSearchFieldAll (0) Search all fields of each

track.

ITPlaylistSearchFieldVisible (1) Search only the fields with

columns that are currently visible in the display for the

playlist.

Note that song name, artist, album, and composer will always be searched, even if these columns are not visible.

ITPlaylistSearchFieldArtists (2) Search only the artist field

of each track (IITTrack::Artist).

ITPlaylistSearchFieldAlbums (3) Search only the album field

of each track

(IITTrack::Album).

ITPlaylistSearchFieldComposers (4) Search only the composer

field of each track

(IITTrack::Composer).

ITPlaylistSearchFieldSongNames (5) Search only the song name

field of each track (IITTrack::Name).

enum ITUserPlaylistSpecialKind

Specifies the user playlist special kind.

Enumeration values:

u	meration values.	
	ITUserPlaylistSpecialKindNone	(0) No special kind.
	ITU ser Play list Special Kind Purchased Music	(1) Purchased Music playlist.
	ITUserPlaylistSpecialKindPartyShuffle	(2) Party Shuffle playlist.
	ITUserPlaylistSpecialKindPodcasts	(3) Podcasts playlist.
	ITUserPlaylistSpecialKindFolder	(4) Folder playlist.
	ITUserPlaylistSpecialKindVideos	(5) Videos playlist (added in iTunes type library 1.6).
	ITUserPlaylistSpecialKindMusic	(6) Music playlist (added in iTunes type library 1.8).
	ITUserPlaylistSpecialKindMovies	(7) Movies playlist (added in iTunes type library 1.8).
	ITUserPlaylistSpecialKindTVShows	(8) TV Shows playlist (added in iTunes type library 1.8).
	ITUserPlaylistSpecialKindAudiobooks	(9) Audiobooks playlist (added in iTunes type library 1.8).

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

iTunesSourceCOM.idl File Reference

Classes

interface	IITSource
	Represents an entry in the Source list (music library, CD, device, etc.). More
interface	IITIPodSource
	Represents an iPod source. <u>More</u>
interface	IITSourceCollection
	Represents a collection of source objects. <u>More</u>

Enumerations

```
enum

ITSourceKind {
    ITSourceKindUnknown = 0,
    ITSourceKindLibrary,
    ITSourceKindIPod,
    ITSourceKindAudioCD,
    ITSourceKindMP3CD,
    ITSourceKindDevice,
    ITSourceKindRadioTuner,
    ITSourceKindSharedLibrary
    }

Specifies the source kind. More...
```

Enumeration Type Documentation

enum ITSourceKind

Specifies the source kind.

Enumeration values:

ITSourceKindUnknown (0) Unknown source kind.

ITSourceKindLibrary (1) Library source.

ITSourceKindIPod (2) iPod source (IITIPodSource).

ITSourceKindAudioCD (3) Audio CD source. ITSourceKindMP3CD (4) MP3 CD source.

ITSourceKindDevice (5) Device source.

ITSourceKindRadioTuner (6) Radio tuner source.

ITSourceKindSharedLibrary (7) Shared library source.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

iTunesTrackCOM.idl File Reference

Classes

interface	IITTrack
	Represents a track. More
interface	IITFileOrCDTrack
	Represents a file or CD track. More
interface	IITURLTrack
interface	IITURLTrack Represents a URL track. <u>More</u>
interface	

Enumerations

```
ITTrackKind {
enum
        ITTrackKindUnknown = 0,
        ITTrackKindFile,
        ITTrackKindCD,
        ITTrackKindURL,
        ITTrackKindDevice.
        ITTrackKindSharedLibrary
       }
       Specifies the track kind. More...
enum ITVideoKind {
        ITVideoKindNone = 0.
        ITVideoKindMovie,
        ITVideoKindMusicVideo,
        ITVideoKindTVShow
       }
       Specifies the video track kind (added in iTunes type library 1.8). More...
enum
       ITRatingKind {
        ITRatingKindUser = 0,
        ITRatingKindComputed
       Specifies the rating kind (added in iTunes type library 1.10). More...
```

Enumeration Type Documentation

enum ITRatingKind

Specifies the rating kind (added in iTunes type library 1.10).

Enumeration values:

ITRatingKindUser (0) User-specified rating.
ITRatingKindComputed (1) iTunes-computed rating.

enum ITTrackKind

Specifies the track kind.

Enumeration values:

ITTrackKindUnknown (0) Unknown track kind.

ITTrackKindFile (1) File track (IITFileOrCDTrack).
ITTrackKindCD (2) CD track (IITFileOrCDTrack).

ITTrackKindURL (3) URL track (IITURLTrack).

ITTrackKindDevice (4) Device track.

ITTrackKindSharedLibrary (5) Shared library track.

enum ITVideoKind

Specifies the video track kind (added in iTunes type library 1.8).

Enumeration values:

ITVideoKindNone (0) Not a video track, or unknown video

track kind.

ITVideoKindMovie (1) Movie video track.ITVideoKindMusicVideo (2) Music video track.ITVideoKindTVShow (3) TV show video track.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

iTunesVisualCOM.idl File Reference

Classes

interface	IITVisual
	Represents an iTunes visual plug-in. More
interface	IITVisualCollection
	Represents a collection of visual plug-in objects. More

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

iTunesWindowCOM.idl File Reference

Classes

interface	IITWindow
	Represents an iTunes window. More
interface	IITBrowserWindow
	Represents the main browser window. More
interface	IITPlaylistWindow
interface	IITPlaylistWindow Represents a playlist window. More
interface	•

Enumerations

```
enum

ITWindowKind {
    ITWindowKindUnknown = 0,
    ITWindowKindBrowser,
    ITWindowKindPlaylist,
    ITWindowKindEQ,
    ITWindowKindArtwork,
    ITWindowKindNowPlaying
    }
    Specifies the window kind. More...
```

Enumeration Type Documentation

enum ITWindowKind

Specifies the window kind.

Enumeration values:

ITWindowKindUnknown (0) Unknown window kind.ITWindowKindBrowser (1) Main browser window

(IITBrowserWindow).

ITWindowKindPlaylist (2) Playlist window

(IITPlaylistWindow).

ITWindowKindEQ (3) EQ window.

ITWindowKindArtwork (4) Artwork window.

ITWindowKindNowPlaying (5) Now Playing window.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

_IITConvertOperationStatusEvents Interface Reference

List of all members.

Detailed Description

Defines the outgoing event interface of the **IITConvertOperationStatus** interface.

Note:

This interface is available in iTunes 4.6 and later (iTunes type library 1.1 and later).

Public Member Functions

HRESULT	OnConvertOperationStatusChangedEvent ([in] BSTR trackName,[in] long progressValue,[in] long maxProgressValue)
	The ITConvertOperationStatusChanged event is fired when status about the conversion changes.
HRESULT	OnConvertOperationCompleteEvent ()
	Onconvertoperation complete Event ()

Member Function Documentation

HRESULT _IITConvertOperationStatusEvents::OnConvertOperation

The **ITConvertOperationComplete** event is fired when the conversion operation has completed. The operation is completed after all tracks have been converted, or when the conversion is canceled.

Note:

This event is available in iTunes 4.8 and later (iTunes type library 1.3 and later).

HRESULT IITConvertOperationStatusEvents::OnConvertOperation

The **ITConvertOperationStatusChanged** event is fired when status about the conversion changes.

Parameters:

trackName The name of the track currently being

converted. May be NULL if no track is

currently being converted.

progress Value The conversion progress value for the track

currently being converted. This number will

always be in the range 0 through

maxProgressValue.

maxProgressValue The maximum conversion progress value for

the track currently being converted. The conversion of the current track is complete

when progressValue reaches

maxProgressValue. Note that maxProgressValue will be different for each track being converted.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

_liTunesEvents Interface Reference

List of all members.

Detailed Description

Defines the outgoing event interface of the **liTunes** interface.

Public Member Functions

HRESULT	OnDatabaseChangedEvent ([in] VARIANT deletedObjectIDs,[in] VARIANT changedObjectIDs)
	The ITEventDatabaseChanged event is fired when the iTunes database is changed.
HRESULT	OnPlayerPlayEvent ([in] VARIANT iTrack)
	The ITEventPlayerPlay event is fired when a track begins playing.
HRESULT	OnPlayerStopEvent ([in] VARIANT iTrack)
	The ITEventPlayerStop event is fired when a track stops playing.
HRESULT	OnPlayerPlayingTrackChangedEvent ([in] VARIANT iTrack)
	The ITEventPlayerPlayingTrackChanged event is fired when information about the currently playing track has changed.
HRESULT	OnCOMCallsDisabledEvent ([in] ITCOMDisabledReason reason)
	The ITEventCOMCallsDisabled event is fired when calls to the iTunes COM interface will be deferred.
HRESULT	OnCOMCallsEnabledEvent ()
	The ITEventCOMCallsEnabled event is fired when calls to the iTunes COM interface will no longer be deferred.
HRESULT	OnQuittingEvent ()
	The ITEventQuitting event is fired when iTunes is about to quit.
HRESULT	OnAboutToPromptUserToQuitEvent ()
	The ITEventAboutToPromptUserToQuit event is fired when iTunes is about prompt the user to quit.
HRESULT	OnSoundVolumeChangedEvent ([in] long newVolume)
	The ITEventSoundVolumeChanged event is fired when the sound output volume has changed.

Member Function Documentation

HRESULT _liTunesEvents::OnAboutToPromptUserToQuitEvent()

The ITEventAboutToPromptUserToQuit event is fired when iTunes is about prompt the user to quit.

This event gives clients the opportunity to prevent the warning dialog prompt from occurring.

If the user attempts to quit iTunes while a client still has outstanding iTunes COM objects instantiated, iTunes will display a warning dialog. This event is fired just before the warning dialog is shown. iTunes will then wait up to 5 seconds for clients to release any outstanding iTunes COM objects. If all objects are released during this time, the warning dialog will not be shown and iTunes will quit immediately.

Otherwise, the warning dialog will be shown. If the user chooses to quit iTunes anyway, the ITEventQuitting event is fired. See _liTunesEvents::OnQuittingEvent() for more details.

Note:

This event is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

HRESULT _liTunesEvents::OnCOMCallsDisabledEvent ([in] ITCON

The **ITEventCOMCallsDisabled** event is fired when calls to the iTunes COM interface will be deferred.

Typically, iTunes will defer COM calls when any modal dialog is being displayed. When the user dismisses the last modal dialog, COM calls will be enabled again, and any deferred COM calls will be executed.

You can use this event to avoid making a COM call which will be deferred.

Note:

This event is available in iTunes 4.6 and later (iTunes type library 1.1 and later).

Parameters:

reason The reason the COM interface is being disabled. This is typically ITCOMDisabledReasonDialog.

HRESULT _liTunesEvents::OnCOMCallsEnabledEvent()

The **ITEventCOMCallsEnabled** event is fired when calls to the iTunes COM interface will no longer be deferred.

Typically, iTunes will defer COM calls when any modal dialog is being displayed. When the user dismisses the last modal dialog, COM calls will be enabled again, and any deferred COM calls will be executed.

Note:

This event is available in iTunes 4.6 and later (iTunes type library 1.1 and later).

HRESULT _liTunesEvents::OnDatabaseChangedEvent ([in] VARIA [in] VARIA)

The ITEventDatabaseChanged event is fired when the iTunes database

Each parameter is a two-dimensional SAFEARRAY of VARIANTS, where VT_I4. The first dimension is the number of objects. The second dimens IITObject IDs, where index 0 is the source ID, index 1 is the playlist ID, i

track database ID. For more information on object IDs, see **IITObject**.

Note that you can use **liTunes::GetITObjectByID()** to retrieve changed they no longer exist).

Even a simple change to the database can result in multiple changed obplaylist containing a single track, deletedObjectIDs will contain both the public will contain at a minimum the main library source, as well as any playlist the track in each playlist).

You should avoid doing long operations during processing of this event (since the iTunes user interface thread will be blocked until this call return

Example JScript code:

```
iTunesApp = WScript.CreateObject("iTunes.Application")
var
var
        numDBChanges = 0;
        maxDBChanges = 3;
var
function ITEventTest_OnDatabaseChangedEvent(deletedObjects, cl
            numObjects;
    var
    var
            sourceID, playlistID, trackID, databaseID;
    var
    numDBChanges++;
    WScript.Echo("Database change " + numDBChanges);
    if (deletedObjects != undefined)
    {
        // calculate the number of objects from the deletedObj
        numObjects = (deletedObjects.ubound() - deletedObjects
        WScript.Echo(numObjects + " deleted objects");
        // enumerate the objects that were deleted
        for (i = deletedObjects.lbound(); i <= deletedObjects.</pre>
        {
            sourceID = deletedObjects.getItem(i, 0);
            playlistID = deletedObjects.getItem(i, 1);
            trackID = deletedObjects.getItem(i, 2);
            databaseID = deletedObjects.getItem(i, 3);
            // note - since the object has been deleted, we ca
            // using iTunesApp.GetITObjectByID()
```

```
WScript.Echo(" sourceID " + sourceID +
                " playlistID " + playlistID +
                " trackID " + trackID +
                " databaseID " + databaseID);
        }
    }
    if (changedObjects != undefined)
        // calculate the number of objects from the changedOb-
        numObjects = (changedObjects.ubound() - changedObjects
        WScript.Echo(numObjects + " changed objects");
        // enumerate the objects that were added or changed
        for (i = changedObjects.lbound(); i <= changedObjects.</pre>
        {
            sourceID = changedObjects.getItem(i, 0);
            playlistID = changedObjects.getItem(i, 1);
            trackID = changedObjects.getItem(i, 2);
            databaseID = changedObjects.getItem(i, 3);
            var changedObject = iTunesApp.GetITObjectByID(sour
            WScript.Echo(" (" + changedObject.Name + ") source
                " playlistID " + playlistID +
                " trackID " + trackID +
                " databaseID " + databaseID);
        }
    }
}
// attach event sink
WScript.ConnectObject(iTunesApp, "ITEventTest_");
WScript.Echo("Waiting for database changes...");
while (numDBChanges < maxDBChanges)</pre>
{
    WScript.Sleep(1000);
}
// disconnect event sink
WScript.DisconnectObject(iTunesApp);
WScript.Echo("Database changed " + numDBChanges + " times, exi
```

Parameters:

deletedObjectIDs A two-dimensional array specifying the objects changedObjectIDs A two-dimensional array specifying the objects

HRESULT _liTunesEvents::OnPlayerPlayEvent ([in] VARIANT iTra

The ITEventPlayerPlay event is fired when a track begins playing.

When iTunes switches to playing another track, you will received an ITEventPlayerStop event followed by an ITEventPlayerPlay event, unless it is playing joined CD tracks (see IiTunesEvents::OnPlayerPlayingTrackChangedEvent).

Parameters:

iTrack An IITTrack object (a VARIANT of type VT_DISPATCH) corresponding to the track that has started playing.

HRESULT _liTunesEvents::OnPlayerPlayingTrackChangedEvent([

The ITEventPlayerPlayingTrackChanged event is fired when information about the currently playing track has changed.

This event is fired when the user changes information about the currently playing track (e.g. the name of the track).

This event is also fired when iTunes plays the next joined CD track in a CD playlist, since joined CD tracks are treated as a single track.

Parameters:

iTrack An **IITTrack** object (a VARIANT of type VT_DISPATCH) corresponding to the track that is now playing.

HRESULT _liTunesEvents::OnPlayerStopEvent ([in] VARIANT iTra

The **ITEventPlayerStop** event is fired when a track stops playing.

When iTunes switches to playing another track, you will received an ITEventPlayerStop event followed by an ITEventPlayerPlay event, unless it is playing joined CD tracks (see

_liTunesEvents::OnPlayerPlayingTrackChangedEvent).

Parameters:

iTrack An IITTrack object (a VARIANT of type VT_DISPATCH) corresponding to the track that has stopped playing.

HRESULT _liTunesEvents::OnQuittingEvent()

The **ITEventQuitting** event is fired when iTunes is about to quit.

If the user attempts to quit iTunes while a client still has outstanding iTunes COM objects instantiated, iTunes will display a warning dialog. The user can still choose to quit iTunes anyway, in which case this event will be fired. After this event is fired, any existing iTunes COM objects will no longer be valid.

This event is only used to notify clients that iTunes is quitting, clients cannot prevent this from happening.

Note:

This event is available in iTunes 4.6 and later (iTunes type library 1.1 and later).

HRESULT _liTunesEvents::OnSoundVolumeChangedEvent ([in] lo

The ITEventSoundVolumeChanged event is fired when the sound

output volume has changed.

Note:

This event is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Parameters:

newVolume The new sound output volume (0 = minimum, 100 = maximum).

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITArtwork Interface Reference

List of all members.

Detailed Description

Defines a single piece of artwork.

Artwork is always associated with an individual track.

To add a piece of artwork to a track, use **IITTrack::AddArtworkFromFile()**. The **IITTrack::Artwork** property returns the collection of artwork for the track.

Public Member Functions

Methods

HRESULT	Delete ()
	Delete this piece of artwork from the track.
HRESULT	SetArtworkFromFile ([in] BSTR filePath)
	Replace existing artwork data with new artwork from an image file.
HRESULT	SaveArtworkToFile ([in] BSTR filePath)
	Save artwork data to an image file.

Properties

HRESULT	Format ([out, retval] ITArtworkFormat *format) Returns the format of the artwork.
HRESULT	IsDownloadedArtwork ([out, retval] VARIANT_BOOL *isDownloadedArtwork) Returns true if the artwork was downloaded by iTunes.
HRESULT	Description ([out, retval] BSTR *description) Returns the description for the artwork. Artwork descriptions are only supported in files that use ID3 tags (i.e. MP3 files).
HRESULT	Description ([in] BSTR description) Sets the description for the artwork. Artwork descriptions are only supported in files that use ID3 tags (i.e. MP3 files).

Member Function Documentation

HRESULT IITArtwork::Delete ()

Delete this piece of artwork from the track.

Return values:

S_*OK* The operation was successful.

ITUNES E OBJECTDELETED This artwork or the track

containing this artwork has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITArtwork::Description ([in] BSTR description)

Sets the description for the artwork. Artwork descriptions are only supported in files that use ID3 tags (i.e. MP3 files).

Note:

This property is available in iTunes 7.7 and later (iTunes type library 1.11 and later).

Parameters:

description The new description for the artwork.

Return values:

S_OK The operation was successful.

E_POINTER description is NULL.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This artwork or the track

containing this artwork has been

deleted.

HRESULT IITArtwork::Description ([out, retval] BSTR * description

Returns the description for the artwork. Artwork descriptions are only supported in files that use ID3 tags (i.e. MP3 files).

Note:

This property is available in iTunes 7.7 and later (iTunes type library 1.11 and later).

Parameters:

description Returns the description for the artwork.

Return values:

S_*OK* The operation was successful.

E_POINTER description is NULL.

ITUNES_E_OBJECTDELETED This artwork or the track

containing this artwork has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITArtwork::Format ([out, retval] ITArtworkFormat * form

Returns the format of the artwork.

Parameters:

format Returns the format of the artwork.

Return values:

S_OK The operation was successful.

E POINTER format is NULL.

ITUNES_E_OBJECTDELETED This artwork or the track

containing this artwork has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITArtwork::IsDownloadedArtwork ([out, retval] VARIAN

Returns true if the artwork was downloaded by iTunes.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

isDownloadedArtwork Returns true if the artwork was downloaded by iTunes.

Return values:

S_OK The operation was successful.

E_POINTER isDownloadedArtwork is NULL.

ITUNES E OBJECTDELETED This artwork or the track

containing this artwork has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITArtwork::SaveArtworkToFile ([in] BSTR filePath)

Save artwork data to an image file.

The format of the saved data is specified by the artwork's format (JPEG, PNG, or BMP). The directory that contains the file must already exist, it will not be created. If the file already exists, its contents will be replaced.

Parameters:

filePath Full path to the artwork image file.

Return values:

S_*OK* The operation was successful.

E_POINTER filePath is NULL.

E_INVALIDARG filePath is not a valid path. *ITUNES E OBJECTDELETED* This artwork or the track

containing this artwork has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITArtwork::SetArtworkFromFile ([in] BSTR filePath)

Replace existing artwork data with new artwork from an image file.

Parameters:

filePath Full path to the artwork image file. The image format must be supported by QuickTime (e.g. JPEG, GIF, PNG, BMP, etc.)

Return values:

S_OK The operation was successful.

E_POINTER filePath is NULL.

E_INVALIDARG filePath is not a valid path.

 $ITUNES_E_OBJECTDELETED$ This artwork or the track

containing this artwork has been

deleted.

E_FAIL An unexpected error occurred.

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITArtworkCollection Interface Reference

List of all members.

Detailed Description

Represents a collection of iTunes artwork objects.

An artwork collection is always associated with an individual track.

The **IITTrack::Artwork** property returns the collection of artwork for the track.

Public Member Functions

Properties

HRESULT	Count ([out, retval] long *count)
	Returns the number of pieces of artwork in the collection.
HRESULT	Item ([in] long index,[out, retval] IITArtwork **iArtwork)
	Returns an IITArtwork object corresponding to the given index (1-based).
HRESULT	_NewEnum ([out, retval] IUnknown **iEnumerator)
	Returns an IEnumVARIANT object which can enumerate the collection.

Member Function Documentation

HRESULT IITArtworkCollection::_NewEnum ([out, retval] IUnknow

Returns an IEnumVARIANT object which can enumerate the collection.

Note:

This property is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Parameters:

iEnumerator Returns an IEnumVARIANT object which can enumerate the collection.

Return values:

S_OK The operation was successful.

E POINTER iEnumerator is NULL.

E_FAIL An unexpected error occurred.

HRESULT IITArtworkCollection::Count ([out, retval] long * count)

Returns the number of pieces of artwork in the collection.

Parameters:

count Returns the number of pieces of artwork in the collection.

Return values:

S_OK The operation was successful.

E_POINTER count is NULL.

ITUNES_E_OBJECTDELETED The track containing this artwork

has been deleted.

E_FAIL An unexpected error occurred.

```
HRESULT IITArtworkCollection::Item ( [in] long inc [out, retval] IITArtwork ** iAi ]
```

Returns an **IITArtwork** object corresponding to the given index (1-based).

Parameters:

index 1-based index of **IITArtwork** to retrieve.

iArtwork Returns an IITArtwork object corresponding to the

specified index.

Return values:

S_OK The operation was successful.

S_FALSE index is too high.

E_INVALIDARG index is less than one.

E_POINTER iArtwork is NULL.

ITUNES_E_OBJECTDELETED The track containing this artwork

has been deleted.

E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITAudioCDPlaylist Interface Reference

Inheritance diagram for IITAudioCDPlaylist:



List of all members.

Detailed Description

Represents an audio CD playlist.

An audio CD playlist is always associated with an **IITSource** of kind **ITSourceKindAudioCD**.

You can retrieve all the playlists defined for a source using **IITSource::Playlists()**.

Public Member Functions

Properties

HRESULT	Artist ([out, retval] BSTR *artist) Returns the audio CD's artist.
HRESULT	Compilation ([out, retval] VARIANT_BOOL *isCompiliation) Returns true if this audio CD is a compilation album.
HRESULT	Composer ([out, retval] BSTR *composer) Returns the audio CD's composer.
HRESULT	DiscCount ([out, retval] long *discCount) Returns the total number of discs in this CD's album.
HRESULT	DiscNumber ([out, retval] long *discNumber) Returns the index of the CD disc in the source album.
HRESULT	Genre ([out, retval] BSTR *genre) Returns the audio CD's genre.
HRESULT	Year ([out, retval] long *year) Returns the year the album was recorded/released.

Methods

HRESULT	Reveal ()
	Reveals the CD playlist in the main browser window.

Member Function Documentation

HRESULT IITAudioCDPlaylist::Artist ([out, retval] BSTR * artist)

Returns the audio CD's artist.

Parameters:

artist Returns the audio CD's artist.

Return values:

S_OK The operation was successful.

E_POINTER artist is NULL.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITAudioCDPlaylist::Compilation ([out, retval] VARIANT_

Returns true if this audio CD is a compilation album.

Parameters:

isCompiliation Returns true if this audio CD is a compilation album.

Return values:

S_OK The operation was successful.

E_POINTER isCompiliation is NULL.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITAudioCDPlaylist::Composer ([out, retval] BSTR * con

Returns the audio CD's composer.

Parameters:

composer Returns the audio CD's composer.

Return values:

S_OK The operation was successful.

E_POINTER composer is NULL.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITAudioCDPlaylist::DiscCount ([out, retval] long * disc

Returns the total number of discs in this CD's album.

Parameters:

discCount Returns the total number of discs in this CD's album.

Return values:

S_OK The operation was successful.

E_POINTER discCount is NULL.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITAudioCDPlaylist::DiscNumber ([out, retval] long * dis

Returns the index of the CD disc in the source album.

Parameters:

discNumber Returns the index of the CD disc in the source album.

Return values:

S_OK The operation was successful.

E_POINTER discNumber is NULL.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITAudioCDPlaylist::Genre ([out, retval] BSTR * genre)

Returns the audio CD's genre.

Parameters:

genre Returns the audio CD's genre.

Return values:

S_OK The operation was successful.

E_POINTER genre is NULL.

 $ITUNES_E_OBJECTDELETED$ This playlist has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITAudioCDPlaylist::Reveal ()

Reveals the CD playlist in the main browser window.

Note:

This method is available in iTunes 7.4 and later (iTunes type library 1.10 and later).

Return values:

S_OK The operation was successful.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITAudioCDPlaylist::Year ([out, retval] long * year)

Returns the year the album was recorded/released.

Parameters:

year Returns the year the album was recorded/released.

Return values:

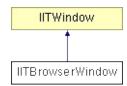
 S_OK The operation was successful. $E_POINTER$ year is NULL. $ITUNES_E_OBJECTDELETED$ This playlist has been deleted. E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITBrowserWindow Interface Reference

Inheritance diagram for IITBrowserWindow:



List of all members.

Detailed Description

Represents the main browser window.

You can retrieve the main browser window using the **liTunes::BrowserWindow** property.

Public Member Functions

Properties

HRESULT	MiniPlayer ([out, retval] VARIANT_BOOL *isMiniPlayer) Returns true if browser window is in Mini Player mode.
HRESULT	MiniPlayer ([in] VARIANT_BOOL shouldBeMiniPlayer) Switch into or out of Mini Player mode.
HRESULT	SelectedTracks ([out, retval] IITTrackCollection **iTrackCollection) Returns a collection containing the currently selected track or tracks.
HRESULT	SelectedPlaylist ([out, retval] IITPlaylist **iPlaylist) Returns the currently selected playlist in the Source list.
HRESULT	SelectedPlaylist ([in] VARIANT *iPlaylist) Sets the currently selected playlist in the Source list.

Member Function Documentation

HRESULT IITBrowserWindow::MiniPlayer ([in] VARIANT_BOOL st

Switch into or out of Mini Player mode.

Parameters:

shouldBeMiniPlayer If true, switch to Mini Player mode.

Return values:

 S_OK The operation was successful. ITUNES_E_OBJECTLOCKED The window is maximized. ITUNES_E_OBJECTDELETED The window has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITBrowserWindow::MiniPlayer ([out, retval] VARIANT_E

Returns true if browser window is in Mini Player mode.

Parameters:

isMiniPlayer Returns true if browser window is in Mini Player mode.

Return values:

 S_OK The operation was successful. $E_POINTER$ is MiniPlayer is NULL. $ITUNES_E_OBJECTDELETED$ The window has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITBrowserWindow::SelectedPlaylist ([in] VARIANT * iPl

Sets the currently selected playlist in the Source list.

Parameters:

iPlaylist The playlist to select. This must be a VARIANT of type VT_DISPATCH that points to an **IITPlaylist**.

Return values:

S_OK The operation was successful.

E_POINTER iPlaylist is NULL.

E_INVALIDARG iPlaylist is not the correct

VARIANT type.

ITUNES_E_OBJECTDELETED The playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITBrowserWindow::SelectedPlaylist ([out, retval] IITPla

Returns the currently selected playlist in the Source list.

Parameters:

iPlaylist Returns the currently selected playlist in the Source list.

Return values:

S_OK The operation was successful.

E_POINTER iPlaylist is NULL.

ITUNES_E_OBJECTDELETED The window has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITBrowserWindow::SelectedTracks ([out, retval] IITTrac

Returns a collection containing the currently selected track or tracks.

Parameters:

iTrackCollection Collection of **IITTrack** objects. Will be set to NULL if there is no current selection.

Return values:

 S_OK The operation was successful. S_FALSE There is no current selection. $E_POINTER$ iTrackCollection is NULL. $ITUNES_E_OBJECTDELETED$ The window has been deleted.

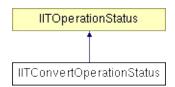
E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITConvertOperationStatus Interface Reference

Inheritance diagram for IITConvertOperationStatus:



List of all members.

Detailed Description

Represents the status of an asynchronous convert operation.

Note:

This interface is available in iTunes 4.6 and later (iTunes type library 1.1 and later).

When converting or importing a track using IiTunes::ConvertFile2(), IiTunes::ConvertFiles2(), IiTunes::ConvertTrack2() or IiTunes::ConvertTracks2(), the conversion will never complete immediately. These methods return an IITConvertOperationStatus object, which can be polled to determine when the operation is done. You can also register an _IITConvertOperationStatusEvents interface to be notified of conversion progress.

Public Member Functions

Methods

HRESULT	long *maxProgressValue)
	Returns status about the track currently being converted.
HRESULT	StopConversion ()
	Stops the current conversion operation.

Properties

HRESULT	TrackName ([out, retval] BSTR *trackName) Returns the name of the track currently being converted.
HRESULT	ProgressValue ([out, retval] long *progressValue) Returns the conversion progress value for the track currently being converted.
HRESULT	MaxProgressValue ([out, retval] long *maxProgressValue) Returns the maximum conversion progress value for the track currently being converted.

Member Function Documentation

Returns status about the track currently being converted.

This method is not usable from scripting clients, since it returns multiple output parameters. Scripting clients must use the

IITConvertOperationStatus::TrackName(),

IITConvertOperationStatus::ProgressValue(), and

IITConvertOperationStatus::MaxProgressValue() properties instead.

Parameters:

trackName Returns the name of the track currently being

converted. May return NULL if no track is

currently being converted.

progress Value Returns the conversion progress value for the

track currently being converted. This number

will always be in the range 0 through

maxProgressValue.

maxProgressValue Returns the maximum conversion progress

value for the track currently being converted.

The conversion of the current track is complete when progressValue reaches

maxProgressValue. Note that

maxProgressValue will be different for each

track being converted.

Return values:

S_OK The operation was successful.

E_POINTER trackName, progressValue, or maxProgressValue is

NULL.

E_FAIL An unexpected error occurred.

HRESULT IITConvertOperationStatus::MaxProgressValue ([out, re-

Returns the maximum conversion progress value for the track currently being converted.

Parameters:

maxProgressValue Returns the maximum conversion progress

value for the track currently being converted. The conversion of the current track is complete when

IITConvertOperationStatus::ProgressValue(

reaches maxProgressValue. Note that maxProgressValue will be different for each track being converted.

Return values:

S_OK The operation was successful.

E_POINTER maxProgressValue is NULL.

E_FAIL An unexpected error occurred.

HRESULT IITConvertOperationStatus::ProgressValue ([out, retval]

Returns the conversion progress value for the track currently being converted.

Parameters:

progress Value Returns the conversion progress value for the track currently being converted. This number will always be in the range 0 through

IITConvertOperationStatus::MaxProgressValue(

Return values:

S_OK The operation was successful.

E_POINTER progressValue is NULL.

E_FAIL An unexpected error occurred.

HRESULT IITConvertOperationStatus::StopConversion ()

Stops the current conversion operation.

Return values:

S_OK The operation was successful.

E_FAIL The conversion could not be stopped, or an unexpected error occurred.

HRESULT IITConvertOperationStatus::TrackName ([out, retval] BS

Returns the name of the track currently being converted.

Parameters:

trackName Returns the name of the track currently being converted. May return NULL if no track is currently being converted.

Return values:

S_OK The operation was successful.

E_POINTER trackName is NULL.

©2004-2007 Apple Computer, Inc.

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITEncoder Interface Reference

List of all members.

Detailed Description

Represents an iTunes encoder.

You can retrieve or set the currently selected encoder using the **liTunes::CurrentEncoder** property.

Public Member Functions

Properties

HRESULT	Name ([out, retval] BSTR *name)
	Returns the name of the encoder (e.g. "AAC Encoder").
HRESULT	Format ([out, retval] BSTR *format)
	Returns the data format created by the encoder (e.g. "AAC").

Member Function Documentation

HRESULT IITEncoder::Format ([out, retval] BSTR * format)

Returns the data format created by the encoder (e.g. "AAC").

Parameters:

format Returns the data format created by the encoder (e.g. "AAC").

Return values:

S *OK* The operation was successful.

E POINTER format is NULL.

ITUNES_E_OBJECTDELETED This encoder is not available.

E_FAIL An unexpected error occurred.

HRESULT IITEncoder::Name ([out, retval] BSTR * name)

Returns the name of the encoder (e.g. "AAC Encoder").

Parameters:

name Returns the name of the encoder (e.g. "AAC Encoder").

Return values:

S_OK The operation was successful.

E_POINTER name is NULL.

 $\label{lem:itunes_encoder} \textit{ITUNES}_\textit{E_OBJECTDELETED} \ \ \textit{This encoder is not available}.$

©2004-2007 Apple Computer, Inc.

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITEncoderCollection Interface Reference

List of all members.

Detailed Description

Represents a collection of encoder objects.

The **liTunes::Encoders** property returns the collection of encoders.

Public Member Functions

Properties

HRESULT	Count ([out, retval] long *count)
	Returns the number of encoders in the collection.
HRESULT	Item ([in] long index,[out, retval] IITEncoder **iEncoder)
	Returns an IITEncoder object corresponding to the given index (1-based).
HRESULT	ItemByName ([in] BSTR name,[out, retval] IITEncoder **iEncoder)
	Returns an IITEncoder object with the specified name.
HRESULT	_NewEnum ([out, retval] IUnknown **iEnumerator)
	Returns an IEnumVARIANT object which can enumerate the collection.

Member Function Documentation

HRESULT IITEncoderCollection::_NewEnum ([out, retval] IUnknow

Returns an IEnumVARIANT object which can enumerate the collection.

Note:

This property is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Parameters:

iEnumerator Returns an IEnumVARIANT object which can enumerate the collection.

Return values:

S_OK The operation was successful.

E POINTER iEnumerator is NULL.

E_FAIL An unexpected error occurred.

HRESULT IITEncoderCollection::Count ([out, retval] long * count

Returns the number of encoders in the collection.

Parameters:

count Returns the number of encoders in the collection.

Return values:

S_OK The operation was successful.

E_POINTER count is NULL.

```
HRESULT IITEncoderCollection::Item ( [in] long in [out, retval] IITEncoder ** il ]
```

Returns an **IITEncoder** object corresponding to the given index (1-based).

Parameters:

index 1-based index of **IITEncoder** to retrieve.

iEncoder Returns an IITEncoder object corresponding to the

specified index.

Return values:

S_OK The operation was successful.

S_FALSE index is too big.

E INVALIDARG index is less than one.

E POINTER iEncoder is NULL.

E_FAIL An unexpected error occurred.

```
HRESULT IITEncoderCollection::ItemByName ( [in] BSTR [out, retval] IITEnco
```

Returns an **IITEncoder** object with the specified name.

Parameters:

name 1-based index of **IITEncoder** to retrieve.

iEncoder Returns an **IITEncoder** object with the specified name.

Return values:

S_OK The operation was successful.

S_FALSE An encoder with the specified name could not be

found.

E_POINTER name or iEncoder is NULL.

E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITEQPreset Interface Reference

List of all members.

Detailed Description

Represents an equalizer preset.

You can retrieve or set the currently selected EQ preset using the **liTunes::CurrentEQPreset** property.

Public Member Functions

Properties

HRESULT	Name ([out, retval] BSTR *name)
	Returns the name of the the EQ preset (e.g. "Acoustic").
HRESULT	Modifiable ([out, retval] VARIANT_BOOL *isModifiable)
	Returns true if the EQ preset can be modified.

Preamp and Band Properties

HRESULT	Preamp ([out, retval] double *level)
	Returns the equalizer preamp level (-12.0 db to +12.0 db).
HRESULT	Preamp ([in] double level)
	Set the equalizer preamp level (-12.0 db to +12.0 db).
HRESULT	Band1 ([out, retval] double *level)
	Returns the equalizer 32Hz band level (-12.0 db to +12.0 db).
HRESULT	Band1 ([in] double level)
	Set the equalizer 32Hz band level (-12.0 db to +12.0 db).
HRESULT	Band2 ([out, retval] double *level)
	Returns the equalizer 64Hz band level (-12.0 db to +12.0 db).
HRESULT	Band2 ([in] double level)
	Set the equalizer 64Hz band level (-12.0 db to +12.0 db).
HRESULT	Band3 ([out, retval] double *level)
	Returns the equalizer 125Hz band level (-12.0 db to +12.0 db).
HRESULT	Band3 ([in] double level)
	Set the equalizer 125Hz band level (-12.0 db to +12.0 db).
HRESULT	Band4 ([out, retval] double *level)
	Returns the equalizer 250Hz band level (-12.0 db to +12.0 db).
HRESULT	Band4 ([in] double level)
	Set the equalizer 250Hz band level (-12.0 db to +12.0 db).
HRESULT	Band5 ([out, retval] double *level)
	Returns the equalizer 500Hz band level (-12.0 db to +12.0 db).
HRESULT	Band5 ([in] double level)
	Set the equalizer 500Hz band level (-12.0 db to +12.0 db).
HRESULT	Band6 ([out, retval] double *level)
	Returns the equalizer 1KHz band level (-12.0 db to +12.0 db).

HRESULT	Band6 ([in] double level)
	Set the equalizer 1KHz band level (-12.0 db to +12.0 db).
HRESULT	Band7 ([out, retval] double *level)
	Returns the equalizer 2KHz band level (-12.0 db to +12.0 db).
HRESULT	Band7 ([in] double level)
	Set the equalizer 2KHz band level (-12.0 db to +12.0 db).
HRESULT	Band8 ([out, retval] double *level)
	Returns the equalizer 4KHz band level (-12.0 db to +12.0 db).
HRESULT	Band8 ([in] double level)
	Set the equalizer 4KHz band level (-12.0 db to +12.0 db).
HRESULT	Band9 ([out, retval] double *level)
	Returns the equalizer 8KHz band level (-12.0 db to +12.0 db).
HRESULT	Band9 ([in] double level)
	Set the equalizer 8KHz band level (-12.0 db to +12.0 db).
HRESULT	Band10 ([out, retval] double *level)
	Returns the equalizer 16KHz band level (-12.0 db to +12.0 db).
HRESULT	Band10 ([in] double level)
	Set the equalizer 16KHz band level (-12.0 db to +12.0 db).

Methods

HRESULT	Delete ([in] VARIANT_BOOL updateAllTracks)
	Delete this EQ preset.
HRESULT	Rename ([in] BSTR newName,[in] VARIANT_BOOL updateAllTracks)
	Rename this EQ preset.

Member Function Documentation

HRESULT IITEQPreset::Band1 ([in] double level)

Set the equalizer 32Hz band level (-12.0 db to +12.0 db).

Parameters:

level The equalizer 32Hz band level (-12.0 db to +12.0 db).

If level is outside the range -12.0 to +12.0, it will be pinned.

Return values:

S_*OK* The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Band1 ([out, retval] double * level)

Returns the equalizer 32Hz band level (-12.0 db to +12.0 db).

Parameters:

level Returns the equalizer 32Hz band level (-12.0 db to +12.0 db).

Return values:

S_OK The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

HRESULT IITEQPreset::Band10 ([in] double level)

Set the equalizer 16KHz band level (-12.0 db to +12.0 db).

Parameters:

level The equalizer 8KHz band level (-12.0 db to +12.0 db).

If level is outside the range -12.0 to +12.0, it will be pinned.

Return values:

S_OK The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Band10 ([out, retval] double * level)

Returns the equalizer 16KHz band level (-12.0 db to +12.0 db).

Parameters:

level Returns the equalizer 8KHz band level (-12.0 db to +12.0 db).

Return values:

S_*OK* The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

HRESULT IITEQPreset::Band2 ([in] double level)

Set the equalizer 64Hz band level (-12.0 db to +12.0 db).

Parameters:

level The equalizer 64Hz band level (-12.0 db to +12.0 db).

If level is outside the range -12.0 to +12.0, it will be pinned.

Return values:

S_OK The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Band2 ([out, retval] double * level)

Returns the equalizer 64Hz band level (-12.0 db to +12.0 db).

Parameters:

level Returns the equalizer 64Hz band level (-12.0 db to +12.0 db).

Return values:

S_OK The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

HRESULT IITEQPreset::Band3 ([in] double *level*)

Set the equalizer 125Hz band level (-12.0 db to +12.0 db).

Parameters:

level The equalizer 125Hz band level (-12.0 db to +12.0 db).

If level is outside the range -12.0 to +12.0, it will be pinned.

Return values:

S_OK The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Band3 ([out, retval] double * level)

Returns the equalizer 125Hz band level (-12.0 db to +12.0 db).

Parameters:

level Returns the equalizer 125Hz band level (-12.0 db to +12.0 db).

Return values:

S_OK The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Band4 ([in] double *level*)

Set the equalizer 250Hz band level (-12.0 db to +12.0 db).

Parameters:

level The equalizer 250Hz band level (-12.0 db to +12.0 db).

If level is outside the range -12.0 to +12.0, it will be pinned.

Return values:

S_OK The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Band4 ([out, retval] double * level)

Returns the equalizer 250Hz band level (-12.0 db to +12.0 db).

Parameters:

level Returns the equalizer 250Hz band level (-12.0 db to +12.0 db).

Return values:

S_*OK* The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Band5 ([in] double level)

Set the equalizer 500Hz band level (-12.0 db to +12.0 db).

Parameters:

level The equalizer 500Hz band level (-12.0 db to +12.0 db).

If level is outside the range -12.0 to +12.0, it will be pinned.

Return values:

S_OK The operation was successful.

E POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Band5 ([out, retval] double * level)

Returns the equalizer 500Hz band level (-12.0 db to +12.0 db).

Parameters:

level Returns the equalizer 500Hz band level (-12.0 db to +12.0 db).

Return values:

S_OK The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Band6 ([in] double level)

Set the equalizer 1KHz band level (-12.0 db to +12.0 db).

Parameters:

level The equalizer 1KHz band level (-12.0 db to +12.0 db).

If level is outside the range -12.0 to +12.0, it will be pinned.

Return values:

S_*OK* The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Band6 ([out, retval] double * level)

Returns the equalizer 1KHz band level (-12.0 db to +12.0 db).

Parameters:

level Returns the equalizer 1KHz band level (-12.0 db to +12.0 db).

Return values:

S_OK The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Band7 ([in] double *level*)

Set the equalizer 2KHz band level (-12.0 db to +12.0 db).

Parameters:

level The equalizer 2KHz band level (-12.0 db to +12.0 db).

If level is outside the range -12.0 to +12.0, it will be pinned.

Return values:

S_OK The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Band7 ([out, retval] double * level)

Returns the equalizer 2KHz band level (-12.0 db to +12.0 db).

Parameters:

level Returns the equalizer 2KHz band level (-12.0 db to +12.0 db).

Return values:

S_OK The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Band8 ([in] double level)

Set the equalizer 4KHz band level (-12.0 db to +12.0 db).

Parameters:

level The equalizer 4KHz band level (-12.0 db to +12.0 db).

If level is outside the range -12.0 to +12.0, it will be pinned.

Return values:

S_OK The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Band8 ([out, retval] double * level)

Returns the equalizer 4KHz band level (-12.0 db to +12.0 db).

Parameters:

level Returns the equalizer 4KHz band level (-12.0 db to +12.0 db).

Return values:

S_OK The operation was successful.

E POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Band9 ([in] double level)

Set the equalizer 8KHz band level (-12.0 db to +12.0 db).

Parameters:

level The equalizer 8KHz band level (-12.0 db to +12.0 db).

If level is outside the range -12.0 to +12.0, it will be pinned.

Return values:

S_*OK* The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E FAIL An unexpected error occurred.

HRESULT IITEQPreset::Band9 ([out, retval] double * level)

Returns the equalizer 8KHz band level (-12.0 db to +12.0 db).

Parameters:

level Returns the equalizer 8KHz band level (-12.0 db to +12.0 db).

Return values:

S_OK The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Delete ([in] VARIANT_BOOL updateAllTrace

Delete this EQ preset.

Any EQ preset can be deleted, including built-in presets, except for the Manual preset.

Note:

This method is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Parameters:

updateAllTracks If true, any tracks that use this EQ preset will be set to have no assigned EQ preset.

Return values:

S_*OK* The operation was successful.

ITUNES_E_OBJECTDELETED This EQ preset has been deleted

already.

ITUNES_E_OBJECTLOCKED This is the Manual EQ preset,

which cannot be deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Modifiable ([out, retval] VARIANT_BOOL *

Returns true if the EQ preset can be modified.

Prior to iTunes 4.7, only the Manual preset was modifiable. As of iTunes 4.7 and later (iTunes type library 1.2 and later), all presets can be modified. In other words, this property will always be true.

Parameters:

isModifiable Returns true if the EQ preset can be modified.

Return values:

S_OK The operation was successful.

E_POINTER is Modifiable is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Name ([out, retval] BSTR * name)

Returns the name of the EQ preset (e.g. "Acoustic").

Parameters:

name Returns the name of the the EQ preset (e.g. "Acoustic").

Return values:

S_*OK* The operation was successful.

E POINTER name is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Preamp ([in] double level)

Set the equalizer preamp level (-12.0 db to +12.0 db).

Parameters:

level The equalizer preamp level (-12.0 db to +12.0 db).

If level is outside the range -12.0 to +12.0, it will be pinned.

Return values:

S_OK The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Preamp ([out, retval] double * level)

Returns the equalizer preamp level (-12.0 db to +12.0 db).

Parameters:

level Returns the equalizer preamp level (-12.0 db to +12.0 db).

Return values:

S_OK The operation was successful.

E_POINTER level is NULL.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

HRESULT IITEQPreset::Rename ([in] BSTR newName, [in] VARIANT_BOOL updateAllTi
)

Rename this EQ preset.

The name of any EQ preset can be changed, including built-in presets, except for the Manual preset.

EQ preset names cannot start with leading spaces. If you specify a name that starts with leading spaces they will be stripped out.

Note:

This method is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Parameters:

newName The new name of the EQ preset.

updateAllTracks If true, any tracks that use this EQ preset will be

updated with the new preset name.

Return values:

S_OK The operation was successful.

E_POINTER newName is NULL.

E_INVALIDARG newName is empty.

ITUNES_E_OBJECTEXISTS An EQ preset with this name

already exists.

ITUNES_E_OBJECTLOCKED This is the Manual EQ preset,

which cannot be renamed.

ITUNES_E_OBJECTDELETED The EQ preset has been

deleted.

E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITEQPresetCollection Interface Reference

List of all members.

Detailed Description

Represents a collection of EQ preset objects.

The **liTunes::EQPresets** property returns the collection of EQ presets.

Public Member Functions

Properties

HRESULT	Count ([out, retval] long *count) Returns the number of EQ presets in the collection.
HRESULT	Item ([in] long index,[out, retval] IITEQPreset **iEQPreset) Returns an IITEQPreset object corresponding to the given index (1-based).
HRESULT	ItemByName ([in] BSTR name,[out, retval] IITEQPreset **iEQPreset) Returns an IITEQPreset object with the specified name.
HRESULT	_NewEnum ([out, retval] IUnknown **iEnumerator) Returns an IEnumVARIANT object which can enumerate the collection.

Member Function Documentation

HRESULT IITEQPresetCollection::_NewEnum ([out, retval] IUnkno

Returns an IEnumVARIANT object which can enumerate the collection.

Note:

This property is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Parameters:

iEnumerator Returns an IEnumVARIANT object which can enumerate the collection.

Return values:

S_OK The operation was successful.

E POINTER iEnumerator is NULL.

E_FAIL An unexpected error occurred.

HRESULT IITEQPresetCollection::Count ([out, retval] long * count

Returns the number of EQ presets in the collection.

Parameters:

count Returns the number of EQ presets in the collection.

Return values:

S_OK The operation was successful.

E_POINTER count is NULL.

E_FAIL An unexpected error occurred.

Returns an **IITEQPreset** object corresponding to the given index (1-based).

Parameters:

index 1-based index of **IITEQPreset** to retrieve.

iEQPreset Returns an **IITEQPreset** object corresponding to the

specified index.

Return values:

S *OK* The operation was successful.

S_FALSE index is too big.

E_INVALIDARG index is less than one. *E_POINTER* iEQPreset is NULL.

E_FAIL An unexpected error occurred.

```
HRESULT IITEQPresetCollection::ItemByName ( [in] BSTR [out, retval] IITEQI )
```

Returns an **IITEQPreset** object with the specified name.

Parameters:

name 1-based index of **IITEQPreset** to retrieve.

iEQPreset Returns an IITEQPreset object with the specified

name.

Return values:

S_OK The operation was successful.

S_FALSE An EQ preset with the specified name could not

be found.

 $E_POINTER$ name or iEQPreset is NULL.

 $E_INVALIDARG$ name is empty.

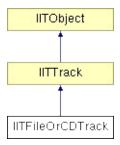
E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITFileOrCDTrack Interface Reference

Inheritance diagram for IITFileOrCDTrack:



List of all members.

Detailed Description

Represents a file or CD track.

Public Member Functions

Properties

•	
HRESULT	Location ([out, retval] BSTR *location) Returns the full path to the file represented by this track.
HRESULT	Podcast ([out, retval] VARIANT_BOOL *isPodcast) Returns true if this track is a podcast track.
HRESULT	RememberBookmark ([out, retval] VARIANT_BOOL *rememberBookmark) Returns true if playback position is remembered for this track.
HRESULT	RememberBookmark ([in] VARIANT_BOOL shouldRememberBookmark) Set whether the playback position is remembered for this track.
HRESULT	ExcludeFromShuffle ([out, retval] VARIANT_BOOL *excludeFromShuffle) Returns true if this track is skipped when shuffling.
HRESULT	ExcludeFromShuffle ([in] VARIANT_BOOL shouldExcludeFromShuffle) Set whether this track is skipped when shuffling.
HRESULT	Lyrics ([out, retval] BSTR *lyrics) Returns the lyrics for the track.
HRESULT	Lyrics ([in] BSTR lyrics) Sets the lyrics for the track.
HRESULT	Category ([out, retval] BSTR *category) Returns the category for the track.
HRESULT	Category ([in] BSTR category) Sets the category for the track.
HRESULT	Description ([out, retval] BSTR *description) Returns the description for the track.
HRESULT	Description ([in] BSTR description) Sets the description for the track.
HRESULT	LongDescription ([out, retval] BSTR *longDescription) Returns the long description for the track.
HRESULT	LongDescription ([in] BSTR longDescription) Sets the long description for the track.
HRESULT	BookmarkTime ([out, retval] long *bookmarkTime) Returns the bookmark time of the track (in seconds). This will always be zero if the track is not bookmarkable.
HRESULT	BookmarkTime ([in] long bookmarkTime) Set the bookmark time of the track (in seconds).

HRESULT	VideoKind ([out, retval] ITVideoKind *videoKind) Returns the video track kind. This will be ITVideoKindNone if the track is not a
	video track.
HRESULT	VideoKind ([in] ITVideoKind videoKind)
	Set the video track kind. Note that the video kind cannot be changed for purchased content.
HRESULT	SkippedCount ([out, retval] long *skippedCount) Returns the number of times the track has been skipped.
HRESULT	SkippedCount ([in] long skippedCount)
	Set the number of times the track has been skipped. This property cannot be set if the track is not playable (e.g. a PDF file).
HRESULT	SkippedDate ([out, retval] DATE *skippedDate)
	Returns the date and time the track was last skipped.
HRESULT	SkippedDate ([in] DATE skippedDate)
	Set the date and time the track was last skipped. This property cannot be set if the track is not playable (e.g. a PDF file).
HRESULT	PartOfGaplessAlbum ([out, retval] VARIANT_BOOL *partOfGaplessAlbum)
	Returns true if this track is part of a gapless album.
HRESULT	PartOfGaplessAlbum ([in] VARIANT_BOOL shouldBePartOfGaplessAlbum) Set whether this track is part of a gapless album.
HRESULT	AlbumArtist ([out, retval] BSTR *albumArtist) Returns the name of the album artist of the track.
HRESULT	AlbumArtist ([in] BSTR albumArtist)
	Set the name of the album artist of the track.
HRESULT	Show ([out, retval] BSTR *showName)
	Returns the show name of the track.
HRESULT	Show ([in] BSTR showName) Set the show name of the track.
HRESULT	SeasonNumber ([out, retval] long *seasonNumber)
	Returns the season number of the track.
HRESULT	SeasonNumber ([in] long seasonNumber) Set the season number of the track.
HRESULT	EpisodeID ([out, retval] BSTR *episodeID) Returns the episode ID of the track.
HRESULT	EpisodelD ([in] BSTR episodelD)
I II (LOOL)	Set the episode ID of the track.
HRESULT	EpisodeNumber ([out, retval] long *episodeNumber)
	Returns the episode number of the track.
HRESULT	EpisodeNumber ([in] long episodeNumber)
	Set the episode number of the track.

HRESULT	Size64High ([out, retval] long *sizeHigh) Returns the high 32-bits of the size of the track (in bytes).
HRESULT	Size64Low ([out, retval] long *sizeLow) Returns the low 32-bits of the size of the track (in bytes).
HRESULT	Unplayed ([out, retval] VARIANT_BOOL *isUnplayed) Returns false if this track has been played.
HRESULT	Unplayed ([in] VARIANT_BOOL shouldBeUnplayed) Set whether this track has been played.
HRESULT	SortAlbum ([out, retval] BSTR *album) Returns the name of the album used for sorting. If a sort album is specified, iTunes will use this string instead of the real album name when sorting.
HRESULT	SortAlbum ([in] BSTR album)
	Set the name of the album used for sorting. If a sort album is specified, iTunes will use this string instead of the real album name when sorting.
HRESULT	SortAlbumArtist ([out, retval] BSTR *albumArtist)
	Returns the name of the album artist used for sorting. If a sort album artist is specified, iTunes will use this string instead of the real album artist when sorting.
HRESULT	SortAlbumArtist ([in] BSTR albumArtist)
	Set the name of the album artist used for sorting. If a sort album artist is specified, iTunes will use this string instead of the real album artist when sorting.
HRESULT	SortArtist ([out, retval] BSTR *artist) Returns the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting.
HRESULT	Returns the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting.
	Returns the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting.
	Returns the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting. SortArtist ([in] BSTR artist) Set the name of the artist used for sorting. If a sort artist is specified, iTunes will
HRESULT	Returns the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting. SortArtist ([in] BSTR artist) Set the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting.
HRESULT	Returns the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting. SortArtist ([in] BSTR artist) Set the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting. SortComposer ([out, retval] BSTR *composer) Returns the composer used for sorting. If a sort composer is specified, iTunes will
HRESULT	Returns the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting. SortArtist ([in] BSTR artist) Set the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting. SortComposer ([out, retval] BSTR *composer) Returns the composer used for sorting. If a sort composer is specified, iTunes will use this string instead of the real composer when sorting. SortComposer ([in] BSTR composer) Set the composer used for sorting. If a sort composer is specified, iTunes will use
HRESULT HRESULT	Returns the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting. SortArtist ([in] BSTR artist) Set the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting. SortComposer ([out, retval] BSTR *composer) Returns the composer used for sorting. If a sort composer is specified, iTunes will use this string instead of the real composer when sorting. SortComposer ([in] BSTR composer) Set the composer used for sorting. If a sort composer is specified, iTunes will use this string instead of the real composer when sorting. SortName ([out, retval] BSTR *name) Returns the name of the track used for sorting. If a sort name is specified, iTunes will use this string instead of the real track name when sorting.
HRESULT HRESULT HRESULT	Returns the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting. SortArtist ([in] BSTR artist) Set the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting. SortComposer ([out, retval] BSTR *composer) Returns the composer used for sorting. If a sort composer is specified, iTunes will use this string instead of the real composer when sorting. SortComposer ([in] BSTR composer) Set the composer used for sorting. If a sort composer is specified, iTunes will use this string instead of the real composer when sorting. SortName ([out, retval] BSTR *name) Returns the name of the track used for sorting. If a sort name is specified, iTunes will use this string instead of the real track name when sorting.
HRESULT HRESULT HRESULT	Returns the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting. SortArtist ([in] BSTR artist) Set the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting. SortComposer ([out, retval] BSTR *composer) Returns the composer used for sorting. If a sort composer is specified, iTunes will use this string instead of the real composer when sorting. SortComposer ([in] BSTR composer) Set the composer used for sorting. If a sort composer is specified, iTunes will use this string instead of the real composer when sorting. SortName ([out, retval] BSTR *name) Returns the name of the track used for sorting. If a sort name is specified, iTunes will use this string instead of the real track name when sorting. SortName ([in] BSTR name) Set the name of the track used for sorting. If a sort name is specified, iTunes will use this string instead of the real track name when sorting.
HRESULT HRESULT HRESULT	Returns the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting. SortArtist ([in] BSTR artist) Set the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting. SortComposer ([out, retval] BSTR *composer) Returns the composer used for sorting. If a sort composer is specified, iTunes will use this string instead of the real composer when sorting. SortComposer ([in] BSTR composer) Set the composer used for sorting. If a sort composer is specified, iTunes will use this string instead of the real composer when sorting. SortName ([out, retval] BSTR *name) Returns the name of the track used for sorting. If a sort name is specified, iTunes will use this string instead of the real track name when sorting. SortName ([in] BSTR name) Set the name of the track used for sorting. If a sort name is specified, iTunes will use this string instead of the real track name when sorting.

HRESULT **SortShow** ([in] BSTR showName) Set the show name used for sorting. If a sort show is specified, iTunes will use this string instead of the real show name when sorting. HRESULT **AlbumRating** ([out, retval] long *rating) Returns the user or computed rating of the album that this track belongs to (0 to 100). If the album rating has never been set, or has been set to 0, it will be computed based on the ratings of tracks in the album. HRESULT **AlbumRating** ([in] long rating) Set the album rating of the album that this track belongs to (0 to 100). If the album rating is set to 0, it will be computed based on the ratings of tracks in the album. HRESULT AlbumRatingKind ([out, retval] ITRatingKind *ratingKind) Returns the album rating kind. If the album rating has never been set, or has been set to 0, the kind is ITRatingKindComputed. Otherwise, the kind is ITRatingKindUser. HRESULT RatingKind ([out, retval] ITRatingKind *ratingKind) Returns the track rating kind. If the track rating has never been set, or has been set to 0, the kind is ITRatingKindComputed. Otherwise, the kind is *ITRatingKindUser* HRESULT Playlists ([out, retval] IITPlaylistCollection **iPlaylistCollection) Returns a collection of playlists that contain the song that this track represents. HRESULT **Location** ([in] BSTR location) Set the full path to the file represented by this track. iTunes will not move or copy any existing file from the track's current location to the new location, the full path must point to an existing file that is playable by iTunes. Attempting to set the location of a track on CD will always fail with E_INVALIDARG. HRESULT **ReleaseDate** ([out, retval] DATE *releaseDate) Returns release date of the track. This property is read-only.

Methods

HRESULT	UpdateInfoFromFile ()
	Update this track's information with the information stored in its file.
HRESULT	UpdatePodcastFeed ()
	Update the podcast feed for this track. This is equivalent to the user choosing Update Podcast from the contextual menu for the podcast feed that contains this track.
HRESULT	Reveal ()
	Reveals the track in the main browser window.

Member Function Documentation

HRESULT IITFileOrCDTrack::AlbumArtist ([in] BSTR albumArtist]

Set the name of the album artist of the track.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

albumArtist The new name of the album artist of the track.

Return values:

S_OK

E_POINTER

albumArtist is NULL.

ITUNES_E_OBJECTLOCKED

This track is not modifiable.

ITUNES_E_OBJECTDELETED

This track has been deleted.

E_FAIL

An unexpected error occurred.

HRESULT IITFileOrCDTrack::AlbumArtist ([out, retval] BSTR * albu

Returns the name of the album artist of the track.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

albumArtist Returns the name of the album artist of the track.

Return values:

S_OK The operation was successful.

E_POINTER albumArtist is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::AlbumRating ([in] long rating)

Set the album rating of the album that this track belongs to (0 to 100). If the album rating is set to 0, it will be computed based on the ratings of tracks in the album.

Note:

This property is available in iTunes 7.4 and later (iTunes type library 1.10 and later).

Parameters:

rating The new album rating of the album that this track belongs to (0 to 100). If rating is outside this range, it will be pinned.

Return values:

S_OK The operation was successful.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::AlbumRating ([out, retval] long * rational retrieval | rational re

Returns the user or computed rating of the album that this track belongs to (0 to 100). If the album rating has never been set, or has been set to 0, it will be computed based on the ratings of tracks in the album.

Note:

This property is available in iTunes 7.4 and later (iTunes type library 1.10 and later).

Parameters:

rating Returns the album rating of the album that this track belongs to (0 to 100).

Return values:

S_OK The operation was successful.

E POINTER rating is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::AlbumRatingKind ([out, retval] ITRati

Returns the album rating kind. If the album rating has never been set, or has been set to 0, the kind is **ITRatingKindComputed**. Otherwise, the kind is **ITRatingKindUser**.

Note:

This property is available in iTunes 7.4 and later (iTunes type library 1.10 and later).

Parameters:

ratingKind Returns the album rating kind.

Return values:

S_OK The operation was successful.

E_POINTER ratingKind is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::BookmarkTime ([in] long bookmarkT

Set the bookmark time of the track (in seconds).

Note:

This property is available in iTunes 6.0.2 and later (iTunes type library 1.7 and later).

Parameters:

bookmarkTime The new bookmark time of the track (in seconds).

Return values:

S_OK

ITUNES_E_OBJECTLOCKED
This track is not modifiable or is not bookmarkable.

ITUNES_E_OBJECTDELETED
This track has been deleted.

E_FAIL
An unexpected error occurred.

HRESULT IITFileOrCDTrack::BookmarkTime ([out, retval] long * bookmarkTime (

Returns the bookmark time of the track (in seconds). This will always be zero if the track is not bookmarkable.

Note:

This property is available in iTunes 6.0.2 and later (iTunes type library 1.7 and later).

Parameters:

bookmarkTime Returns the bookmark time of the track (in seconds).

Return values:

S_OK The operation was successful. E POINTER bookmarkTime is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::Category ([in] BSTR category)

Sets the category for the track.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

category The new category for the track.

Return values:

S_OK

E_POINTER

Category is NULL.

ITUNES_E_OBJECTLOCKED

This track is not modifiable.

ITUNES_E_OBJECTDELETED

This track has been deleted.

E_FAIL

An unexpected error occurred.

HRESULT IITFileOrCDTrack::Category ([out, retval] BSTR * category

Returns the category for the track.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

category Returns the category for the track.

Return values:

S_OK The operation was successful.

E_POINTER category is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::Description ([in] BSTR description)

Sets the description for the track.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

description The new description for the track.

Return values:

S_OK The operation was successful.

E_POINTER description is NULL.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::Description ([out, retval] BSTR * des

Returns the description for the track.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

description Returns the description for the track.

Return values:

S_*OK* The operation was successful.

E_POINTER description is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::EpisodeID ([in] BSTR episodeID)

Set the episode ID of the track.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

episodeID The new episode ID of the track.

Return values:

S_OK The operation was successful.

E_POINTER episodeID is NULL.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::EpisodeID ([out, retval] BSTR * episo

Returns the episode ID of the track.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

episodeID Returns the episode ID of the track.

Return values:

S_OK The operation was successful.

E_POINTER episodeID is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::EpisodeNumber ([in]long episodeNumber ([in]long episodeNumber)

Set the episode number of the track.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

episodeNumber The new episode number of the track.

Return values:

 S_OK The operation was successful. $ITUNES_E_OBJECTLOCKED$ This track is not modifiable. $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::EpisodeNumber ([out, retval] long * 6

Returns the episode number of the track.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

episodeNumber Returns the episode number of the track.

Return values:

 S_OK The operation was successful. $E_POINTER$ episodeNumber is NULL. $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::ExcludeFromShuffle ([in] VARIANT_E

Set whether this track is skipped when shuffling.

Audio books are always skipped when shuffling.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

shouldExcludeFromShuffle True if this track should be skipped when shuffling.

Return values:

S_OK

ITUNES_E_OBJECTLOCKED
This track is not modifiable, or this track is an audio book.

ITUNES_E_OBJECTDELETED
This track has been deleted.

E FAIL

An unexpected error occurred.

HRESULT IITFileOrCDTrack::ExcludeFromShuffle ([out, retval] VAI

Returns true if this track is skipped when shuffling.

Audio books are always skipped when shuffling.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

excludeFromShuffle Returns true if this track is skipped when shuffling.

Return values:

S_OK The operation was successful.

E_POINTER excludeFromShuffle is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::Location ([in] BSTR location)

Set the full path to the file represented by this track. iTunes will not move or copy any existing file from the track's current location to the new location, the full path must point to an existing file that is playable by iTunes. Attempting to set the location of a track on CD will always fail with E INVALIDARG.

Note:

This property is available in iTunes 8.1 and later (iTunes type library 1.12 and later).

Parameters:

location The full path to the file represented by this track.

Return values:

S_OK The operation was successful.

E_INVALIDARG location is not a valid file (e.g. no

file at this location, the file can't be played by iTunes, the location is a directory, or the track is on

CD)

E POINTER location is NULL.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::Location ([out, retval] BSTR * locatio

Returns the full path to the file represented by this track.

Parameters:

location Returns the full path to the file represented by this track.

Return values:

S_OK The operation was successful.

S_FALSE The location could not be

retrieved (e.g. no file exists at

the expected location).

E POINTER location is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::LongDescription ([in] BSTR longDes

Sets the long description for the track.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

longDescription The new long description for the track.

Return values:

S_OK

E_POINTER

IngDescription is NULL.

ITUNES_E_OBJECTLOCKED

This track is not modifiable.

ITUNES_E_OBJECTDELETED

This track has been deleted.

E_FAIL

An unexpected error occurred.

HRESULT IITFileOrCDTrack::LongDescription ([out, retval] BSTR *

Returns the long description for the track.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

longDescription Returns the long description for the track.

Return values:

 S_OK The operation was successful. $E_POINTER$ longDescription is NULL. $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::Lyrics ([in] BSTR lyrics)

Sets the lyrics for the track.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

lyrics The new lyrics for the track.

Return values:

S_OK The operation was successful.

E_POINTER lyrics is NULL.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::Lyrics ([out, retval] BSTR * lyrics)

Returns the lyrics for the track.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

lyrics Returns the lyrics for the track.

Return values:

S_OK The operation was successful.

E_POINTER lyrics is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::PartOfGaplessAlbum ([in] VARIANT_I

Set whether this track is part of a gapless album.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

shouldBePartOfGaplessAlbum True if this track should be part of a gapless album.

Return values:

S_OK The operation was successful.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::PartOfGaplessAlbum ([out, retval] VA

Returns true if this track is part of a gapless album.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

partOfGaplessAlbum Returns true if this track is part of a gapless album.

Return values:

S_OK The operation was successful.

E_POINTER partOfGaplessAlbum is NULL.

ITUNES E OBJECTDELETED This track has been deleted.

HRESULT IITFileOrCDTrack::Playlists ([out, retval] IITPlaylistColle

Returns a collection of playlists that contain the song that this track represents.

This is the same collection of playlists that are shown in the "Show in Playlist" contextual menu for a track, plus the specific playlist that contains this track.

A track represents a song in a single playlist, use **IITTrack::Playlist()** to get the specific playlist that contains this track.

Note:

This property is available in iTunes 7.7 and later (iTunes type library 1.11 and later).

Parameters:

iPlaylistCollection Collection of IITPlaylist objects.

Return values:

 S_OK The operation was successful. $E_POINTER$ iPlaylistCollection is NULL. $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::Podcast ([out, retval] VARIANT_BOO

Returns true if this track is a podcast track.

Note:

This property is available in iTunes 4.9 and later (iTunes type

library 1.4 and later).

Parameters:

isPodcast Returns true if this track is a podcast track.

Return values:

S_OK The operation was successful.

E_POINTER is Podcast is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::RatingKind ([out, retval] ITRatingKinc

Returns the track rating kind. If the track rating has never been set, or has been set to 0, the kind is **ITRatingKindComputed**. Otherwise, the kind is **ITRatingKindUser**.

Note:

This property is available in iTunes 7.4 and later (iTunes type library 1.10 and later).

Parameters:

ratingKind Returns the track rating kind.

Return values:

S_OK The operation was successful.

E_POINTER ratingKind is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::ReleaseDate ([out, retval] DATE * rele

Returns release date of the track. This property is read-only.

A value of zero means no release date.

Note:

This property is available in iTunes 8.1 and later (iTunes type library 1.12 and later).

Parameters:

releaseDate Returns the release date of the track.

Return values:

E FAIL

S_OK The operation was successful.

E_POINTER releaseDate is NULL.

ITUNES E OBJECTDELETED This track has been deleted.

An unexpected error occurred.

HRESULT IITFileOrCDTrack::RememberBookmark ([in] VARIANT_I

Set whether the playback position is remembered for this track.

The playback position is always remembered for audio books.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

shouldRememberBookmark True if playback position should be remembered for this track.

Return values:

S_OK The operation was successful.

ITUNES_E_OBJECTLOCKED This track is not modifiable, or this track is an audio book.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::RememberBookmark ([out, retval] VA

Returns true if playback position is remembered for this track.

The playback position is always remembered for audio books.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

rememberBookmark Returns true if playback position is remembered for this track.

Return values:

 S_OK The operation was successful. $E_POINTER$ rememberBookmark is NULL. $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::Reveal ()

Reveals the track in the main browser window.

Note:

This method is available in iTunes 7.4 and later (iTunes type library 1.10 and later).

Return values:

S OK

The operation was successful.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::SeasonNumber ([in] long seasonNumber

Set the season number of the track.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

seasonNumber The new season number of the track.

Return values:

S_OK The operation was successful.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::SeasonNumber ([out, retval] long * s

Returns the season number of the track.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

seasonNumber Returns the season number of the track.

Return values:

S_OK The operation was successful.

E_POINTER seasonNumber is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::Show ([in] BSTR showName)

Set the show name of the track.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

showName The new show name of the track.

Return values:

S_OK

E_POINTER

ShowName is NULL.

ITUNES_E_OBJECTLOCKED

This track is not modifiable.

ITUNES_E_OBJECTDELETED

This track has been deleted.

E_FAIL

An unexpected error occurred.

HRESULT IITFileOrCDTrack::Show ([out, retval] BSTR * showNam

Returns the show name of the track.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

showName Returns the show name of the track.

Return values:

S_OK The operation was successful.

E_POINTER showName is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::Size64High ([out, retval] long * sizeH

Returns the high 32-bits of the size of the track (in bytes).

You can use IITTrack::Size() if you know the size of the track is less than 4GB. Otherwise, use IITFileOrCDTrack::Size64High() and IITFileOrCDTrack::Size64Low() to determine the full 64-bit size.

Note:

This property is available in iTunes 7.1 and later (iTunes type library 1.9 and later).

Parameters:

sizeHigh Returns the high 32-bits of the size of the track (in bytes).

Return values:

S_OK The operation was successful.

E_POINTER sizeHigh is NULL.

 $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::Size64Low ([out, retval] long * sizeLc

Returns the low 32-bits of the size of the track (in bytes).

You can use **IITTrack::Size()** if you know the size of the track is less than 4GB. Otherwise, use **IITFileOrCDTrack::Size64High()** and **IITFileOrCDTrack::Size64Low()** to determine the full 64-bit size.

Note:

This property is available in iTunes 7.1 and later (iTunes type library 1.9 and later).

Parameters:

sizeLow Returns the low 32-bits of the size of the track (in bytes).

Return values:

S_OK The operation was successful.

E_POINTER sizeLow is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::SkippedCount ([in] long skippedCou

Set the number of times the track has been skipped. This property cannot be set if the track is not playable (e.g. a PDF file).

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

skippedCount The new number of times the track has been skipped.

Return values:

S_OK The operation was successful.

ITUNES E OBJECTLOCKED This track is not modifiable.

ITUNES E OBJECTDELETED This track has been deleted.

HRESULT IITFileOrCDTrack::SkippedCount ([out, retval] long * sk

Returns the number of times the track has been skipped.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

skippedCount Returns the number of times the track has been skipped.

Return values:

 S_OK The operation was successful. $E_POINTER$ skippedCount is NULL. $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::SkippedDate ([in] DATE skippedDate

Set the date and time the track was last skipped. This property cannot be set if the track is not playable (e.g. a PDF file).

A value of zero means no skipped date.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

skippedDate The new date and time the track was last skipped.

Return values:

S_OK The operation was successful.

E_INVALIDARG skippedDate is invalid.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES E OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::SkippedDate ([out, retval] DATE * ski

Returns the date and time the track was last skipped.

A value of zero means no skipped date.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

skippedDate Returns the date and time the track was last skipped.

Return values:

S_*OK* The operation was successful.

E_POINTER skippedDate is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::SortAlbum ([in] BSTR album)

Set the name of the album used for sorting. If a sort album is specified, iTunes will use this string instead of the real album name when sorting.

Note:

This property is available in iTunes 7.1 and later (iTunes type library 1.9 and later).

Parameters:

album The new sort name of the album.

Return values:

S_OK The operation was successful.

E POINTER album is NULL.

 $\begin{array}{ll} \textit{ITUNES_E_OBJECTLOCKED} & \text{This track is not modifiable.} \\ \textit{ITUNES_E_OBJECTDELETED} & \text{This track has been deleted.} \\ \textit{E_FAIL} & \text{An unexpected error occurred.} \\ \end{array}$

HRESULT IITFileOrCDTrack::SortAlbum ([out, retval] BSTR * albui

Returns the name of the album used for sorting. If a sort album is specified, iTunes will use this string instead of the real album name when sorting.

Note:

This property is available in iTunes 7.1 and later (iTunes type library 1.9 and later).

Parameters:

album Returns the name of the album used for sorting.

Return values:

S_OK The operation was successful.

E_POINTER album is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::SortAlbumArtist ([in] BSTR albumAr

Set the name of the album artist used for sorting. If a sort album artist is specified, iTunes will use this string instead of the real album artist when sorting.

Note:

This property is available in iTunes 7.1 and later (iTunes type library 1.9 and later).

Parameters:

albumArtist The new name of the album artist used for sorting.

Return values:

S_OK

E_POINTER

ITUNES_E_OBJECTLOCKED

ITUNES_E_OBJECTDELETED

This track is not modifiable.

This track has been deleted.

An unexpected error occurred.

HRESULT IITFileOrCDTrack::SortAlbumArtist ([out, retval] BSTR *

Returns the name of the album artist used for sorting. If a sort album artist is specified, iTunes will use this string instead of the real album artist when sorting.

Note:

This property is available in iTunes 7.1 and later (iTunes type library 1.9 and later).

Parameters:

albumArtist Returns the name of the album artist used for sorting.

Return values:

S_OK The operation was successful.

E_POINTER albumArtist is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::SortArtist ([in] BSTR artist)

Set the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting.

Note:

This property is available in iTunes 7.1 and later (iTunes type library 1.9 and later).

Parameters:

artist The new sort artist for the track.

Return values:

S_*OK* The operation was successful.

E_POINTER artist is NULL.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::SortArtist ([out, retval] BSTR * artist

Returns the name of the artist used for sorting. If a sort artist is specified, iTunes will use this string instead of the real artist when sorting.

Note:

This property is available in iTunes 7.1 and later (iTunes type

library 1.9 and later).

Parameters:

artist Returns the name of the artist used for sorting.

Return values:

S OK The operation was successful.

E POINTER artist is NULL.

ITUNES E OBJECTDELETED This track has been deleted. E FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::SortComposer ([in] BSTR composer

Set the composer used for sorting. If a sort composer is specified, iTunes will use this string instead of the real composer when sorting.

Note:

This property is available in iTunes 7.1 and later (iTunes type library 1.9 and later).

Parameters:

composer The new sort composer for the track.

Return values:

The operation was successful. S OK

E POINTER composer is NULL.

ITUNES E OBJECTLOCKED This track is not modifiable. ITUNES E OBJECTDELETED This track has been deleted.

E FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::SortComposer ([out, retval] BSTR * c

Returns the composer used for sorting. If a sort composer is specified, iTunes will use this string instead of the real composer when sorting.

Note:

This property is available in iTunes 7.1 and later (iTunes type library 1.9 and later).

Parameters:

composer Returns the composer used for sorting.

Return values:

S_OK The operation was successful.

E_POINTER composer is NULL.

 $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::SortName ([in] BSTR name)

Set the name of the track used for sorting. If a sort name is specified, iTunes will use this string instead of the real track name when sorting.

Note:

This property is available in iTunes 7.1 and later (iTunes type library 1.9 and later).

Parameters:

name The new sort name for the track.

Return values:

S_OK The operation was successful.

E_POINTER name is NULL.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::SortName ([out, retval] BSTR * name

Returns the name of the track used for sorting. If a sort name is specified, iTunes will use this string instead of the real track name when sorting.

Note:

This property is available in iTunes 7.1 and later (iTunes type library 1.9 and later).

Parameters:

name Returns the name of the track used for sorting.

Return values:

S_OK The operation was successful.

E_POINTER name is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::SortShow ([in] BSTR showName)

Set the show name used for sorting. If a sort show is specified, iTunes will use this string instead of the real show name when sorting.

Note:

This property is available in iTunes 7.1 and later (iTunes type library 1.9 and later).

Parameters:

showName The new sort show name of the track.

Return values:

S_OK The operation was successful.

E_POINTER showName is NULL.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::SortShow ([out, retval] BSTR * show

Returns the show name used for sorting. If a sort show is specified, iTunes will use this string instead of the real show name when sorting.

Note:

This property is available in iTunes 7.1 and later (iTunes type library 1.9 and later).

Parameters:

showName Returns the sort show name of the track.

Return values:

S_OK The operation was successful.

E_POINTER showName is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::Unplayed ([in] VARIANT_BOOL shou

Set whether this track has been played.

Note:

This property is available in iTunes 7.1 and later (iTunes type library 1.9 and later).

Parameters:

shouldBeUnplayed True if this track should marked as unplayed.

Return values:

S_OK The operation was successful.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::Unplayed ([out, retval] VARIANT_BO(

Returns false if this track has been played.

Note:

This property is available in iTunes 7.1 and later (iTunes type library 1.9 and later).

Parameters:

isUnplayed Returns false if this track has been played.

Return values:

 S_OK The operation was successful. $E_POINTER$ is Unplayed is NULL. $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::UpdateInfoFromFile ()

Update this track's information with the information stored in its file.

If you have edited this track's metadata (such as song name, album, artist, etc.) with another program outside of iTunes, you can call this method to force iTunes to update its internal information with the

current information stored in the track's file.

Note:

This method is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Return values:

S_OK The operation was successful.

S_FALSE The file could not be found (e.g. no file exists at the expected

location).

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITFileOrCDTrack::UpdatePodcastFeed()

Update the podcast feed for this track. This is equivalent to the user choosing Update Podcast from the contextual menu for the podcast feed that contains this track.

Note:

This method is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Return values:

S_OK The operation was

successful.

E_INVALIDARG This track is not a podcast.

ITUNES_E_PODCASTSDISABLED Access to podcasts is

disabled in the preferences.

E_FAIL An unexpected error

occurred.

HRESULT IITFileOrCDTrack::VideoKind ([in] ITVideoKind videoKin

Set the video track kind. Note that the video kind cannot be changed for purchased content.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

videoKind The new video kind of the track.

Return values:

S_OK	The operation was successful.
E_INVALIDARG	Invalid video kind specified, or
	the track is not a video track.
ITUNES_E_OBJECTLOCKED	This track is not modifiable or is purchased content.
ITUNES_E_OBJECTDELETED	This track has been deleted.
E FAIL	An unexpected error occurred.

HRESULT IITFileOrCDTrack::VideoKind ([out, retval] ITVideoKind '

Returns the video track kind. This will be **ITVideoKindNone** if the track is not a video track.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

videoKind Returns the video kind of the track.

Return values:

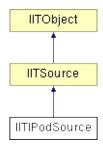
 S_OK The operation was successful. $E_POINTER$ videoKind is NULL. $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITIPodSource Interface Reference

Inheritance diagram for IITIPodSource:



List of all members.

Detailed Description

Represents an iPod source.

Public Member Functions

Methods

HRESULT	UpdatelPod ()
	Update the contents of the iPod.
HRESULT	EjectlPod ()
	Eject the iPod.

Properties

HRESULT	SoftwareVersion ([out, retval] BSTR *softwareVersion)	
	Returns the version of the iPod's software, formatted as a string (e.g. "1.0" or "3.0.2").	

Member Function Documentation

HRESULT IITIPodSource::EjectlPod ()

Eject the iPod.

Note:

This property is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Return values:

S_OK
The operation was successful.

S_FALSE
The iPod could not be ejected.

ITUNES_E_OBJECTDELETED
This iPod source is no longer available.

E_FAIL
An unexpected error occurred.

HRESULT IITIPodSource::SoftwareVersion ([out, retval] BSTR * sc

Returns the version of the iPod's software, formatted as a string (e.g. "1.0" or "3.0.2").

Note:

This property is available in iTunes 4.8 and later (iTunes type library 1.3 and later).

Parameters:

software Version Returns the version of the iPod's software.

Return values:

S_OK The operation was successful. E_POINTER softwareVersion is NULL. ITUNES_E_OBJECTDELETED This iPod source is no longer available.

E_FAIL An unexpected error occurred.

HRESULT IITIPodSource::UpdatelPod ()

Update the contents of the iPod.

You can only update an iPod if it is configured for automatic updating.

Return values:

S_OK The operation was successful.

S_FALSE The iPod could not be updated

(e.g. it isn't configured for

automatic updating).

ITUNES_E_OBJECTDELETED This iPod source is no longer

available.

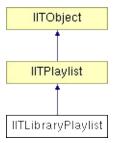
E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 Apple Computer, Inc.

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITLibraryPlaylist Interface Reference

Inheritance diagram for IITLibraryPlaylist:



List of all members.

Detailed Description

Represents a library playlist.

A library playlist consists of all the tracks in a user's library.

For convenience, you can retrieve the main library playlist using **liTunes::LibraryPlaylist()**.

Public Member Functions

Methods

HRESULT	AddFile ([in] BSTR filePath,[out, retval] IITOperationStatus **iStatus) Add a file or files inside a folder to the playlist.
HRESULT	AddFiles ([in] VARIANT *filePaths,[out, retval] IITOperationStatus **iStatus) Add an array of files and/or folders to the playlist.
HRESULT	AddURL ([in] BSTR url,[out, retval] IITURLTrack **iURLTrack) Add a streaming audio URL to the playlist.
HRESULT	AddTrack ([in] VARIANT *iTrackToAdd,[out, retval] IITTrack **iAddedTrack) Add an existing track to the playlist.

Member Function Documentation

```
HRESULT IITLibraryPlaylist::AddFile ( [in] BSTR [out, retval] IITOperationState)
```

Add a file or files inside a folder to the playlist.

You cannot use this method to add a file that requires conversion to be added (e.g. a CD track), use **liTunes::ConvertFile()** or **liTunes::ConvertFile2()** instead. If you add a folder that contains files that require conversion, they will be skipped.

Parameters:

filePath The full path to the file or folder to add.

iStatus Returns an IITOperationStatus object corresponding to

the asynchronous operation. If an error occurs, or no files

were added, this will be set to NULL.

Return values:

S_OK The operation was successful.

E_POINTER filePath or iStatus is NULL.

E_INVALIDARG filePath is empty or too long, or

refers to a file that would require

conversion.

ITUNES_E_OBJECTLOCKED This playlist is read-only.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITLibraryPlaylist::AddFiles ([in] VARIANT *

[out, retval] IITOperationSta

)

Add an array of files and/or folders to the playlist.

You cannot use this method to add a file that requires conversion to be added (e.g. a CD track), use **liTunes::ConvertFile()** or **liTunes::ConvertFile2()** instead. If you add a folder that contains files that require conversion, they will be skipped.

Parameters:

filePaths The single-dimensioned array of full paths to each file or

folder. filePaths can be of type

VT_ARRAY|VT_VARIANT, where each entry is a

VT_BSTR, or VT_ARRAY|VT_BSTR. You can also pass

a JScript Array object.

iStatus Returns an IITOperationStatus object corresponding to

the asynchronous operation. If an error occurs, or no

files were added, this will be set to NULL.

Return values:

S_OK The operation was successful.

E_POINTER filePaths or iStatus is NULL.

E_INVALIDARG filePaths is the wrong VARIANT

type, or includes a file that would

require conversion.

ITUNES_E_OBJECTLOCKED This playlist is read-only.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITLibraryPlaylist::AddTrack ([in] VARIANT * iTrack ([in] VARIANT * iTrack ** iAccount ([in

Add an existing track to the playlist.

You cannot use this method to add a CD track (ITTrackKindCD) to another playlist, use IiTunes::ConvertTrack() instead. or IiTunes::ConvertTrack2() instead.

You cannot add a shared library track (ITTrackKindSharedLibrary) to another playlist.

Parameters:

iTrackToAdd The track to add. This is a VARIANT of type VT DISPATCH that points to an **IITTrack**.

iAddedTrack Returns an **IITTrack** object corresponding to the new track.

Return values:

S_OK The operation was successful.

E POINTER iTrackToAdd or iAddedTrack is

NULL.

E INVALIDARG iTrackToAdd is the wrong

VARIANT type, or is a CD or

shared track.

ITUNES_E_OBJECTLOCKED This playlist is read-only.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITLibraryPlaylist::AddURL ([in] BSTR [out, retval] IITURLTrack **
)

Add a streaming audio URL to the playlist.

Parameters:

url The URL to add. The length of the URL can be 255

characters or less.

iURLTrack Returns an **IITURLTrack** object corresponding to the new track.

Return values:

S_OK The operation was successful.

E_POINTERurl or iURLTrack is NULL.E_INVALIDARGurl is empty or too long.ITUNES_E_OBJECTLOCKEDThis playlist is read-only.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

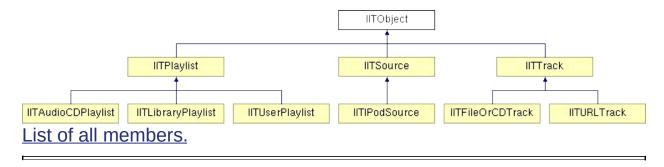
E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITObject Interface Reference

Inheritance diagram for IITObject:



Detailed Description

Defines a source, playlist or track.

An IITObject uniquely identifies a source, playlist, or track in iTunes using four separate IDs. These are runtime IDs, they are only valid while the current instance of iTunes is running.

As of iTunes 7.7, you can also identify an IITObject using a 64-bit persistent ID, which is valid across multiple invocations of iTunes.

The main use of the IITObject interface is to allow clients to track iTunes database changes using

_liTunesEvents::OnDatabaseChangedEvent().

You can retrieve an IITObject with a specified runtime ID using **IiTunes::GetITObjectByID()**.

An IITObject will always have a valid, non-zero source ID.

An IITObject corresponding to a playlist or track will always have a valid playlist ID. The playlist ID will be zero for a source.

An IITObject corresponding to a track will always have a valid track and track database ID. These IDs will be zero for a source or playlist.

A track ID is unique within the track's playlist. A track database ID is unique across all playlists. For example, if the same music file is in two different playlists, each of the tracks could have different track IDs, but they will have the same track database ID.

An IITObject also has a 64-bit persistent ID which can be used to identify the IITObject across multiple invocations of iTunes. Note that persistent IDs are only really persistent for objects that iTunes stores in its database. For example, the persistent IDs for transient objects like radio streams and objects from audio CDs will differ across multiple invocations of iTunes.

The persistent ID was exposed in the COM interface starting with iTunes 7.7, so methods and properties to retrieve the persistent ID could not be added to the IITObject interface without breaking existing COM clients. You can find these routines in **liTunes** instead:

- liTunes::GetITObjectPersistentIDs()
 liTunes::ITObjectPersistentIDHigh()
- IiTunes::ITObjectPersistentIDLow()

You can retrieve a source, playlist, or track with a specified persistent ID using the ItemByPersistentID property of the appropriate collection interface:

- IITSourceCollection::ItemByPersistentID()
- IITPlaylistCollection::ItemByPersistentID()
- IITTrackCollection::ItemByPersistentID()

Public Member Functions

Methods

HRESULT	GetITObjectIDs ([out] long *sourceID,[out] long *playlistID,[out] long *trackID,[out] long *databaseID)
	Returns the four IDs that uniquely identify this object.

Properties

HRESULT	Name ([out, retval] BSTR *name)
	Returns the name of the object.
HRESULT	Name ([in] BSTR name)
	Set the name of the object.
HRESULT	Index ([out, retval] long *index)
	Returns the index of the object in internal application order.
HRESULT	SourceID ([out, retval] long *sourceID)
	Returns the ID that identifies the source.
HRESULT	PlaylistID ([out, retval] long *playlistID)
	Returns the ID that identifies the playlist.
HRESULT	TrackID ([out, retval] long *trackID)
	Returns the ID that identifies the track within the playlist.
HRESULT	TrackDatabaseID ([out, retval] long *databaseID)
	Returns the ID that identifies the track, independent of its playlist.

Member Function Documentation

Returns the four IDs that uniquely identify this object.

This method is not usable from scripting clients, since it returns multiple output parameters. Scripting clients must use the IITObject::SourceID(), IITObject::PlaylistID(), IITObject::TrackID(), and IITObject::TrackDatabaseID() properties instead.

These are runtime IDs, they are only valid while the current instance of iTunes is running.

Parameters:

sourceID Returns the ID that identifies the source. Valid for a

source, playlist, or track.

playlistID Returns the ID that identifies the playlist. Valid for a

playlist or track. Will be zero for a source.

trackID Returns the ID that identifies the track within the

playlist. Valid for a track. Will be zero for a source or

playlist.

databaseID Returns the ID that identifies the track, independent

of its playlist. Valid for a track. Will be zero for a

source or playlist.

Return values:

S_OK The operation was successful.

E POINTER sourceID, playlistID, trackID, or

databaseID is NULL.

ITUNES E OBJECTDELETED The source, playlist, or track

corresponding to this object has

been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITObject::Index ([out, retval] long * index)

Returns the index of the object in internal application order.

You can pass this index to IITSourceCollection::Item(), IITPlaylistCollection::Item(), or IITTrackCollection::Item() to retrieve this object.

For tracks, this index is independent of play order. The play order index of a track can be retrieved using **IITTrack::PlayOrderIndex()**.

Parameters:

index Returns the index of the object in internal application order.

Return values:

S_OK The operation was successful.

E POINTER index is NULL.

ITUNES_E_OBJECTDELETED The source, playlist, or track

corresponding to this object has

been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITObject::Name ([in] BSTR name)

Set the name of the object.

You can only change the name of a track or playlist. You cannot change

the name of a source.

Object names are not unique in iTunes, e.g. multiple playlists may share the same name.

Parameters:

name The new name of the object.

Return values:

S_OK The operation was successful.

E_POINTER name is NULL or empty.

ITUNES_E_OBJECTLOCKED The source, playlist, or track

cannot be renamed.

ITUNES_E_OBJECTDELETED The source, playlist, or track

corresponding to this object has

been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITObject::Name ([out, retval] BSTR * name)

Returns the name of the object.

For a source, this is the name of the source (as displayed in the Source list).

For a playlist, this is the name of the playlist.

For a track, this is the name of the track.

Object names are not unique in iTunes, e.g. multiple playlists may share the same name.

Parameters:

name Returns the name of the object.

Return values:

S_OK The operation was successful.

E_POINTER name is NULL.

ITUNES_E_OBJECTDELETED The source, playlist, or track

corresponding to this object has

been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITObject::PlaylistID ([out, retval] long * playlistID)

Returns the ID that identifies the playlist.

Valid for a playlist or track. Will be zero for a source.

This is a runtime ID, it is only valid while the current instance of iTunes is running.

Parameters:

playlistID Returns the ID that identifies the playlist.

Return values:

S_OK The operation was successful.

E_POINTER playlistID is NULL. *ITUNES_E_OBJECTDELETED* The playlist or track

corresponding to this object has

been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITObject::SourceID ([out, retval] long * sourceID)

Returns the ID that identifies the source.

Valid for a source, playlist, or track.

This is a runtime ID, it is only valid while the current instance of iTunes is running.

Parameters:

sourceID Returns the ID that identifies the source.

Return values:

S_OK The operation was successful.

E POINTER sourceID is NULL.

ITUNES E OBJECTDELETED The source, playlist, or track

corresponding to this object has

been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITObject::TrackDatabaseID ([out, retval] long * databas

Returns the ID that identifies the track, independent of its playlist.

Valid for a track. Will be zero for a source or playlist. If the same music file is in two different playlists, each of the tracks will have the same track database ID.

This is a runtime ID, it is only valid while the current instance of iTunes is running.

Parameters:

databaseID Returns the ID that identifies the track, independent of its playlist.

Return values:

S_OK The operation was successful.

E_POINTER databaseID is NULL.

ITUNES_E_OBJECTDELETED The track corresponding to this

E FAIL

object has been deleted.
An unexpected error occurred.

HRESULT IITObject::TrackID ([out, retval] long * trackID)

Returns the ID that identifies the track within the playlist.

Valid for a track. Will be zero for a source or playlist.

This is a runtime ID, it is only valid while the current instance of iTunes is running.

Parameters:

trackID Returns the ID that identifies the track within the playlist.

Return values:

S_*OK* The operation was successful.

E POINTER trackID is NULL.

ITUNES_E_OBJECTDELETED The track corresponding to this

object has been deleted.

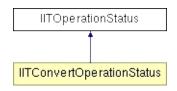
E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITOperationStatus Interface Reference

Inheritance diagram for IITOperationStatus:



List of all members.

Detailed Description

Represents the status of an asynchronous add or convert operation.

When a track is added using IITLibraryPlaylist::AddFile(), IITLibraryPlaylist::AddFiles(), IITUserPlaylist::AddFile(), or IITUserPlaylist::AddFiles(), the add may not complete immediately if iTunes needs to make a copy of the file.

Similarly, when converting or importing a file or track using IiTunes::ConvertFile(), IiTunes::ConvertFiles(), IiTunes::ConvertTrack() or IiTunes::ConvertTracks(), the conversion will never complete immediately.

These methods return an **IITOperationStatus** object, which can be polled to determine when the operation is done. This object will also return the collection of newly added or converted tracks.

As of version 1.1 of the iTunes type library, you should use IiTunes::ConvertFile2(), IiTunes::ConvertFiles2(), IiTunes::ConvertTrack2() or IiTunes::ConvertTracks2() instead of the original convert methods. These new methods return an IITConvertOperationStatus object to allow clients to retrieve additional conversion progress information.

Public Member Functions

Properties

HRESULT	InProgress ([out, retval] VARIANT_BOOL *isInProgress)
	Returns true if the operation is still in progress.
HRESULT	Tracks ([out, retval] IITTrackCollection **iTrackCollection)
	Returns a collection containing the tracks that were generated by the operation.

Member Function Documentation

HRESULT IITOperationStatus::InProgress ([out, retval] VARIANT_I

Returns true if the operation is still in progress.

You cannot retrieve the **IITOperationStatus::Tracks()** property until the operation completes.

Parameters:

isInProgress Returns true if the operation is still in progress.

Return values:

S_OK The operation was successful.

E_POINTER isInProgress is NULL.

ITUNES_E_OBJECTDELETED The playlist containing the tracks

has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITOperationStatus::Tracks ([out, retval] IITTrackCollecti

Returns a collection containing the tracks that were generated by the operation.

You cannot retrieve this property until

IITOperationStatus::InProgress() returns false

Parameters:

iTrackCollection Returns a collection containing the tracks that were generated by the operation.

Return values:

S_OK The operation was successful.

E_POINTER iTrackCollection is NULL.

S_FALSE The operation is still in progress,

or no tracks were generated. This can happen, for example, if

the user cancels a convert

operation.

ITUNES_E_OBJECTDELETED The playlist containing the tracks

has been deleted.

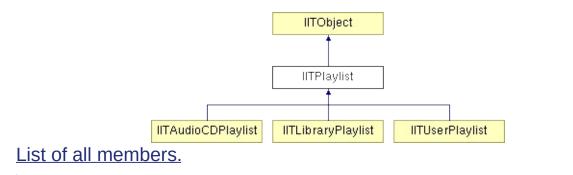
E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITPlaylist Interface Reference

Inheritance diagram for IITPlaylist:



Detailed Description

Represents a playlist.

A playlist is always associated with an **IITSource**.

You can retrieve all the playlists defined for a source using **IITSource::Playlists()**.

For convenience, you can retrieve the main library playlist using **liTunes::LibraryPlaylist()**.

You can create a new playlist using IiTunes::CreatePlaylist().

Public Member Functions

Methods

HRESULT	Delete () Delete this playlist.
HRESULT	PlayFirstTrack () Start playing the first track in this playlist.
HRESULT	Print ([in] VARIANT_BOOL showPrintDialog,[in] ITPlaylistPrintKind printKind,[in] BSTR theme)
HRESULT	Print this playlist. Search ([in] BSTR searchText,[in] ITPlaylistSearchField searchFields,[out,
TIRESOLI	retval] IITTrackCollection **iTrackCollection) Returns a collection containing the tracks with the specified text.

Properties

HRESULT	Kind ([out, retval] ITPlaylistKind *kind)
	Returns the kind of the playlist.
HRESULT	Source ([out, retval] IITSource **iSource)
	Returns an IITSource object corresponding to the source that contains the playlist.
HRESULT	Duration ([out, retval] long *duration)
	Returns the total length of all songs in the playlist (in seconds).
HRESULT	Shuffle ([out, retval] VARIANT_BOOL *isShuffle)
	Returns true if songs in the playlist are played in random order.
HRESULT	Shuffle ([in] VARIANT_BOOL shouldShuffle)
	Set whether songs in the playlist should be played in random order.
HRESULT	Size ([out, retval] double *size)
	Returns the total size of all songs in the playlist (in bytes).
HRESULT	SongRepeat ([out, retval] ITPlaylistRepeatMode *repeatMode)
	Returns the playback repeat mode.
HRESULT	SongRepeat ([in] ITPlaylistRepeatMode repeatMode)
	Sets the playback repeat mode.
HRESULT	Time ([out, retval] BSTR *time)
	Returns the total length of all songs in the playlist (in MM:SS format).
HRESULT	(L , 1 =)
	Returns true if the playlist is visible in the Source list.

HRESULT **Tracks** ([out, retval] **IITTrackCollection** **iTrackCollection)

Returns a collection containing the tracks in this playlist.

Member Function Documentation

HRESULT IITPlaylist::Delete ()

Delete this playlist.

Return values:

S_OK The operation was successful. ITUNES_E_OBJECTDELETED This playlist has been deleted already.

E_FAIL An unexpected error occurred.

HRESULT IITPlaylist::Duration ([out, retval] long * duration)

Returns the total length of all songs in the playlist (in seconds).

Parameters:

duration Returns the total length of all songs in the playlist (in seconds).

Return values:

S_OK The operation was successful.

S_FALSE The playlist does not have a total

duration.

E_POINTER duration is NULL.

 $ITUNES_E_OBJECTDELETED$ This playlist has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITPlaylist::Kind ([out, retval] ITPlaylistKind * kind)

Returns the kind of the playlist.

Parameters:

kind Returns the kind of the playlist.

Return values:

S_*OK* The operation was successful.

E_POINTER kind is NULL.

 $ITUNES_E_OBJECTDELETED$ This playlist has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITPlaylist::PlayFirstTrack()

Start playing the first track in this playlist.

Return values:

 S_OK The operation was successful. ITUNES_E_OBJECTDELETED This playlist has been deleted. E_FAIL An unexpected error occurred.

Print this playlist.

Parameters:

showPrintDialog If true, display the print dialog.

printKind The printout kind.

theme The name of the theme to use. This

corresponds to the name of a Theme combo box item in the print dialog for the specified

printKind (e.g. "Track length").

This string cannot be longer than 255 characters, but it may be NULL or empty.

Return values:

S_OK The operation was successful.

ITUNES_E_USERCANCEL The operation was canceled by

the user.

E_INVALIDARG printKind is invalid or theme is

too long.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

```
HRESULT IITPlaylist::Search ( [in] BSTR searchField searchField [out, retval] IITTrackCollection ** iTr
```

Returns a collection containing the tracks with the specified text.

If searchFields is ITPlaylistSearchFieldVisible, this is identical to the list of tracks displayed if the user enters the search text in the Search edit field in iTunes when this playlist is being displayed.

Parameters:

searchText The text to search for. This string cannot be

longer than 255 characters.

searchFields Specifies which fields of each track should be

searched for searchText.

iTrackCollection Collection of IITTrack objects. This will be NULL

if no tracks meet the search criteria.

Return values:

S_OK The operation was successful.

E_POINTER iTrackCollection is NULL. *E_INVALIDARG* searchFields is invalid, or

searchText is empty or too long.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E FAIL An unexpected error occurred.

HRESULT IITPlaylist::Shuffle ([in] VARIANT_BOOL shouldShuffle

Set whether songs in the playlist should be played in random order.

Parameters:

shouldShuffle True if songs in the playlist should be played in random order.

Return values:

S_OK The operation was successful.

S_FALSE This playlist does not support

shuffle mode.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITPlaylist::Shuffle ([out, retval] VARIANT_BOOL * isShi

Returns true if songs in the playlist are played in random order.

Parameters:

isShuffle Returns true if songs in the playlist are played in random order.

Return values:

S_OK The operation was successful.

E_POINTER is Shuffle is NULL.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITPlaylist::Size ([out, retval] double * size)

Returns the total size of all songs in the playlist (in bytes).

Parameters:

size Returns the total size of all songs in the playlist (in bytes).

Return values:

S_OK The operation was successful.

S_FALSE The playlist does not have a total

size.

E POINTER size is NULL.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITPlaylist::SongRepeat ([in] ITPlaylistRepeatMode repe

Sets the playback repeat mode.

Parameters:

repeatMode The new playback repeat mode.

Return values:

S_OK The operation was successful.

S_FALSE This playlist does not support

repeat mode.

 $ITUNES_E_OBJECTDELETED$ This playlist has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITPlaylist::SongRepeat ([out, retval] ITPlaylistRepeatMc

Returns the playback repeat mode.

Parameters:

repeatMode Returns the playback repeat mode.

Return values:

S_OK The operation was successful.

E_POINTER repeatMode is NULL.

 $ITUNES_E_OBJECTDELETED$ This playlist has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITPlaylist::Source ([out, retval] IITSource ** iSource)

Returns an **IITSource** object corresponding to the source that contains the playlist.

Parameters:

iSource Returns an **IITSource** object corresponding to the source that contains the playlist.

Return values:

S_OK The operation was successful.

E_POINTER iSource is NULL.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITPlaylist::Time ([out, retval] BSTR * time)

Returns the total length of all songs in the playlist (in MM:SS format).

Parameters:

time Returns the total length of all songs in the playlist (in MM:SS format).

Return values:

S_OK The operation was successful.

E_POINTER time is NULL.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E FAIL An unexpected error occurred.

HRESULT IITPlaylist::Tracks ([out, retval] IITTrackCollection ** iTr

Returns a collection containing the tracks in this playlist.

Parameters:

iTrackCollection Collection of IITTrack objects.

Return values:

S_OK The operation was successful.

E_POINTER iTrackCollection is NULL.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E FAIL An unexpected error occurred.

HRESULT IITPlaylist::Visible ([out, retval] VARIANT_BOOL * isVisi

Returns true if the playlist is visible in the Source list.

Parameters:

isVisible Returns true if the playlist is visible in the Source list.

Return values:

S_*OK* The operation was successful.

E_POINTER isVisible is NULL.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITPlaylistCollection Interface Reference

List of all members.

Detailed Description

Represents a collection of playlist objects.

Note that collection indices are always 1-based.

You can retrieve all the playlists defined for a source using **IITSource::Playlists()**.

Public Member Functions

Properties

HRESULT	Count ([out, retval] long *count) Returns the number of playlists in the collection.
HRESULT	Item ([in] long index,[out, retval] IITPlaylist **iPlaylist) Returns an IITPlaylist object corresponding to the given index (1-based).
HRESULT	ItemByName ([in] BSTR name,[out, retval] IITPlaylist **iPlaylist) Returns an IITPlaylist object with the specified name.
HRESULT	_NewEnum ([out, retval] IUnknown **iEnumerator) Returns an IEnumVARIANT object which can enumerate the collection.
HRESULT	ItemByPersistentID ([in] long highID,[in] long lowID,[out, retval] IITPlaylist **iPlaylist)
	Returns an IITPlaylist object with the specified persistent ID. See the documentation on IITObject for more information on persistent IDs.

Member Function Documentation

HRESULT IITPlaylistCollection::_NewEnum ([out, retval] IUnknown

Returns an IEnumVARIANT object which can enumerate the collection.

Note:

This property is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Parameters:

iEnumerator Returns an IEnumVARIANT object which can enumerate the collection.

Return values:

S_OK The operation was successful.

E POINTER iEnumerator is NULL.

E_FAIL An unexpected error occurred.

HRESULT IITPlaylistCollection::Count ([out, retval] long * count)

Returns the number of playlists in the collection.

Parameters:

count Returns the number of playlists in the collection.

Return values:

S_OK The operation was successful.

E_POINTER count is NULL.

ITUNES_E_OBJECTDELETED The source containing the

playlists has been deleted.

E_FAIL An unexpected error occurred.

```
HRESULT IITPlaylistCollection::Item ( [in] long ind [out, retval] IITPlaylist ** iPlate | iPl
```

Returns an **IITPlaylist** object corresponding to the given index (1-based).

Parameters:

index Index of the playlist to retrieve, must be less than or

equal to IITPlaylistCollection::Count().

iPlaylist Returns an **IITPlaylist** object corresponding to the given

index. Will be set to NULL if no playlist could be

retrieved.

Return values:

S *OK* The operation was successful.

S_FALSE index is too high.

E INVALIDARG index is less than one.

E_POINTER iPlaylist is NULL.

ITUNES_E_OBJECTDELETED The source containing the

playlists has been deleted.

E_FAIL An unexpected error occurred.

```
HRESULT IITPlaylistCollection::ItemByName ( [in] BSTR [out, retval] IITPlayli )
```

Returns an **IITPlaylist** object with the specified name.

This property will return the first playlist with the specified name, but there can be additional playlists with the same name, since playlist names are not unique. To determine all the playlists with the same name, you should use the **IITPlaylistCollection::Item()** property to enumerate the entire collection.

Parameters:

name The name of the playlist to retrieve.

iPlaylist Returns an IITPlaylist object with the specified name.

Will be set to NULL if no playlist could be retrieved.

Return values:

S_OK The operation was successful.
S_FALSE A playlist with the specified name

could not be found.

 $E_INVALIDARG$ name is empty.

E_POINTER name or iPlaylist is NULL. *ITUNES_E_OBJECTDELETED* The source containing the

playlists has been deleted.

E_FAIL An unexpected error occurred.

Returns an **IITPlaylist** object with the specified persistent ID. See the documentation on **IITObject** for more information on persistent IDs.

Note:

This property is available in iTunes 7.7 and later (iTunes type library 1.11 and later).

Parameters:

highID The high 32 bits of the 64-bit persistent ID.

lowID The low 32 bits of the 64-bit persistent ID.

iPlaylist Returns an IITPlaylist object with the specified

persistent ID. Will be set to NULL if no playlist could be

retrieved.

Return values:

S_OK The operation was successful.

S_FALSE A playlist with the specified

persistent ID could not be found.

E_POINTER iPlaylist is NULL.

ITUNES_E_OBJECTDELETED The source containing the

playlists has been deleted.

E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITPlaylistWindow Interface Reference

Inheritance diagram for IITPlaylistWindow:



List of all members.

Detailed Description

Represents a playlist window.

Public Member Functions

Properties

HRESULT	SelectedTracks ([out, retval] IITTrackCollection **iTrackCollection)
	Returns a collection containing the currently selected track or tracks.
HRESULT	Playlist ([out, retval] IITPlaylist **iPlaylist)
	Returns the playlist being displayed by this window.

Member Function Documentation

HRESULT IITPlaylistWindow::Playlist ([out, retval] IITPlaylist ** iPl

Returns the playlist being displayed by this window.

Parameters:

iPlaylist Returns the playlist being displayed by this window.

Return values:

S_OK The operation was successful.

E_POINTER iPlaylist is NULL.

ITUNES_E_OBJECTDELETED The window has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITPlaylistWindow::SelectedTracks ([out, retval] IITTrack

Returns a collection containing the currently selected track or tracks.

Parameters:

iTrackCollection Collection of **IITTrack** objects. Will be set to NULL if there is no current selection.

Return values:

S_OK The operation was successful.

S_FALSE There is no current selection.

E_POINTER iTrackCollection is NULL.

ITUNES_E_OBJECTDELETED The window has been deleted.

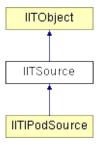
E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITSource Interface Reference

Inheritance diagram for IITSource:



List of all members.

Detailed Description

Represents an entry in the Source list (music library, CD, device, etc.).

You can retrieve all the sources using liTunes::Sources().

Public Member Functions

Properties

HRESULT	Kind ([out, retval] ITSourceKind *kind)
	Returns the kind of the source.
HRESULT	Capacity ([out, retval] double *capacity)
	Returns the total size of the source, if it has a fixed size.
HRESULT	FreeSpace ([out, retval] double *freespace)
	Returns the free space on the source, if it has a fixed size.
HRESULT	Playlists ([out, retval] IITPlaylistCollection **iPlaylistCollection)
	Returns a collection containing the playlists in this source.

Member Function Documentation

HRESULT IITSource::Capacity ([out, retval] double * capacity)

Returns the total size of the source, if it has a fixed size.

Parameters:

capacity Returns the total size of the source, if it has a fixed size.

Will be set to 0 if the source does not have a fixed size.

Return values:

S_*OK* The operation was successful.

E_POINTER capacity is NULL.

ITUNES_E_OBJECTDELETED This source is no longer

available.

E_FAIL An unexpected error occurred.

HRESULT IITSource::FreeSpace ([out, retval] double * freespace)

Returns the free space on the source, if it has a fixed size.

Parameters:

freespace Returns the free space on the source, if it has a fixed size. Will be set to 0 if the source does not have a fixed size.

Return values:

S_OK The operation was successful.

E_POINTER freespace is NULL.

ITUNES_E_OBJECTDELETED This source is no longer

available.

HRESULT IITSource::Kind ([out, retval] ITSourceKind * kind)

Returns the kind of the source.

Parameters:

kind Returns the kind of the source.

Return values:

S_OK The operation was successful.

E_POINTER kind is NULL.

ITUNES_E_OBJECTDELETED This source is no longer

available.

E_FAIL An unexpected error occurred.

HRESULT IITSource::Playlists ([out, retval] IITPlaylistCollection **

Returns a collection containing the playlists in this source.

The source's primary playlist is always the first playlist in the collection.

Parameters:

iPlaylistCollection Collection of IITPlaylist objects.

Return values:

S_OK The operation was successful.

E_POINTER iPlaylistCollection is NULL.

ITUNES_E_OBJECTDELETED This source is no longer

available.

E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITSourceCollection Interface Reference

List of all members.

Detailed Description

Represents a collection of source objects.

Note that collection indices are always 1-based.

You can retrieve all the sources using liTunes::Sources().

Public Member Functions

Properties

HRESULT	Count ([out, retval] long *count) Returns the number of sources in the collection.
HRESULT	Item ([in] long index,[out, retval] IITSource **iSource) Returns an IITSource object corresponding to the given index (1-based).
HRESULT	ItemByName ([in] BSTR name,[out, retval] IITSource **iSource) Returns an IITSource object with the specified name.
HRESULT	_NewEnum ([out, retval] IUnknown **iEnumerator) Returns an IEnumVARIANT object which can enumerate the collection.
HRESULT	ItemByPersistentID ([in] long highID,[in] long lowID,[out, retval] IITSource **iSource)
	Returns an IITSource object with the specified persistent ID. See the documentation on IITObject for more information on persistent IDs.

Member Function Documentation

HRESULT IITSourceCollection::_NewEnum ([out, retval] IUnknowr

Returns an IEnumVARIANT object which can enumerate the collection.

Note:

This property is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Parameters:

iEnumerator Returns an IEnumVARIANT object which can enumerate the collection.

Return values:

S_OK The operation was successful.

E_POINTER iEnumerator is NULL.

E_FAIL An unexpected error occurred.

HRESULT IITSourceCollection::Count ([out, retval] long * count)

Returns the number of sources in the collection.

Parameters:

count Returns the number of sources in the collection.

Return values:

S_OK The operation was successful.

E_POINTER count is NULL.

ITUNES_E_OBJECTDELETED The source has been deleted.

E_FAIL An unexpected error occurred.

```
HRESULT IITSourceCollection::Item ( [in] long inde [out, retval] IITSource ** iSou
```

Returns an **IITSource** object corresponding to the given index (1-based).

Parameters:

index Index of the source to retrieve, must be less than or

equal to IITSourceCollection::Count().

iSource Returns an **IITSource** object corresponding to the given

index. Will be set to NULL if no source could be

retrieved.

Return values:

S_OK The operation was successful.

S_FALSE index is too high.

E_INVALIDARG index is less than one.

E_POINTER iSource is NULL.

E_FAIL An unexpected error occurred.

```
HRESULT IITSourceCollection::ItemByName ( [in] BSTR [out, retval] IITSource )
```

Returns an **IITSource** object with the specified name.

Parameters:

name The name of the source to retrieve.

iSource Returns an **IITSource** object with the specified name.

Will be set to NULL if no source could be retrieved.

Return values:

S_OK The operation was successful.

S_FALSE A source with the specified name could not be

found.

E_INVALIDARG name is empty.

E_POINTER name or iSource is NULL.

E FAIL An unexpected error occurred.

Returns an **IITSource** object with the specified persistent ID. See the documentation on **IITObject** for more information on persistent IDs.

Note:

This property is available in iTunes 7.7 and later (iTunes type library 1.11 and later).

Parameters:

highID The high 32 bits of the 64-bit persistent ID.

lowID The low 32 bits of the 64-bit persistent ID.

iSource Returns an **IITSource** object with the specified persistent

ID. Will be set to NULL if no source could be retrieved.

Return values:

S_OK The operation was successful.

S_FALSE A source with the specified persistent ID could not

be found.

E_POINTER iSource is NULL.

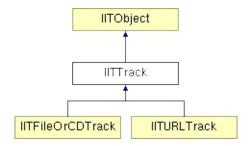
E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITTrack Interface Reference

Inheritance diagram for IITTrack:



List of all members.

Detailed Description

Represents a track.

A track represents a song in a single playlist. A song may be in more than one playlist, in which case it would be represented by multiple tracks.

You can retrieve the currently targeted (playing) track using **liTunes::CurrentTrack()**.

Typically, an IITrack is accessed through an IITTrackCollection.

You can retrieve all the tracks defined for a playlist using IITPlaylist::Tracks().

You can retrieve the currently selected track or tracks using **liTunes::SelectedTracks()**.

After calling IITLibraryPlaylist::AddFile(),

IITLibraryPlaylist::AddFiles(), IITUserPlaylist::AddFile(),

IITUserPlaylist::AddFiles(), IiTunes::ConvertFile(),

liTunes::ConvertFile2(), liTunes::ConvertFiles(),

liTunes::ConvertFiles2(), liTunes::ConvertTrack(),

IiTunes::ConvertTrack2(), IiTunes::ConvertTracks(), or

liTunes::ConvertTracks2(), you can retrieve the added or converted

tracks using IITOperationStatus::Tracks().

Public Member Functions

Methods

HRESULT	Delete ()
	Delete this track.
HRESULT	Play ()
	Start playing this track.
HRESULT	AddArtworkFromFile ([in] BSTR filePath,[out, retval] IITArtwork **iArtwork)
	Add artwork from an image file to this track.

Properties

HRESULT	Kind ([out, retval] ITTrackKind *kind) Returns the kind of the track.
HRESULT	Playlist ([out, retval] IITPlaylist **iPlaylist)
	Returns an IITPlaylist object corresponding to the playlist that contains the track. Use IITFileOrCDTrack::Playlists() or IITURLTrack::Playlists() to get the collection of all playlists that contain the song this track represents.
HRESULT	Album ([out, retval] BSTR *album) Returns the name of the album containing the track.
HRESULT	Album ([in] BSTR album)
	Set the name of the album containing the track.
HRESULT	Artist ([out, retval] BSTR *artist)
	Returns the name of the artist/source of the track.
HRESULT	Artist ([in] BSTR artist)
	Set the name of the artist/source of the track.
HRESULT	BitRate ([out, retval] long *bitrate)
	Returns the bit rate of the track (in kbps).
HRESULT	BPM ([out, retval] long *beatsPerMinute)
	Returns the tempo of the track (in beats per minute).
HRESULT	BPM ([in] long beatsPerMinute)
	Set the tempo of the track (in beats per minute).
HRESULT	(L / 1
	Returns freeform notes about the track.
HRESULT	Comment ([in] BSTR comment)
	Set freeform notes about the track.
HRESULT	Compilation ([out, retval] VARIANT_BOOL *isCompilation)

	Returns true if this track is from a compilation album.
HRESULT	Compilation ([in] VARIANT_BOOL shouldBeCompilation)
	Set whether this track is from a compilation album.
HRESULT	Composer ([out, retval] BSTR *composer)
LIDECLUT	Returns the composer of the track.
HRESULT	Composer ([in] BSTR composer) Set the composer of the track.
HRESULT	DateAdded ([out, retval] DATE *dateAdded)
	Returns the date the track was added to the playlist.
HRESULT	DiscCount ([out, retval] long *discCount) Returns the total number of discs in the source album.
HRESULT	DiscCount ([in] long discCount)
	Set the total number of discs in the source album.
HRESULT	DiscNumber ([out, retval] long *discNumber)
	Returns the index of the disc containing the track on the source album.
HRESULT	DiscNumber ([in] long discNumber) Set the index of the disc containing the track on the source album.
HRESULT	Duration ([out, retval] long *duration)
	Returns the length of the track (in seconds).
HRESULT	Enabled ([out, retval] VARIANT_BOOL *isEnabled) Returns true if the track is checked for playback.
HRESULT	Enabled ([in] VARIANT_BOOL shouldBeEnabled) Set whether this track is checked for playback.
HRESULT	EQ ([out, retval] BSTR *eq)
	Returns the name of the EQ preset of the track.
HRESULT	EQ ([in] BSTR eq)
	Set the name of the EQ preset of the track.
HRESULT	Finish ([in] long finish)
	Set the stop time of the track (in seconds).
HRESULT	Finish ([out, retval] long *finish)
	Returns the stop time of the track (in seconds).
HRESULT	Genre ([out, retval] BSTR *genre) Returns the music/audio genre (category) of the track.
HRESULT	Genre ([in] BSTR genre) Set the music/audio genre (category) of the track.
HRESULT	Grouping ([out, retval] BSTR *grouping)
LOOL1	Returns the grouping (piece) of the track.
HRESULT	Grouping ([in] BSTR grouping)
	Set the grouping (piece) of the track.

HRESULT	KindAsString ([out, retval] BSTR *kind) Returns the text description of the track (e.g. "AAC audio file").
HRESULT	ModificationDate ([out, retval] DATE *dateModified) Returns the modification date of the content of the track.
HRESULT	PlayedCount ([out, retval] long *playedCount) Returns the number of times the track has been played.
HRESULT	PlayedCount ([in] long playedCount) Set the number of times the track has been played. This property cannot be set if the track is not playable (e.g. a PDF file).
HRESULT	PlayedDate ([out, retval] DATE *playedDate) Returns the date and time the track was last played.
HRESULT	PlayedDate ([in] DATE playedDate) Set the date and time the track was last played. This property cannot be set if the track is not playable (e.g. a PDF file).
HRESULT	PlayOrderIndex ([out, retval] long *index) Returns the play order index of the track in the owner playlist (1-based).
HRESULT	Rating ([out, retval] long *rating) Returns the rating of the track (0 to 100). If the track rating has never been set, or has been set to 0, it will be computed based on the album rating.
HRESULT	Rating ([in] long rating) Set the rating of the track (0 to 100). If the track rating is set to 0, it will be computed based on the album rating.
HRESULT	SampleRate ([out, retval] long *sampleRate) Returns the sample rate of the track (in Hz).
HRESULT	Size ([out, retval] long *size) Returns the size of the track (in bytes).
HRESULT	Start ([out, retval] long *start) Returns the start time of the track (in seconds).
HRESULT	Start ([in] long start) Set the start time of the track (in seconds).
HRESULT	Time ([out, retval] BSTR *time) Returns the length of the track (in MM:SS format).
HRESULT	TrackCount ([out, retval] long *trackCount) Returns the total number of tracks on the source album.
HRESULT	TrackCount ([in] long trackCount) Set the total number of tracks on the source album.
HRESULT	TrackNumber ([out, retval] long *trackNumber) Returns the index of the track on the source album.
HRESULT	TrackNumber ([in] long trackNumber) Set the index of the track on the source album.

HRESULT	VolumeAdjustment ([out, retval] long *volumeAdjustment)
	Returns the relative volume adjustment of the track (-100% to 100%).
HRESULT	VolumeAdjustment ([in] long volumeAdjustment)
	Set the relative volume adjustment of the track (-100% to 100%).
HRESULT	Year ([out, retval] long *year)
	Returns the year the track was recorded/released.
HRESULT	Year ([in] long year)
	Set the year the track was recorded/released.
HRESULT	Artwork ([out, retval] IITArtworkCollection **iArtworkCollection)
	Returns a collection containing the artwork for the track.

_

Member Function Documentation

```
HRESULT IITTrack::AddArtworkFromFile ( [in] BSTR [out, retval] IITArtwork **
)
```

Add artwork from an image file to this track.

Parameters:

filePath The full path to the image file containing artwork to add. *iArtwork* Returns an **IITArtwork** object representing the new artwork.

Return values:

 S_OK The operation was successful. $E_POINTER$ filePath or iArtwork is NULL. $E_INVALIDARG$ filePath is empty or too long. $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITTrack::Album ([in] BSTR album)

Set the name of the album containing the track.

Parameters:

album The new name of the album containing the track.

Return values:

S_OK The operation was successful.

E_POINTER album is NULL.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Album ([out, retval] BSTR * album)

Returns the name of the album containing the track.

Parameters:

album Returns the name of the album containing the track.

Return values:

S_OK The operation was successful.

E_POINTER album is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Artist ([in] BSTR artist)

Set the name of the artist/source of the track.

Parameters:

artist The new artist/source of the track.

Return values:

S_OK The operation was successful.

E_POINTER artist is NULL.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Artist ([out, retval] BSTR * artist)

Returns the name of the artist/source of the track.

Parameters:

artist Returns the name of the artist/source of the track.

Return values:

S_OK The operation was successful.

E POINTER artist is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Artwork ([out, retval] IITArtworkCollection ** i/

Returns a collection containing the artwork for the track.

Parameters:

iArtworkCollection Collection of IITArtwork objects.

Return values:

S_OK The operation was successful.

E_POINTER iArtworkCollection is NULL.

ITUNES_E_OBJECTDELETED This source is no longer

available.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::BitRate ([out, retval] long * bitrate)

Returns the bit rate of the track (in kbps).

Parameters:

bitrate Returns the bit rate of the track (in kbps).

Return values:

S_OK The operation was successful.

E_POINTER bitrate is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::BPM ([in] long beatsPerMinute)

Set the tempo of the track (in beats per minute).

Parameters:

beatsPerMinute The new tempo of the track (in beats per minute).

Return values:

S_OK The operation was successful.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::BPM ([out, retval] long * beatsPerMinute)

Returns the tempo of the track (in beats per minute).

Parameters:

beatsPerMinute Returns the tempo of the track (in beats per minute).

Return values:

 S_OK The operation was successful. $E_POINTER$ beatsPerMinute is NULL. $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITTrack::Comment ([in] BSTR comment)

Set freeform notes about the track.

Parameters:

comment The new freeform notes about the track.

Return values:

S_OK The operation was successful. E POINTER comment is NULL.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Comment ([out, retval] BSTR * comment)

Returns freeform notes about the track.

Parameters:

comment. Returns freeform notes about the track.

Return values:

S_OK The operation was successful.

E POINTER comment is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Compilation ([in] VARIANT_BOOL shouldBeC

Set whether this track is from a compilation album.

Parameters:

shouldBeCompilation True if this track should be from a compilation album.

Return values:

S_OK The operation was successful.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Compilation ([out, retval] VARIANT_BOOL * is

Returns true if this track is from a compilation album.

Parameters:

isCompilation Returns true if this track is from a compilation album.

Return values:

S_OK The operation was successful.

E_POINTER isCompilation is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Composer ([in] BSTR composer)

Set the composer of the track.

Parameters:

composer The new composer of the track.

Return values:

S_*OK* The operation was successful.

E_POINTER composer is NULL.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Composer ([out, retval] BSTR * composer)

Returns the composer of the track.

Parameters:

composer Returns the composer of the track.

Return values:

S_OK The operation was successful.

E_POINTER composer is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::DateAdded ([out, retval] DATE * dateAdded)

Returns the date the track was added to the playlist.

Parameters:

dateAdded Returns the date the track was added to the playlist.

Return values:

S_OK The operation was successful.

E_POINTER dateAdded is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Delete ()

Delete this track.

Note that if a track is deleted from a source's primary playlist, it will be deleted from all playlist's in that source.

Return values:

S_OK The operation was successful.

ITUNES_E_OBJECTLOCKED The track cannot be deleted.

ITUNES_E_OBJECTDELETED This track has been deleted already.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::DiscCount ([in] long discCount)

Set the total number of discs in the source album.

Parameters:

discCount The new total number of discs in the source album.

Return values:

S_OK The operation was successful.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

HRESULT IITTrack::DiscCount ([out, retval] long * discCount)

Returns the total number of discs in the source album.

Parameters:

discCount Returns the total number of discs in the source album.

Return values:

S_OK The operation was successful.

E_POINTER discCount is NULL.

 $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITTrack::DiscNumber ([in]long discNumber)

Set the index of the disc containing the track on the source album.

Parameters:

discNumber The new index of the disc containing the track on the source album.

Return values:

 S_OK The operation was successful. $ITUNES_E_OBJECTLOCKED$ This track is not modifiable. $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITTrack::DiscNumber ([out, retval] long * discNumber)

Returns the index of the disc containing the track on the source album.

Parameters:

discNumber Returns the index of the disc containing the track on the source album.

Return values:

S_OK The operation was successful.

E_POINTER discNumber is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Duration ([out, retval] long * duration)

Returns the length of the track (in seconds).

Parameters:

duration Returns the length of the track (in seconds).

Return values:

S_OK The operation was successful.

E_POINTER duration is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Enabled ([in] VARIANT_BOOL shouldBeEnabl

Set whether this track is checked for playback.

Parameters:

shouldBeEnabled True if the track should be checked for playback.

Return values:

 S_OK The operation was successful. ITUNES_E_OBJECTLOCKED This track is not modifiable. ITUNES_E_OBJECTDELETED This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITTrack::Enabled ([out, retval] VARIANT_BOOL * isEna

Returns true if the track is checked for playback.

Parameters:

isEnabled Returns true if the track is checked for playback.

Return values:

S_OK The operation was successful.

E_POINTER isEnabled is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::EQ ([in] BSTR eq)

Set the name of the EQ preset of the track.

Parameters:

eq The new name of the EQ preset of the track.

Return values:

S_OK The operation was successful.

E_POINTER eq is NULL.

ITUNES_E_OBJECTLOCKED This track is not modifiable. ITUNES_E_OBJECTDELETED This track has been deleted.

HRESULT IITTrack::EQ ([out, retval] BSTR * eq)

Returns the name of the EQ preset of the track.

Parameters:

eq Returns the name of the EQ preset of the track. Will be set to NULL if no EQ preset is assigned to the track.

Return values:

S_OK The operation was successful.

E_POINTER eq is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Finish ([out, retval] long * finish)

Returns the stop time of the track (in seconds).

Parameters:

finish Returns the stop time of the track (in seconds).

Return values:

S_OK The operation was successful.

E_POINTER finish is NULL.

 $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITTrack::Finish ([in] long finish)

Set the stop time of the track (in seconds).

Parameters:

finish The new stop time of the track (in seconds).

Return values:

S_OK The operation was successful.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Genre ([in] BSTR genre)

Set the music/audio genre (category) of the track.

Parameters:

genre The new music/audio genre (category) of the track.

Return values:

 S_OK The operation was successful. $E_POINTER$ genre is NULL. $ITUNES_E_OBJECTLOCKED$ This track is not modifiable. $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITTrack::Genre ([out, retval] BSTR * genre)

Returns the music/audio genre (category) of the track.

Parameters:

genre Returns the music/audio genre (category) of the track.

Return values:

S_OK The operation was successful.

E_POINTER genre is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Grouping ([in] BSTR grouping)

Set the grouping (piece) of the track.

Generally used to denote movements within classical work.

Parameters:

grouping The new grouping (piece) of the track.

Return values:

S_OK The operation was successful.

E_POINTER grouping is NULL.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E FAIL An unexpected error occurred.

HRESULT IITTrack::Grouping ([out, retval] BSTR * grouping)

Returns the grouping (piece) of the track.

Generally used to denote movements within classical work.

Parameters:

grouping Returns the grouping (piece) of the track.

Return values:

S_OK The operation was successful.

E_POINTER grouping is NULL.

 $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITTrack::Kind ([out, retval] ITTrackKind * kind)

Returns the kind of the track.

Parameters:

kind Returns the kind of the track.

Return values:

S_OK The operation was successful.

E_POINTER kind is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::KindAsString ([out, retval] BSTR * kind)

Returns the text description of the track (e.g. "AAC audio file").

Parameters:

kind Returns the text description of the track.

Return values:

S_OK The operation was successful.

E_POINTER kind is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::ModificationDate ([out, retval] DATE * dateMod

Returns the modification date of the content of the track.

Parameters:

dateModified Returns the modification date of the content of the track.

Return values:

S_OK The operation was successful.

E_POINTER dateModified is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Play ()

Start playing this track.

Return values:

 S_OK The operation was successful. ITUNES_E_OBJECTDELETED This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITTrack::PlayedCount ([in] long playedCount)

Set the number of times the track has been played. This property cannot be set if the track is not playable (e.g. a PDF file).

Parameters:

playedCount The new number of times the track has been played.

Return values:

 S_OK The operation was successful. ITUNES_E_OBJECTLOCKED This track is not modifiable. ITUNES_E_OBJECTDELETED This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITTrack::PlayedCount ([out, retval] long * playedCount

Returns the number of times the track has been played.

Parameters:

playedCount Returns the number of times the track has been played.

Return values:

 S_OK The operation was successful. $E_POINTER$ playedCount is NULL. $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITTrack::PlayedDate ([in] DATE playedDate)

Set the date and time the track was last played. This property cannot be set if the track is not playable (e.g. a PDF file).

A value of zero means no played date.

Parameters:

playedDate The new date and time the track was last played.

Return values:

S_OK

The operation was successful.

E_INVALIDARG playedDate is invalid.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E FAIL An unexpected error occurred.

HRESULT IITTrack::PlayedDate ([out, retval] DATE * playedDate)

Returns the date and time the track was last played.

A value of zero means no played date.

Parameters:

playedDate Returns the date and time the track was last played.

Return values:

 S_OK The operation was successful. $E_POINTER$ playedDate is NULL. $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITTrack::Playlist ([out, retval] IITPlaylist ** iPlaylist)

Returns an **IITPlaylist** object corresponding to the playlist that contains the track. Use **IITFileOrCDTrack::Playlists()** or **IITURLTrack::Playlists()** to get the collection of all playlists that contain the song this track represents.

Parameters:

iPlaylist Returns an **IITPlaylist** object corresponding to the playlist that contains the track.

Return values:

S_OK The operation was successful.

E_POINTER iPlaylist is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E FAIL An unexpected error occurred.

HRESULT IITTrack::PlayOrderIndex ([out, retval] long * index)

Returns the play order index of the track in the owner playlist (1-based).

You can pass this index to IITTrackCollection::ItemByPlayOrder() for the collection returned by IITPlaylist::Tracks() to retrieve an IITTrack object corresponding to this track.

Parameters:

index Returns the play order index of the track in the owner playlist (1-based).

Return values:

S_*OK* The operation was successful.

E POINTER index is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Rating ([in] long rating)

Set the rating of the track (0 to 100). If the track rating is set to 0, it will be computed based on the album rating.

Parameters:

rating The new rating of the track (0 to 100). If rating is outside this range, it will be pinned.

Return values:

 S_OK The operation was successful. ITUNES_E_OBJECTLOCKED This track is not modifiable. ITUNES_E_OBJECTDELETED This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITTrack::Rating ([out, retval] long * rating)

Returns the rating of the track (0 to 100). If the track rating has never been set, or has been set to 0, it will be computed based on the album rating.

Parameters:

rating Returns the rating of the track (0 to 100).

Return values:

S_OK The operation was successful.

E_POINTER rating is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::SampleRate ([out, retval] long * sampleRate)

Returns the sample rate of the track (in Hz).

Parameters:

sampleRate Returns the sample rate of the track (in Hz).

Return values:

S_OK The operation was successful. E_POINTER sampleRate is NULL. ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Size ([out, retval] long * size)

Returns the size of the track (in bytes).

Parameters:

size Returns the size of the track (in bytes).

Return values:

S_OK The operation was successful.

E POINTER size is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Start ([in] long start)

Set the start time of the track (in seconds).

Parameters:

start The new start time of the track (in seconds).

Return values:

S_OK The operation was successful.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::Start ([out, retval] long * start)

Returns the start time of the track (in seconds).

Parameters:

start Returns the start time of the track (in seconds).

Return values:

S_*OK* The operation was successful.

E_POINTER start is NULL.

 $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITTrack::Time ([out, retval] BSTR * time)

Returns the length of the track (in MM:SS format).

Parameters:

time Returns the length of the track (in MM:SS format).

Return values:

S_OK The operation was successful.

E_POINTER time is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::TrackCount ([in] long trackCount)

Set the total number of tracks on the source album.

Parameters:

trackCount The new total number of tracks on the source album.

Return values:

S_OK The operation was successful.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::TrackCount ([out, retval] long * trackCount)

Returns the total number of tracks on the source album.

Parameters:

trackCount Returns the total number of tracks on the source album.

Return values:

S_OK The operation was successful.

E_POINTER trackCount is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::TrackNumber ([in] long trackNumber)

Set the index of the track on the source album.

Parameters:

trackNumber The new index of the track on the source album.

Return values:

S_OK The operation was successful.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES E OBJECTDELETED This track has been deleted.

HRESULT IITTrack::TrackNumber ([out, retval] long * trackNumber

Returns the index of the track on the source album.

Parameters:

trackNumber Returns the index of the track on the source album.

Return values:

S_OK The operation was successful.

E_POINTER trackNumber is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrack::VolumeAdjustment ([in] long volumeAdjustme

Set the relative volume adjustment of the track (-100% to 100%).

Parameters:

volumeAdjustment The new relative volume adjustment of the track (-100% to 100%). If volumeAdjustment is outside this range, it will be pinned.

Return values:

S_OK The operation was successful.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E FAIL An unexpected error occurred.

HRESULT IITTrack::VolumeAdjustment ([out, retval] long * volume

Returns the relative volume adjustment of the track (-100% to 100%).

Parameters:

volumeAdjustment Returns the relative volume adjustment of the track (-100% to 100%).

Return values:

 S_OK The operation was successful. $E_POINTER$ volumeAdjustment is NULL. $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITTrack::Year ([in] long year)

Set the year the track was recorded/released.

Parameters:

year The new year the track was recorded/released.

Return values:

 S_OK The operation was successful. ITUNES_E_OBJECTLOCKED This track is not modifiable. ITUNES_E_OBJECTDELETED This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITTrack::Year ([out, retval] long * year)

Returns the year the track was recorded/released.

Parameters:

year Returns the year the track was recorded/released.

Return values:

S_*OK* The operation was successful.

E_POINTER year is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITTrackCollection Interface Reference

List of all members.

Detailed Description

Represents a collection of track objects.

Note that collection indices are always 1-based.

You can retrieve all the tracks defined for a playlist using IITPlaylist::Tracks().

You can retrieve the currently selected track or tracks using **liTunes::SelectedTracks()**.

After calling IITLibraryPlaylist::AddFile(),

IITLibraryPlaylist::AddFiles(), IITUserPlaylist::AddFile(),

IITUserPlaylist::AddFiles(), IiTunes::ConvertFile(),

liTunes::ConvertFile2(), liTunes::ConvertFiles(),

liTunes::ConvertFiles2(), liTunes::ConvertTrack(),

IiTunes::ConvertTrack2(), IiTunes::ConvertTracks(), or

liTunes::ConvertTracks2(), you can retrieve the added or converted

tracks using IITOperationStatus::Tracks().

Public Member Functions

Properties

HRESULT	Count ([out, retval] long *count) Returns the number of tracks in the collection.
HRESULT	Item ([in] long index,[out, retval] IITTrack **iTrack) Returns an IITTrack object corresponding to the given index (1-based).
HRESULT	ItemByPlayOrder ([in] long index,[out, retval] IITTrack **iTrack) Returns an IITTrack object corresponding to the given index (1-based).
HRESULT	ItemByName ([in] BSTR name,[out, retval] IITTrack **iTrack) Returns an IITTrack object with the specified name.
HRESULT	_NewEnum ([out, retval] IUnknown **iEnumerator) Returns an IEnumVARIANT object which can enumerate the collection.
HRESULT	ItemByPersistentID ([in] long highID,[in] long lowID,[out, retval] IITTrack **iTrack) Returns an IITTrack object with the specified persistent ID. See the documentation on IITObject for more information on persistent IDs.

Member Function Documentation

HRESULT IITTrackCollection::_NewEnum ([out, retval] IUnknown *

Returns an IEnumVARIANT object which can enumerate the collection.

Note that track enumerators use IITTrackCollection::Item(), not IITTrackCollection::ItemByPlayOrder(). In other words, you cannot use an enumerator to enumerate tracks by play order.

Note:

This property is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Parameters:

iEnumerator Returns an IEnumVARIANT object which can enumerate the collection.

Return values:

S_OK The operation was successful.

E_POINTER iEnumerator is NULL.

E_FAIL An unexpected error occurred.

HRESULT IITTrackCollection::Count ([out, retval] long * count)

Returns the number of tracks in the collection.

Parameters:

count Returns the number of tracks in the collection.

Return values:

S OK

The operation was successful.

E_POINTER count is NULL.

ITUNES_E_OBJECTDELETED The playlist containing the tracks has been deleted.

E_FAIL An unexpected error occurred.

Returns an IITTrack object corresponding to the given index (1-based).

The index is independent of the play order in the playlist.

Parameters:

index Index of the track to retrieve, must be less than or equal to IITTrackCollection::Count().

iTrack Returns an **IITTrack** object corresponding to the given index. Will be set to NULL if no track could be retrieved.

Return values:

S_OK The operation was successful.

S_FALSE index is too high.

E_INVALIDARG index is less than one.

E_POINTER iTrack is NULL.

ITUNES_E_OBJECTDELETED The playlist containing the tracks

has been deleted.

E_FAIL An unexpected error occurred.

```
HRESULT IITTrackCollection::ItemByName ( [in] BSTR [out, retval] IITTrack **
)
```

Returns an **IITTrack** object with the specified name.

Parameters:

name The name of the track to retrieve.

iTrack Returns an **IITTrack** object with the specified name. Will be set to NULL if no track could be retrieved.

Return values:

S_OK The operation was successful. S_FALSE A track with the specified name

could not be found.

 $E_INVALIDARG$ name is empty.

E_POINTER name or iTrack is NULL.

ITUNES_E_OBJECTDELETED The playlist containing the tracks

has been deleted.

E_FAIL An unexpected error occurred.

Returns an **IITTrack** object with the specified persistent ID. See the documentation on **IITObject** for more information on persistent IDs.

Note:

This property is available in iTunes 7.7 and later (iTunes type library 1.11 and later).

Parameters:

highID The high 32 bits of the 64-bit persistent ID.

lowID The low 32 bits of the 64-bit persistent ID.

iTrack Returns an IITTrack object with the specified persistent

ID. Will be set to NULL if no track could be retrieved.

Return values:

S_OK The operation was successful.

S FALSE A track with the specified

persistent ID could not be found.

E_POINTER iTrack is NULL.

ITUNES_E_OBJECTDELETED The playlist containing the tracks

has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITTrackCollection::ItemByPlayOrder ([in] long [out, retval] IITTra

Returns an IITTrack object corresponding to the given index (1-based).

The index is defined by the play order of the playlist containing the track collection.

Parameters:

index Index of the track to retrieve, must be less than or equal to IITTrackCollection::Count().

iTrack Returns an IITTrack object corresponding to the given play order index. Will be set to NULL if no track could be retrieved.

Return values:

S_*OK* The operation was successful.

S_FALSE index is too high.

E_INVALIDARG index is less than one.

E POINTER iTrack is NULL.

ITUNES_E_OBJECTDELETED The playlist containing the tracks

 E_FAIL

has been deleted. An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

liTunes Interface Reference

List of all members.

Detailed Description

Defines the top-level iTunes application object.

This interface defines the top-level iTunes application object. All other iTunes interfaces are accessed through this object.

This is the only externally creatable iTunes COM interface. All other iTunes interfaces are accessed through this interface. It can be created using the **iTunesApp** class.

To create an liTunes object from C++:

```
#include "iTunesCOMInterface.h"

HRESULT hRes;
IiTunes* iITunes;

// note - CLSID_iTunesApp and IID_IiTunes are defined in iTunes
hRes = ::CoCreateInstance(CLSID_iTunesApp, NULL, CLSCTX_LOCAL_S
```

To create an liTunes object from C#:

```
iTunesLib.IiTunes iTunesApp;
iTunesApp = new iTunesLib.iTunesAppClass();
```

To create an liTunes object from Visual Basic:

```
Imports iTunesLib
Dim iTunesApp as new iTunesApp
```

To create an liTunes object from VBScript:

```
Dim iTunesApp;
Set iTunesApp = WScript.CreateObject("iTunes.Application");
```

To create an liTunes object from JScript:

```
var iTunesApp = WScript.CreateObject("iTunes.Application");
```

To create an liTunes object from Perl:

```
$iTunesApp = new Win32::OLE("iTunes.Application");
```

Public Member Functions

Player Controls

HRESULT	BackTrack ()
	Reposition to the beginning of the current track or go to the previous track if already at start of current track.
HRESULT	FastForward ()
	Skip forward in a playing track.
HRESULT	NextTrack () Advance to the next track in the current playlist.
HRESULT	Pause ()
	Pause playback.
HRESULT	Play ()
	Play the currently targeted track.
HRESULT	PlayFile ([in] BSTR filePath)
	Play the specified file path, adding it to the library if not already present.
HRESULT	
	Toggle the playing/paused state of the current track.
HRESULT	PreviousTrack ()
	Return to the previous track in the current playlist.
HRESULT	Resume ()
	Disable fast forward/rewind and resume playback, if playing.
HRESULT	Rewind () Skip backwards in a playing track
LIDEOLUT	Skip backwards in a playing track.
HRESULT	Stop () Stop playback.
LIDECLILE	
HRESULT	GetPlayerButtonsState ([out] VARIANT_BOOL *previousEnabled,[out] ITPlayButtonState *playPauseStopState,[out] VARIANT_BOOL *nextEnabled)
	Retrieves the current state of the player buttons in the window containing the
	currently targeted track. If there is no currently targeted track, returns the current state of the player buttons in the main browser window.
HRESULT	PlayerButtonClicked ([in] ITPlayerButton playerButton,[in] long
	playerButtonModifierKeys)
	Simulate a click on a player control button in the window containing the currently targeted track. If there is no currently targeted track, simulates a click on a player control button in the main browser window.
	Constant and main provider window

Conversion Methods

HRESULT	ConvertFile ([in] BSTR filePath,[out, retval] IITOperationStatus **iStatus) Start converting the specified file or folder.
HRESULT	ConvertFiles ([in] VARIANT *filePaths,[out, retval] IITOperationStatus **iStatus)
	Start converting the specified files and/or folders.
HRESULT	ConvertTrack ([in] VARIANT *iTrackToConvert,[out, retval] IITOperationStatus **iStatus)
	Start converting the specified track.
HRESULT	ConvertTracks ([in] VARIANT *iTracksToConvert,[out, retval] IITOperationStatus **iStatus)
	Start converting the specified tracks.
HRESULT	ConvertFile2 ([in] BSTR filePath,[out, retval] IITConvertOperationStatus **iStatus)
	Start converting the specified file or folder.
HRESULT	ConvertFiles2 ([in] VARIANT *filePaths,[out, retval] IITConvertOperationStatus **iStatus)
	Start converting the specified files and/or folders.
HRESULT	ConvertTrack2 ([in] VARIANT *iTrackToConvert,[out, retval] IITConvertOperationStatus **iStatus)
	Start converting the specified track.
HRESULT	ConvertTracks2 ([in] VARIANT *iTracksToConvert,[out, retval] IITConvertOperationStatus **iStatus)
	Start converting the specified tracks.

Miscellaneous Methods

HRESULT	CheckVersion ([in] long majorVersion,[in] long minorVersion,[out, retval] VARIANT_BOOL *isCompatible)
	Returns true if this version of the iTunes type library is compatible with the specified version.
HRESULT	GetITObjectByID ([in] long sourceID,[in] long playlistID,[in] long trackID,[in] long databaseID,[out, retval] IITObject **iObject)
	Returns an IITObject corresponding to the specified IDs.
HRESULT	CreatePlaylist ([in] BSTR playlistName,[out, retval] IITPlaylist **iPlaylist)
	Creates a new playlist in the main library.
HRESULT	OpenURL ([in] BSTR url)
	Open the specified iTunes Store or streaming audio URL.
HRESULT	GotoMusicStoreHomePage ()
	Go to the iTunes Store home page.
HRESULT	UpdatelPod ()
	Update the contents of the iPod.

HRESULT	Quit ()
	Exits the iTunes application.
HRESULT	CreateEQPreset ([in] BSTR eqPresetName,[out, retval] IITEQPreset **iEQPreset)
	Creates a new EQ preset.
HRESULT	CreatePlaylistInSource ([in] BSTR playlistName,[in] VARIANT *iSource,[out, retval] IITPlaylist **iPlaylist)
	Creates a new playlist in an existing source.
HRESULT	SubscribeToPodcast ([in] BSTR url)
	Subscribes to the specified podcast feed URL. Any "unsafe" characters in the URL should already be converted into their corresponding escape sequences, iTunes will not do this.
HRESULT	UpdatePodcastFeeds ()
	Updates all podcast feeds. This is equivalent to the user pressing the Update button when Podcasts is selected in the Source list.
HRESULT	CreateFolder ([in] BSTR folderName,[out, retval] IITPlaylist **iFolder)
	Creates a new folder in the main library.
HRESULT	CreateFolderInSource ([in] BSTR folderName,[in] VARIANT *iSource,[out, retval] IITPlaylist **iFolder)
	Creates a new folder in an existing source.
HRESULT	GetITObjectPersistentIDs ([in] VARIANT *iObject,[out] long *highID,[out] long *lowID)
	Returns the high and low 32 bits of the persistent ID of the specified IITObject . See the documentation on IITObject for more information on persistent IDs.

Collection Properties

HRESULT	Sources ([out, retval] IITSourceCollection **iSourceCollection) Returns a collection of music sources (music library, CD, device, etc.).
HRESULT	Encoders ([out, retval] IITEncoderCollection **iEncoderCollection) Returns a collection of encoders.
HRESULT	EQPresets ([out, retval] IITEQPresetCollection **iEQPresetCollection) Returns a collection of EQ presets.
HRESULT	Visuals ([out, retval] IITVisualCollection **iVisualCollection) Returns a collection of visual plug-ins.
HRESULT	Windows ([out, retval] IITWindowCollection **iWindowCollection) Returns a collection of visual plug-ins.

Sound Properties

HRESULT	SoundVolume ([out, retval] long *volume) Returns the sound output volume (0 = minimum, 100 = maximum).
HRESULT	SoundVolume ([in] long volume)

	Sets the sound output volume (0 = minimum, 100 = maximum).
HRESULT	Mute ([out, retval] VARIANT_BOOL *isMuted)
	Returns true if the sound output is muted.
HRESULT	Mute ([in] VARIANT_BOOL shouldMute)
	Sets sound output mute state.

Player Properties

HRESULT	PlayerState ([out, retval] ITPlayerState *playerState) Returns the current player state.
HRESULT	PlayerPosition ([out, retval] long *playerPos) Returns the player's position within the currently playing track in seconds.
HRESULT	PlayerPosition ([in] long playerPos) Sets the player's position within the currently playing track in seconds.

Encoder Properties

HRESULT	CurrentEncoder ([out, retval] IITEncoder **iEncoder)
	Returns the currently selected encoder (AAC, MP3, AIFF, WAV, etc.).
HRESULT	CurrentEncoder ([in] IITEncoder *iEncoder)
	Sets the current encoder (AAC, MP3, AIFF, WAV, etc.).

Visual Properties

HRESULT	VisualsEnabled ([out, retval] VARIANT_BOOL *isEnabled) Returns true if visuals are currently being displayed.
HRESULT	VisualsEnabled ([in] VARIANT_BOOL shouldEnable) Set whether visuals should be displayed.
HRESULT	FullScreenVisuals ([out, retval] VARIANT_BOOL *isFullScreen) Returns true if the visuals are displayed using the entire screen.
HRESULT	FullScreenVisuals ([in] VARIANT_BOOL shouldUseFullScreen) Set whether visuals should be displayed using the entire screen.
HRESULT	VisualSize ([out, retval] ITVisualSize *visualSize) Returns the size of the displayed visual.
HRESULT	VisualSize ([in] ITVisualSize visualSize) Sets the size of the displayed visual.
HRESULT	CurrentVisual ([out, retval] IITVisual **iVisual) Returns the currently selected visual plug-in.
HRESULT	CurrentVisual ([in] IITVisual *iVisual) Sets the current visual plug-in.

EQ Properties

HRESULT	EQEnabled ([out, retval] VARIANT_BOOL *isEnabled)
	Returns true if the equalizer is enabled.
HRESULT	EQEnabled ([in] VARIANT_BOOL shouldEnable) Set whether the equalizer should be enabled.
HRESULT	CurrentEQPreset ([out, retval] IITEQPreset **iEQPreset) Returns the currently selected EQ preset.

Streaming Properties

HRESULT	CurrentStreamTitle ([out, retval] BSTR *streamTitle)
	Returns the name of the current song in the playing stream (provided by streaming server).
HRESULT	CurrentStreamURL ([out, retval] BSTR *streamURL)
	Returns the URL of the playing stream or streaming web site (provided by streaming server).

Miscellaneous Properties

HRESULT	BrowserWindow ([out, retval] IITBrowserWindow **iBrowserWindow)
	Returns the main iTunes browser window.
HRESULT	EQWindow ([out, retval] IITWindow **iEQWindow)
	Returns the EQ window.
HRESULT	LibrarySource ([out, retval] IITSource **iLibrarySource)
	Returns the source that represents the main library.
HRESULT	LibraryPlaylist ([out, retval] IITLibraryPlaylist **iLibraryPlaylist)
	Returns the main library playlist in the main library source.
HRESULT	([,
	Returns the currently targeted track.
HRESULT	CurrentPlaylist ([out, retval] IITPlaylist **iPlaylist)
	Returns the playlist containing the currently targeted track.
HRESULT	(L , 1
	Returns a collection containing the currently selected track or tracks.
HRESULT	, ,
	Returns the version of the iTunes application.
HRESULT	AppCommandMessageProcessingEnabled ([out, retval] VARIANT_BOOL *isEnabled)
	Returns true if iTunes will process APPCOMMAND Windows messages.
HRESULT	AppCommandMessageProcessingEnabled ([in] VARIANT_BOOL shouldEnable)

	Controls whether iTunes should process APPCOMMAND Windows messages.
HRESULT	ForceToForegroundOnDialog ([out, retval] VARIANT_BOOL *forceToForegroundOnDialog)
	Returns true if iTunes will force itself to be the foreground application when it displays a dialog.
HRESULT	ForceToForegroundOnDialog ([in] VARIANT_BOOL forceToForegroundOnDialog)
	Controls whether iTunes will force itself to be the foreground application when it displays a dialog.
HRESULT	CanSetShuffle ([in] VARIANT *iPlaylist,[out, retval] VARIANT_BOOL *canSetShuffle)
	Returns true if the IITPlaylist::Shuffle property is writable for the specified playlist.
HRESULT	CanSetSongRepeat ([in] VARIANT *iPlaylist,[out, retval] VARIANT_BOOL *canSetSongRepeat)
	Returns true if the IITPlaylist::SongRepeat property is writable for the specified playlist.
HRESULT	ConvertOperationStatus ([out, retval] IITConvertOperationStatus **iStatus)
	Returns an IITConvertOperationStatus object if there is currently a conversion in progress.
HRESULT	SoundVolumeControlEnabled ([out, retval] VARIANT_BOOL *isEnabled) Returns true if the sound volume control is enabled.
HRESULT	LibraryXMLPath ([out, retval] BSTR *filePath)
	Returns the full path to the current iTunes library XML file. The default file is "iTunes Music Library.xml" inside the "iTunes" folder in the user's "My Music" folder.
HRESULT	ITObjectPersistentIDHigh ([in] VARIANT *iObject,[out, retval] long *highID)
	Returns the high 32 bits of the persistent ID of the specified IITObject . See the documentation on IITObject for more information on persistent IDs.
HRESULT	ITObjectPersistentIDLow ([in] VARIANT *iObject,[out, retval] long *lowID)
	Returns the low 32 bits of the persistent ID of the specified IITObject . See the documentation on IITObject for more information on persistent IDs.

Member Function Documentation

HRESULT liTunes::AppCommandMessageProcessingEnabled ([in]

Controls whether iTunes should process APPCOMMAND Windows messages.

If true, iTunes will process Windows APPCOMMAND messages such as APPCOMMAND_MEDIA_PLAY, APPCOMMAND_MEDIA_PAUSE, APPCOMMAND_MEDIA_NEXTTRACK, etc. Typically, these messages are generated by special keys on multimedia keyboards.

Note:

This property is available in iTunes 4.6 and later (iTunes type library 1.1 and later).

Parameters:

shouldEnable True if iTunes should process APPCOMMAND Windows messages.

Return values:

S_OK The operation was successful.

E_FAIL An unexpected error occurred.

HRESULT liTunes::AppCommandMessageProcessingEnabled ([ou

Returns true if iTunes will process APPCOMMAND Windows messages.

If true, iTunes will process Windows APPCOMMAND messages such as APPCOMMAND_MEDIA_PLAY, APPCOMMAND_MEDIA_PAUSE, APPCOMMAND_MEDIA_NEXTTRACK, etc. Typically, these messages are generated by special keys on multimedia keyboards.

Note:

This property is available in iTunes 4.6 and later (iTunes type library 1.1 and later).

Parameters:

isEnabled Returns true if iTunes will process APPCOMMAND Windows messages.

Return values:

S_OK The operation was successful.

E POINTER is Enabled is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::BackTrack ()

Reposition to the beginning of the current track or go to the previous track if already at start of current track.

Note:

As of iTunes 4.7 and later (iTunes type library 1.2 and later), iTunes will suppress the authorization dialog if this method attempts to start playing a purchased track that has not been authorized for playback on the current machine. The unauthorized track is skipped in this case.

Return values:

S_OK The operation was successful.

E_FAIL An unexpected error occurred.

HRESULT liTunes::BrowserWindow ([out, retval] IITBrowserWindo

Returns the main iTunes browser window.

Parameters:

iBrowserWindow An **IITBrowserWindow** object corresponding to the main iTunes browser window.

Return values:

S_OK The operation was successful.

E POINTER iBrowserWindow is NULL.

E_FAIL An unexpected error occurred.

Returns true if the **IITPlaylist::Shuffle** property is writable for the specified playlist.

Note:

This property is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Parameters:

iPlaylist A VARIANT of type VT_DISPATCH that points to

an IITPlaylist.

canSetShuffle True if the IITPlaylist::Shuffle property is writable

for the specified playlist.

Return values:

S_OK The operation was successful.

E_INVALIDARG iPlaylist is not the correct

VARIANT type.

E_POINTER iPlaylist or canSetShuffle is

NULL.

ITUNES_E_OBJECTDELETED The playlist has been deleted.

Returns true if the **IITPlaylist::SongRepeat** property is writable for the specified playlist.

Note:

This property is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Parameters:

iPlaylist A VARIANT of type VT DISPATCH that

points to an IITPlaylist.

canSetSongRepeat True if the IITPlaylist::SongRepeat property

is writable for the specified playlist.

Return values:

S_*OK* The operation was successful.

E_INVALIDARG iPlaylist is not the correct

VARIANT type.

E_POINTER iPlaylist or canSetShuffle is

NULL.

ITUNES_E_OBJECTDELETED The playlist has been deleted.

E FAIL An unexpected error occurred.

```
HRESULT liTunes::CheckVersion ( [in] long [in] long [in] long [out, retval] VARIANT_BOOL * i
```

Returns true if this version of the iTunes type library is compatible with the specified version.

The current major version is **kITTypeLibrary_MajorVersion**. The current minor version is **kITTypeLibrary_MinorVersion**.

iTunes 4.5 included version 1.0 of the iTunes type library. iTunes 4.6 included version 1.1 of the iTunes type library.

Parameters:

majorVersion Major version of iTunes interface.minorVersion Minor version of iTunes interface.isCompatible Returns true if iTunes is compatible with the indicated interface version.

Return values:

S_OK The operation was successful.

E_POINTER is Compatible is NULL.

E_FAIL An unexpected error occurred.

Start converting the specified file or folder.

The file or files will added to the main library after conversion. For a file on an audio CD, this is equivalent to importing the song.

Use **liTunes::CurrentEncoder()** to set the current encoder before converting.

Note:

As of iTunes 4.6 and later (iTunes type library 1.1 and later),

you should use **liTunes::ConvertFile2()**, which returns an **IITConvertOperationStatus** object to allow clients to retrieve additional conversion progress information.

Parameters:

filePath The full path to the file or folder to convert.

iStatus Returns an IITOperationStatus object corresponding to

the asynchronous operation.

Return values:

S_OK The operation was

successful.

E POINTER filePath or iStatus is

NULL.

E_INVALIDARG filePath is empty or

too long.

ITUNES_E_CONVERSIONINPROGRESS There is already a

conversion in

progress.

E_FAIL An unexpected error

occurred.

HRESULT liTunes::ConvertFile2 ([in] BSTR

[out, retval] IITConvertOperations

)

Start converting the specified file or folder.

The file or files will added to the main library after conversion. For a file on an audio CD, this is equivalent to importing the song.

Use **liTunes::CurrentEncoder()** to set the current encoder before converting.

Note:

This method is available in iTunes 4.6 and later (iTunes type library 1.1 and later).

Parameters:

filePath The full path to the file or folder to convert.

iStatus Returns an IITConvertOperationStatus object

corresponding to the asynchronous operation.

Return values:

S_OK The operation was

successful.

E POINTER filePath or iStatus is

NULL.

E_INVALIDARG filePath is empty or

too long.

ITUNES_E_CONVERSIONINPROGRESS There is already a

conversion in

progress.

E_FAIL An unexpected error

occurred.

HRESULT liTunes::ConvertFiles ([in] VARIANT *

[out, retval] IITOperationStatus **

)

Start converting the specified files and/or folders.

The files will added to the main library after conversion. For files on an audio CD, this is equivalent to importing the songs.

Use **liTunes::CurrentEncoder()** to set the current encoder before converting.

Note:

As of iTunes 4.6 and later (iTunes type library 1.1 and later), you should use liTunes::ConvertFiles2(), which returns an IITConvertOperationStatus object to allow clients to retrieve additional conversion progress information.

Parameters:

filePaths The single-dimensioned array of full paths to each file or

folder. filePaths can be of type

VT ARRAY|VT VARIANT, where each entry is a

VT_BSTR, or VT_ARRAY|VT_BSTR. You can also pass

a JScript Array object.

iStatus Returns an IITOperationStatus object corresponding to

the asynchronous operation.

Return values:

S *OK* The operation was

successful.

E_POINTER filePaths or iStatus is

NULL.

E_INVALIDARG filePaths is not the

correct VARIANT

type.

ITUNES E CONVERSIONINPROGRESS There is already a

conversion in

progress.

E_FAIL An unexpected error

occurred.

Start converting the specified files and/or folders.

The files will added to the main library after conversion. For files on an audio CD, this is equivalent to importing the songs.

Use **liTunes::CurrentEncoder()** to set the current encoder before converting.

Note:

This method is available in iTunes 4.6 and later (iTunes type library 1.1 and later).

Parameters:

filePaths The single-dimensioned array of full paths to each file or

folder. filePaths can be of type

VT ARRAY|VT VARIANT, where each entry is a

VT_BSTR, or VT_ARRAY|VT_BSTR. You can also pass

a JScript Array object.

iStatus Returns an IITConvertOperationStatus object

corresponding to the asynchronous operation.

Return values:

S_*OK* The operation was

successful.

E POINTER filePaths or iStatus is

NULL.

E INVALIDARG filePaths is not the

correct VARIANT

type.

ITUNES_E_CONVERSIONINPROGRESS There is already a

conversion in

progress.

E_FAIL An unexpected error

occurred.

HRESULT liTunes::ConvertOperationStatus ([out, retval] IITConve

Returns an **IITConvertOperationStatus** object if there is currently a conversion in progress.

Note:

This property is available in iTunes 4.8 and later (iTunes type library 1.3 and later).

Parameters:

iStatus An IITConvertOperationStatus object which can be used to monitor the current conversion operation. Will be set to NULL if there is no conversion operation in progress.

Return values:

S_OK The operation was successful.

S_FALSE There is no conversion operation in progress.

E POINTER iStatus is NULL.

E_FAIL An unexpected error occurred.

Start converting the specified track.

For a track in an audio CD playlist, this is equivalent to importing the song.

Use **liTunes::CurrentEncoder()** to set the current encoder before converting.

Note:

As of iTunes 4.6 and later (iTunes type library 1.1 and later), you should use liTunes::ConvertTrack2(), which returns an IITConvertOperationStatus object to allow clients to retrieve

additional conversion progress information.

Parameters:

iTrackToConvert The track to convert. This must be a VARIANT

of type VT_DISPATCH that points to an

IITTrack.

iStatus Returns an IITOperationStatus object

corresponding to the asynchronous operation.

Return values:

S *OK* The operation was

successful.

E POINTER iTrackToConvert or

iStatus is NULL.

E_INVALIDARG iTrackToConvert is

not the correct VARIANT type.

ITUNES E CONVERSIONINPROGRESS There is already a

conversion in

progress.

E_FAIL An unexpected error

occurred.

HRESULT liTunes::ConvertTrack2 ([in] VARIANT *

[out, retval] IITConvertOperatio

)

Start converting the specified track.

For a track in an audio CD playlist, this is equivalent to importing the song.

Use **liTunes::CurrentEncoder()** to set the current encoder before converting.

Note:

This method is available in iTunes 4.6 and later (iTunes type library 1.1 and later).

Parameters:

iTrackToConvert The track to convert. This must be a VARIANT

of type VT_DISPATCH that points to an

IITTrack.

iStatus Returns an IITConvertOperationStatus object

corresponding to the asynchronous operation.

Return values:

S_*OK* The operation was

successful.

E_POINTER iTrackToConvert or

iStatus is NULL.

E_INVALIDARG iTrackToConvert is

not the correct VARIANT type.

ITUNES E CONVERSIONINPROGRESS There is already a

conversion in progress.

E_FAIL An unexpected error

occurred.

HRESULT liTunes::ConvertTracks ([in] VARIANT *

[out, retval] IITOperationStatus

)

Start converting the specified tracks.

For a tracks in an audio CD playlist, this is equivalent to importing the songs.

Use **liTunes::CurrentEncoder()** to set the current encoder before converting.

Note:

As of iTunes 4.6 and later (iTunes type library 1.1 and later), you should use liTunes::ConvertTracks2(), which returns an IITConvertOperationStatus object to allow clients to retrieve additional conversion progress information.

Parameters:

iTracksToConvert The tracks to convert. This must be a VARIANT

of type VT DISPATCH that points to an

IITTrackCollection.

iStatus Returns an **IITOperationStatus** object

corresponding to the asynchronous operation.

Return values:

S_*OK* The operation was

successful.

E_POINTER iTracksToConvert or

iStatus is NULL.

E_INVALIDARG iTracksToConvert is

not the correct VARIANT type.

ITUNES_E_CONVERSIONINPROGRESS There is already a

conversion in

progress.

E_FAIL An unexpected error

occurred.

HRESULT liTunes::ConvertTracks2 ([in] VARIANT *

[out, retval] IITConvertOperati

)

Start converting the specified tracks.

For a tracks in an audio CD playlist, this is equivalent to importing the songs.

Use **liTunes::CurrentEncoder()** to set the current encoder before converting.

Note:

This method is available in iTunes 4.6 and later (iTunes type library 1.1 and later).

Parameters:

iTracksToConvert The tracks to convert. This must be a VARIANT

of type VT DISPATCH that points to an

IITTrackCollection.

iStatus Returns an IITConvertOperationStatus object

corresponding to the asynchronous operation.

Return values:

S_OK The operation was

successful.

E_POINTER iTracksToConvert or

iStatus is NULL.

E_INVALIDARG iTracksToConvert is

not the correct VARIANT type.

ITUNES_E_CONVERSIONINPROGRESS There is already a

conversion in

progress.

E_FAIL An unexpected error

occurred.

HRESULT liTunes::CreateEQPreset ([in] BSTR

eq

[out, retval] IITEQPreset ** iE

Creates a new EQ preset.

The EQ preset will be created "flat", i.e. the preamp and all band levels will be set to 0.

EQ preset names cannot start with leading spaces. If you specify a name that starts with leading spaces they will be stripped out.

If eqPresetName is NULL or empty, the EQ preset will be created with a default name.

Note:

This method is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Parameters:

eqPresetName The name of the new EQ preset (may be NULL or

empty).

iEQPreset Returns an **IITEQPreset** object corresponding to

the new EQ preset.

Return values:

S_OK The operation was successful.

E_POINTER iEQPreset is NULL.

ITUNES_E_OBJECTEXISTS An EQ preset with this name

already exists.

E_FAIL An unexpected error occurred.

Creates a new folder in the main library.

If folderName is NULL or empty, the folder will be created with a default name.

Note:

This method is available in iTunes 6.0.2 and later (iTunes type library 1.7 and later).

Parameters:

folderName The name of the new folder (may be NULL or

empty).

iFolder Returns an **IITPlaylist** object corresponding to the

new folder.

Return values:

S_*OK* The operation was successful.

E_POINTER iFolder is NULL.

E_FAIL An unexpected error occurred.

Creates a new folder in an existing source.

You may not be able to create a folder in every source. For example, you cannot create a folder in an audio CD source, or in an iPod source if it is in auto update mode.

If folderName is NULL or empty, the folder will be created with a default name.

Note:

This method is available in iTunes 6.0.2 and later (iTunes type library 1.7 and later).

Parameters:

folderName The name of the new folder (may be NULL or

empty).

iSource The source that will contain the new folder. This is a

VARIANT of type VT_DISPATCH that points to an

IITSource.

iFolder Returns an **IITPlaylist** object corresponding to the

new folder.

Return values:

S_OK The operation was successful.

E_POINTER iSource or iFolder is NULL. *E_INVALIDARG* iSource is not the correct

VARIANT type.

ITUNES_E_OBJECTLOCKED A folder cannot be created in this

source.

ITUNES_E_OBJECTDELETED The source has been deleted.

E_FAIL An unexpected error occurred.

Creates a new playlist in the main library.

If playlistName is NULL or empty, the playlist will be created with a default name.

Parameters:

playlistName The name of the new playlist (may be NULL or empty).

iPlaylist Returns an **IITPlaylist** object corresponding to the

new playlist.

Return values:

S_OK The operation was successful.

E_POINTER iPlaylist is NULL.

E_FAIL An unexpected error occurred.

Creates a new playlist in an existing source.

You may not be able to create a playlist in every source. For example, you cannot create a playlist in an audio CD source, or in an iPod source if it is in auto update mode.

If playlistName is NULL or empty, the playlist will be created with a default name.

Note:

This method is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Parameters:

playlistName The name of the new playlist (may be NULL or

empty).

iSource The source that will contain the new playlist. This is

a VARIANT of type VT_DISPATCH that points to an

IITSource.

iPlaylist Returns an **IITPlaylist** object corresponding to the

new playlist.

Return values:

S_OK The operation was successful.

E POINTER iSource or iPlaylist is NULL.

E INVALIDARG iSource is not the correct

VARIANT type.

ITUNES_E_OBJECTLOCKED A playlist cannot be created in

this source.

ITUNES_E_OBJECTDELETED The source has been deleted.

E_FAIL An unexpected error occurred.

HRESULT liTunes::CurrentEncoder ([in] IITEncoder * iEncoder)

Sets the current encoder (AAC, MP3, AIFF, WAV, etc.).

Parameters:

iEncoder An **IITEncoder** object corresponding to the new encoder.

Return values:

S_*OK* The operation was successful.

E_POINTER iEncoder is NULL.

ITUNES_E_OBJECTDELETED The encoder corresponding to

iEncoder is not available.

E_FAIL An unexpected error occurred.

HRESULT liTunes::CurrentEncoder ([out, retval] IITEncoder ** iEn

Returns the currently selected encoder (AAC, MP3, AIFF, WAV, etc.).

Parameters:

iEncoder Returns an **IITEncoder** object corresponding to the

current encoder.

Return values:

S_OK The operation was successful.

E POINTER iEncoder is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::CurrentEQPreset ([in] IITEQPreset * iEQPreset

Sets the current EQ preset.

Parameters:

iEQPreset An **IITEQPreset** object corresponding to the new EQ preset.

Return values:

S_OK The operation was successful.

E_POINTER iEQPreset is NULL.

ITUNES_E_OBJECTDELETED The EQ preset corresponding to

iEQPreset is not available.

E_FAIL An unexpected error occurred.

HRESULT liTunes::CurrentEQPreset ([out, retval] IITEQPreset ** il

Returns the currently selected EQ preset.

Parameters:

iEQPreset Returns an **IITEQPreset** object corresponding to the current EQ preset.

Return values:

S_OK The operation was successful.

E_POINTER iEQPreset is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::CurrentPlaylist ([out, retval] IITPlaylist ** iPlayl

Returns the playlist containing the currently targeted track.

Parameters:

iPlaylist An **IITPlaylist** object corresponding to the playlist containing the currently targeted track. Will be set to NULL if there is no currently targeted playlist.

Return values:

S_OK The operation was successful.

S_FALSE There is no currently targeted track, so there is no

currently targeted playlist.

E_POINTER iPlaylist is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::CurrentStreamTitle ([out, retval] BSTR * stream

Returns the name of the current song in the playing stream (provided by streaming server).

Parameters:

streamTitle Returns the name of the current song in the playing stream. Will be set to NULL if no stream title is available.

Return values:

S_OK The operation was successful.

E_POINTER streamTitle is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::CurrentStreamURL ([out, retval] BSTR * stream

Returns the URL of the playing stream or streaming web site (provided by streaming server).

Parameters:

streamURL Returns the URL of the playing stream or streaming web site. Will be set to NULL if no stream URL is available.

Return values:

S_OK The operation was successful.

E POINTER streamURL is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::CurrentTrack ([out, retval] | ITTrack ** iTrack)

Returns the currently targeted track.

Parameters:

iTrack An **IITTrack** object corresponding to the currently targeted track. Will be set to NULL if there is no currently targeted track.

Return values:

S_OK The operation was successful.

S_FALSE There is no currently targeted track.

E_POINTER iTrack is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::CurrentVisual ([in] IITVisual * iVisual)

Sets the current visual plug-in.

Parameters:

iVisual An IITVisual object corresponding to the new visual plugin.

Return values:

S_OK The operation was successful.

E_POINTER iVisual is NULL.

ITUNES_E_OBJECTDELETED The visual plug-in corresponding

to iVisual is not available.

E_FAIL An unexpected error occurred.

HRESULT liTunes::CurrentVisual ([out, retval] IITVisual ** iVisual]

Returns the currently selected visual plug-in.

Parameters:

iVisual Returns an **IITVisual** object corresponding to the current visual plug-in.

Return values:

S_*OK* The operation was successful.

E_POINTER iVisual is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::Encoders ([out, retval] IITEncoderCollection **

Returns a collection of encoders.

Parameters:

iEncoderCollection Collection of IITEncoder objects.

Return values:

S_OK The operation was successful.

E POINTER iEncoderCollection is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::EQEnabled ([in] VARIANT_BOOL shouldEnable

Set whether the equalizer should be enabled.

Parameters:

shouldEnable True if the equalizer should be enabled.

Return values:

S_OK The operation was successful.

E_FAIL An unexpected error occurred.

HRESULT liTunes::EQEnabled ([out, retval] VARIANT_BOOL * isE

Returns true if the equalizer is enabled.

Parameters:

isEnabled True if the equalizer is enabled.

Return values:

S_OK The operation was successful.

E_POINTER is Enabled is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::EQPresets ([out, retval] IITEQPresetCollection:

Returns a collection of EQ presets.

Parameters:

iEQPresetCollection Collection of IITEQPreset objects.

Return values:

S_OK The operation was successful.

E_POINTER iEQPresetCollection is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::EQWindow ([out, retval] IITWindow ** iEQWind

Returns the EQ window.

Parameters:

iEQWindow An **IITWindow** object corresponding to the EQ window.

Return values:

S_OK The operation was successful.

E_POINTER iEQWindow is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::FastForward ()

Skip forward in a playing track.

Return values:

S_OK The operation was successful.

HRESULT liTunes::ForceToForegroundOnDialog ([in] VARIANT_B(

Controls whether iTunes will force itself to be the foreground application when it displays a dialog.

This property is false by default, since typically a user will not want iTunes to come to the foreground if it isn't already the foreground application. For example, a user may be listening to a radio stream in the background, and a progress dialog could be displayed if the stream needs to be rebuffered. Since this dialog will go away when the rebuffering is complete, there is no reason for iTunes be forced to the foreground.

Note:

This property is available in iTunes 4.6 and later (iTunes type library 1.1 and later).

Parameters:

forceDialogsToForeground True if iTunes should force itself to be the foreground application when it displays a dialog.

Return values:

S_OK The operation was successful.

E_FAIL An unexpected error occurred.

HRESULT liTunes::ForceToForegroundOnDialog ([out, retval] VAR

Returns true if iTunes will force itself to be the foreground application when it displays a dialog.

This property is false by default, since typically a user will not want iTunes to come to the foreground if it isn't already the foreground application. For example, a user may be listening to a radio stream in the background, and a progress dialog could be displayed if the stream needs to be rebuffered. Since this dialog will go away when the rebuffering is complete, there is no reason for iTunes be forced to the foreground.

Note:

This property is available in iTunes 4.6 and later (iTunes type library 1.1 and later).

Parameters:

forceDialogsToForeground Returns true if iTunes will force itself to be the foreground application when it displays a dialog.

Return values:

S_OK The operation was successful.

E_POINTER forceDialogsToForeground is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::FullScreenVisuals ([in] VARIANT BOOL should

Set whether visuals should be displayed using the entire screen.

Parameters:

shouldUseFullScreen True if the visuals should be displayed using the entire screen.

Return values:

S_OK The operation was successful.

E_ACCESSDENIED This setting cannot be changed currently.

E_FAIL An unexpected error occurred.

HRESULT liTunes::FullScreenVisuals ([out, retval] VARIANT_BOO

Returns true if the visuals are displayed using the entire screen.

Parameters:

isFullScreen True if the visuals are displayed using the entire screen.

Return values:

S_OK The operation was successful.

E POINTER isFullScreen is NULL.

E FAIL An unexpected error occurred.

```
HRESULT liTunes::GetITObjectByID ( [in] long | play!
| [in] long | track|
| [in] long | data!
| [out, retval] IITObject ** iObject **
```

Returns an IITObject corresponding to the specified IDs.

The object may be a source, playlist, or track.

Parameters:

sourceID The ID that identifies the source. Valid for a source,

playlist, or track.

playlistID The ID that identifies the playlist. Valid for a playlist

or track. Must be zero for a source.

trackID The ID that identifies the track within the playlist.

Valid for a track. Must be zero for a source or playlist.

databaseID The ID that identifies the track, independent of its

playlist. Valid for a track. Must be zero for a source or

playlist.

iObject Returns an **IITObject** object corresponding to the

specified IDs. Will be set to NULL if no object could

be retrieved.

Return values:

S_*OK* The operation was successful.

S FALSE An object with the specified IDs could not be

found.

E INVALIDARG Invalid combination of IDs.

E_POINTER iObject is NULL.

E_FAIL An unexpected error occurred.

```
HRESULT liTunes::GetITObjectPersistentIDs ( [in] VARIANT * iObjectPersistentIDs ( [in] VARIANT * iObjectPers
```

Returns the high and low 32 bits of the persistent ID of the specified **IITObject**. See the documentation on **IITObject** for more information on persistent IDs.

This method is not usable from scripting clients, since it returns multiple output parameters. Scripting clients must use

IiTunes::ITObjectPersistentIDHigh() and

IiTunes::ITObjectPersistentIDLow() properties instead.

The object may be a source, playlist, or track.

Note:

This property is available in iTunes 7.7 and later (iTunes type library 1.11 and later).

Parameters:

iObject A VARIANT of type VT_DISPATCH that points to an **IITObject**.

highID The high 32 bits of the 64-bit persistent ID. *lowID* The low 32 bits of the 64-bit persistent ID.

Return values:

S_OK The operation was successful. E_POINTER iObject, highID, or lowID is NULL.

E_INVALIDARG iObject is not the correct VARIANT type.

E_FAIL An unexpected error occurred.

Retrieves the current state of the player buttons in the window containing the currently targeted track. If there is no currently targeted track, returns the current state of the player buttons in the main browser window.

This method is not usable from scripting clients, since it returns multiple output parameters.

Note:

This method is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Parameters:

previousEnabled Returns whether the previous button is

enabled.

playPauseStopState Returns whether the play/pause/stop button

is enabled, and whether it is currently

showing play, pause, or stop.

nextEnabled Returns whether the next button is enabled.

Return values:

S_OK The operation was successful.

E_POINTER previousEnabled, playPauseStopState, or

nextEnabled is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::GotoMusicStoreHomePage ()

Go to the iTunes Store home page.

If the iTunes Store is already open in a separate window, the home page will be displayed there. Otherwise, it will be displayed in the main browser window. The window containing the store will be brought to the foreground.

Return values:

S_OK The operation was

successful.

ITUNES_E_MUSICSTOREDISABLED Access to the iTunes

Store is disabled in the

preferences.

E_FAIL An unexpected error

occurred.

```
HRESULT liTunes::ITObjectPersistentIDHigh ( [in] VARIANT * i( [out, retval] long * h )
```

Returns the high 32 bits of the persistent ID of the specified **IITObject**.

See the documentation on **IITObject** for more information on persistent IDs.

This property and **liTunes::ITObjectPersistentIDLow()** are meant to be used from scripting clients, since they can't handle multiple output parameters. If you don't need an automation-compatible interface, use the **liTunes::GetITObjectPersistentIDs()** method instead.

The object may be a source, playlist, or track.

Note:

This property is available in iTunes 7.7 and later (iTunes type library 1.11 and later).

Parameters:

iObject A VARIANT of type VT_DISPATCH that points to an **IITObject**.

highID The high 32 bits of the 64-bit persistent ID.

Return values:

S_*OK* The operation was successful.

E_POINTER iObject or highID is NULL.

E_INVALIDARG iObject is not the correct VARIANT type.

E_FAIL An unexpected error occurred.

```
HRESULT liTunes::ITObjectPersistentIDLow ( [in] VARIANT * iC [out, retval] long * lo
```

Returns the low 32 bits of the persistent ID of the specified **IITObject**. See the documentation on **IITObject** for more information on persistent IDs.

This property and **liTunes::ITObjectPersistentIDHigh()** are meant to be used from scripting clients, since they can't handle multiple output

parameters. If you don't need an automation-compatible interface, use the **liTunes::GetITObjectPersistentIDs()** method instead.

The object may be a source, playlist, or track.

Note:

This property is available in iTunes 7.7 and later (iTunes type library 1.11 and later).

Parameters:

iObject A VARIANT of type VT_DISPATCH that points to an **IITObject**.

lowID The low 32 bits of the 64-bit persistent ID.

Return values:

S_OK The operation was successful.

E_POINTER iObject or highID is NULL.

E_INVALIDARG iObject is not the correct VARIANT type.

E_FAIL An unexpected error occurred.

HRESULT liTunes::LibraryPlaylist ([out, retval] IITLibraryPlaylist **

Returns the main library playlist in the main library source.

Parameters:

iLibraryPlaylist An IITLibraryPlaylist object corresponding to the main library playlist.

Return values:

S_OK The operation was successful.

E_POINTER iLibraryPlaylist is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::LibrarySource ([out, retval] IITSource ** iLibrar

Returns the source that represents the main library.

You can also find the main library source by iterating over **IiTunes::Sources()** and looking for an **IITSource** of kind ITSourceKindLibrary.

Parameters:

iLibrarySource An **IITSource** object corresponding to the main library.

Return values:

S_*OK* The operation was successful.

E_POINTER iLibrarySource is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::LibraryXMLPath ([out, retval] BSTR * filePath)

Returns the full path to the current iTunes library XML file. The default file is "iTunes Music Library.xml" inside the "iTunes" folder in the user's "My Music" folder.

Note:

This property is available in iTunes 7.0 and later (iTunes type library 1.8 and later).

Parameters:

filePath The full path to the current iTunes library XML file.

Return values:

S_OK The operation was successful.

E_POINTER filePath is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::Mute ([in] VARIANT_BOOL shouldMute)

Sets sound output mute state.

Parameters:

shouldMute If true, sound output will be muted.

Return values:

S_OK The operation was successful.

E_FAIL An unexpected error occurred.

HRESULT liTunes::Mute ([out, retval] VARIANT_BOOL * isMuted)

Returns true if the sound output is muted.

Parameters:

isMuted True is sound output is muted.

Return values:

S_OK The operation was successful.

E_POINTER isMuted is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::NextTrack()

Advance to the next track in the current playlist.

Note:

As of iTunes 4.7 and later (iTunes type library 1.2 and later), iTunes will suppress the authorization dialog if this method

attempts to start playing a purchased track that has not been authorized for playback on the current machine. The unauthorized track is skipped in this case.

Return values:

S_OK The operation was successful.

E_FAIL An unexpected error occurred.

HRESULT liTunes::OpenURL ([in] BSTR url)

Open the specified iTunes Store or streaming audio URL.

If the specified URL is an iTunes Store URL, the window containing the store will be brought to the foreground. If the iTunes Store is already open in a separate window, the home page will be displayed there. Otherwise, it will be displayed in the main browser window.

Parameters:

url The URL to open. The length of the URL cannot exceed 512 characters. iTunes Store URLs start with itms:// or itmss://.
Streaming audio URLs start with http://.

Return values:

S_OK The operation was successful.

E_POINTER url is NULL.

E_INVALIDARG url has an invalid length.

E_FAIL An unexpected error occurred.

HRESULT liTunes::Pause ()

Pause playback.

Return values:

S_OK The operation was successful.

E_FAIL An unexpected error occurred.

HRESULT liTunes::Play()

Play the currently targeted track.

Return values:

S_OK The operation was successful.

E_FAIL An unexpected error occurred.

```
HRESULT liTunes::PlayerButtonClicked ( [in] ITPlayerButton playe [in] long playe )
```

Simulate a click on a player control button in the window containing the currently targeted track. If there is no currently targeted track, simulates a click on a player control button in the main browser window.

You can specify modifier keys to get the special behaviors built into iTunes (e.g. shift-control-next button to go to the next chapter in an audio book).

Note:

This method is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Parameters:

playerButton The player button to click

playerButtonModifierKeys A combination of one or more flags in

ITPlayerButtonModifierKey. Specify ITPlayerButtonModifierKeyNone for

no modifiers.

Return values:

S_OK The operation was successful.

E_INVALIDARG playerButton is invalid.

E_FAIL An unexpected error occurred.

HRESULT liTunes::PlayerPosition ([in] long playerPos)

Sets the player's position within the currently playing track in seconds.

If playerPos specifies a position before the beginning of the track, the position will be set to the beginning. If playerPos specifies a position after the end of the track, the position will be set to the end.

Parameters:

playerPos The player's position within the currently playing track in seconds.

Return values:

S_OK The operation was successful.

E_FAIL An unexpected error occurred, e.g. no track is being played.

HRESULT liTunes::PlayerPosition ([out, retval] long * playerPos)

Returns the player's position within the currently playing track in seconds.

Parameters:

playerPos The player's position within the currently playing track in seconds.

Return values:

S_OK The operation was successful.

E_POINTER playerPos is NULL.

E_FAIL An unexpected error occurred, e.g. no track is being

played.

HRESULT liTunes::PlayerState ([out, retval] ITPlayerState * player

Returns the current player state.

Parameters:

playerState The current player state.

Return values:

S_OK The operation was successful.

E_POINTER playerState is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::PlayFile ([in] BSTR filePath)

Play the specified file path, adding it to the library if not already present.

Parameters:

filePath Full path to the file to be played.

Return values:

S_OK The operation was successful.

E_POINTER filePath is NULL. *E INVALIDARG* filePath is invalid.

E_FAIL An unexpected error occurred.

HRESULT liTunes::PlayPause ()

Toggle the playing/paused state of the current track.

Return values:

S_OK The operation was successful.

E_FAIL An unexpected error occurred.

HRESULT liTunes::PreviousTrack()

Return to the previous track in the current playlist.

Note:

As of iTunes 4.7 and later (iTunes type library 1.2 and later), iTunes will suppress the authorization dialog if this method attempts to start playing a purchased track that has not been authorized for playback on the current machine. The unauthorized

track is skipped in this case.

Return values:

S_OK The operation was successful.

E_FAIL An unexpected error occurred.

HRESULT liTunes::Quit()

Exits the iTunes application.

Return values:

S_OK The operation was successful.

ITUNES_E_USERCANCEL The operation was canceled by the

user.

E_FAIL An unexpected error occurred.

HRESULT liTunes::Resume ()

Disable fast forward/rewind and resume playback, if playing.

Return values:

S_OK The operation was successful.

E_FAIL An unexpected error occurred.

HRESULT liTunes::Rewind ()

Skip backwards in a playing track.

Return values:

S_OK The operation was successful.

E_FAIL An unexpected error occurred.

HRESULT liTunes::SelectedTracks ([out, retval] IITTrackCollection

Returns a collection containing the currently selected track or tracks.

The frontmost visible window in iTunes must be a browser or playlist window. If there is no frontmost visible window (e.g. iTunes is minimized to the system tray), the main browser window is used.

Parameters:

iTrackCollection Collection of IITTrack objects. Will be set to NULL if there is no current selection.

Return values:

S_OK The operation was successful.

S FALSE There is no current selection.

E_POINTER iTrackCollection is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::SoundVolume ([in]long volume)

Sets the sound output volume (0 = minimum, 100 = maximum).

Note that volume will be pinned to the range 0 through 100.

Parameters:

volume New sound output volume

Return values:

S_OK The operation was successful.

E_FAIL An unexpected error occurred.

HRESULT liTunes::SoundVolume ([out, retval] long * volume)

Returns the sound output volume (0 = minimum, 100 = maximum).

Parameters:

volume Current sound output volume

Return values:

S_OK The operation was successful.

E POINTER volume is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::SoundVolumeControlEnabled ([out, retval] VAF

Returns true if the sound volume control is enabled.

Note:

This property is available in iTunes 6.0.2 and later (iTunes type library 1.7 and later).

Parameters:

is Enabled True if the sound volume control is enabled.

Return values:

S_OK The operation was successful.

E_POINTER is Enabled is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::Sources ([out, retval] IITSourceCollection ** iS

Returns a collection of music sources (music library, CD, device, etc.).

Parameters:

iSourceCollection Collection of IITSource objects.

Return values:

S_OK The operation was successful.

E POINTER iSourceCollection is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::Stop()

Stop playback.

Return values:

S_OK The operation was successful.

E_FAIL An unexpected error occurred.

HRESULT liTunes::SubscribeToPodcast ([in] BSTR url)

Subscribes to the specified podcast feed URL. Any "unsafe" characters in the URL should already be converted into their corresponding escape sequences, iTunes will not do this.

Note:

This method is available in iTunes 4.9 and later (iTunes type library 1.4 and later).

Parameters:

url The URL to subscribe to.

Return values:

S *OK* The operation was

successful.

E_POINTER url is NULL.

E_INVALIDARG url has an invalid length.

ITUNES_E_PODCASTSDISABLED Access to podcasts is

disabled in the preferences.

E_FAIL An unexpected error

occurred.

HRESULT liTunes::UpdatelPod ()

Update the contents of the iPod.

Return values:

S_OK The operation was successful.

S_FALSE No iPod source was found.

HRESULT liTunes::UpdatePodcastFeeds ()

Updates all podcast feeds. This is equivalent to the user pressing the Update button when Podcasts is selected in the Source list.

Note:

This method is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Return values:

S_OK The operation was

successful.

ITUNES_E_PODCASTSDISABLED Access to podcasts is

disabled in the preferences.

E_FAIL An unexpected error

occurred.

HRESULT liTunes::Version ([out, retval] BSTR * version)

Returns the version of the iTunes application.

Parameters:

version Returns the version of the iTunes application

Return values:

S_OK The operation was successful.

E_POINTER version is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::Visuals ([out, retval] IITVisualCollection ** iVisualCollection **

Returns a collection of visual plug-ins.

Parameters:

iVisualCollection Collection of IITVisual objects.

Return values:

S_OK The operation was successful.

E POINTER iVisualCollection is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::VisualsEnabled ([in] VARIANT_BOOL shouldEnabled ([in] VARIANT_BOOL should ([in]

Set whether visuals should be displayed.

Parameters:

shouldEnable True if visuals should be displayed.

Return values:

S OK The operation was successful.

E_ACCESSDENIED Visuals cannot be enabled or disabled. For

example, this can happen if the iTunes

Store is currently being viewed.

E_FAIL An unexpected error occurred.

HRESULT liTunes::VisualsEnabled ([out, retval] VARIANT_BOOL *

Returns true if visuals are currently being displayed.

Parameters:

isEnabled True if visuals are currently being displayed.

Return values:

S_OK The operation was successful.

E_POINTER is Enabled is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::VisualSize ([in] ITVisualSize visualSize)

Sets the size of the displayed visual.

Parameters:

visualSize The new size of the visuals.

Return values:

S_OK The operation was successful.

E_FAIL An unexpected error occurred.

HRESULT liTunes::VisualSize ([out, retval] ITVisualSize * visualSiz

Returns the size of the displayed visual.

Parameters:

visualSize The current size of the visuals.

Return values:

S_*OK* The operation was successful.

E_POINTER visualSize is NULL.

E_FAIL An unexpected error occurred.

HRESULT liTunes::Windows ([out, retval] IITWindowCollection ** /

Returns a collection of visual plug-ins.

Parameters:

iWindowCollection Collection of IITWindow objects.

Return values:

S_*OK* The operation was successful.

E_POINTER iWindowCollection is NULL.

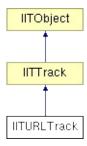
E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITURLTrack Interface Reference

Inheritance diagram for IITURLTrack:



List of all members.

Detailed Description

Represents a URL track.

A URL track references a network audio stream.

Public Member Functions

Properties

HRESULT	URL ([out, retval] BSTR *url) Returns the URL of the stream represented by this track.
HRESULT	URL ([in] BSTR url) Set the URL of the stream represented by this track.
HRESULT	Podcast ([out, retval] VARIANT_BOOL *isPodcast) Returns true if this track is a podcast track. If a podcast track is an IITURLTrack, the podcast episode has not been downloaded.
HRESULT	Category ([out, retval] BSTR *category) Returns the category for the track.
HRESULT	Category ([in] BSTR category) Sets the category for the track.
HRESULT	Description ([out, retval] BSTR *description) Returns the description for the track.
HRESULT	Description ([in] BSTR description) Sets the description for the track.
HRESULT	LongDescription ([out, retval] BSTR *longDescription) Returns the long description for the track.
HRESULT	LongDescription ([in] BSTR longDescription) Sets the long description for the track.
HRESULT	AlbumRating ([out, retval] long *rating) Returns the user or computed rating of the album that this track belongs to (0 to 100). If the album rating has never been set, or has been set to 0, it will be computed based on the ratings of tracks in the album.
HRESULT	AlbumRating ([in] long rating) Set the album rating of the album that this track belongs to (0 to 100). If the album rating is set to 0, it will be computed based on the ratings of tracks in the album.
HRESULT	AlbumRatingKind ([out, retval] ITRatingKind *ratingKind) Returns the album rating kind. If the album rating has never been set, or has been set to 0, the kind is ITRatingKindComputed. Otherwise, the kind is ITRatingKindUser.
HRESULT	RatingKind ([out, retval] ITRatingKind *ratingKind) Returns the track rating kind. If the track rating has never been set, or has been set to 0, the kind is ITRatingKindComputed. Otherwise, the kind is ITRatingKindUser.

HRESULT Playlists ([out, retval] IITPlaylistCollection **iPlaylistCollection)

Returns a collection of playlists that contain the song that this track represents.

Methods

HRESULT	UpdatePodcastFeed ()
	Update the podcast feed for this track. This is equivalent to the user choosing Update Podcast from the contextual menu for the podcast feed that contains this track.
HRESULT	DownloadPodcastEpisode ()
	Start downloading the podcast episode that corresponds to this track. This is

equivalent to the user clicking the Get button next to this track.

HRESULT Reveal ()

Reveals the track in the main browser window.

Member Function Documentation

HRESULT IITURLTrack::AlbumRating ([in] long rating)

Set the album rating of the album that this track belongs to (0 to 100). If the album rating is set to 0, it will be computed based on the ratings of tracks in the album.

Note:

This property is available in iTunes 7.4 and later (iTunes type library 1.10 and later).

Parameters:

rating The new album rating of the album that this track belongs to (0 to 100). If rating is outside this range, it will be pinned.

Return values:

```
S\_OK The operation was successful. 

ITUNES\_E\_OBJECTLOCKED This track is not modifiable. 

ITUNES\_E\_OBJECTDELETED This track has been deleted. 

E\_FAIL An unexpected error occurred.
```

HRESULT IITURLTrack::AlbumRating ([out, retval] long * rating)

Returns the user or computed rating of the album that this track belongs to (0 to 100). If the album rating has never been set, or has been set to 0, it will be computed based on the ratings of tracks in the album.

Note:

This property is available in iTunes 7.4 and later (iTunes type library 1.10 and later).

Parameters:

rating Returns the album rating of the album that this track belongs to (0 to 100).

Return values:

S_OK The operation was successful.

E_POINTER rating is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E FAIL

An unexpected error occurred.

HRESULT IITURLTrack::AlbumRatingKind ([out, retval] ITRatingKi

Returns the album rating kind. If the album rating has never been set, or has been set to 0, the kind is **ITRatingKindComputed**. Otherwise, the kind is **ITRatingKindUser**.

Note:

This property is available in iTunes 7.4 and later (iTunes type library 1.10 and later).

Parameters:

ratingKind Returns the album rating kind.

Return values:

S_OK The operation was successful.

E_POINTER ratingKind is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITURLTrack::Category ([in] BSTR category)

Sets the category for the track.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

category The new category for the track.

Return values:

S_OK The operation was successful.

E_POINTER category is NULL.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITURLTrack::Category ([out, retval] BSTR * category)

Returns the category for the track.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

category Returns the category for the track.

Return values:

S_OK The operation was successful.

E_POINTER category is NULL.

 $ITUNES_E_OBJECTDELETED$ This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITURLTrack::Description ([in] BSTR description)

Sets the description for the track.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

description The new description for the track.

Return values:

S_*OK* The operation was successful.

E_POINTER description is NULL.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITURLTrack::Description ([out, retval] BSTR * descripti

Returns the description for the track.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

description Returns the description for the track.

Return values:

S_*OK* The operation was successful.

E_POINTER description is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITURLTrack::DownloadPodcastEpisode ()

Start downloading the podcast episode that corresponds to this track. This is equivalent to the user clicking the Get button next to this track.

Note:

This method is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Return values:

S_OK The operation was

successful.

E_INVALIDARG This track is not a podcast.

ITUNES_E_PODCASTSDISABLED Access to podcasts is

disabled in the preferences.

E_FAIL An unexpected error

occurred.

HRESULT IITURLTrack::LongDescription ([in] BSTR longDescripti

Sets the long description for the track.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

longDescription The new long description for the track.

Return values:

S_OK The operation was successful.

E_POINTER longDescription is NULL.

ITUNES_E_OBJECTLOCKED This track is not modifiable.

ITUNES_E_OBJECTDELETED This track has been deleted.

HRESULT IITURLTrack::LongDescription ([out, retval] BSTR * long

Returns the long description for the track.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

longDescription Returns the long description for the track.

Return values:

S_OK	The operation was successful.
E_POINTER	longDescription is NULL.
ITUNES_E_OBJECTDELETED	This track has been deleted.
E_FAIL	An unexpected error occurred.

HRESULT IITURLTrack::Playlists ([out, retval] IITPlaylistCollection

Returns a collection of playlists that contain the song that this track represents.

This is the same collection of playlists that are shown in the "Show in Playlist" contextual menu for a track, plus the specific playlist that contains this track.

A track represents a song in a single playlist, use **IITTrack::Playlist()** to get the specific playlist that contains this track.

Note:

This property is available in iTunes 7.7 and later (iTunes type

library 1.11 and later).

Parameters:

iPlaylistCollection Collection of IITPlaylist objects.

Return values:

S_OK The operation was successful.

E_POINTER iPlaylistCollection is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITURLTrack::Podcast ([out, retval] VARIANT_BOOL * is

Returns true if this track is a podcast track. If a podcast track is an IITURLTrack, the podcast episode has not been downloaded.

Note:

This property is available in iTunes 4.9 and later (iTunes type library 1.4 and later).

Parameters:

isPodcast Returns true if this track is a podcast track.

Return values:

S_OK The operation was successful.

E POINTER isPodcast is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITURLTrack::RatingKind ([out, retval] ITRatingKind * ra

Returns the track rating kind. If the track rating has never been set, or

has been set to 0, the kind is **ITRatingKindComputed**. Otherwise, the kind is **ITRatingKindUser**.

Note:

This property is available in iTunes 7.4 and later (iTunes type library 1.10 and later).

Parameters:

ratingKind Returns the track rating kind.

Return values:

S_OK The operation was successful.

E_POINTER ratingKind is NULL.

ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITURLTrack::Reveal ()

Reveals the track in the main browser window.

Note:

This method is available in iTunes 7.4 and later (iTunes type library 1.10 and later).

Return values:

 S_OK The operation was successful. ITUNES_E_OBJECTDELETED This track has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITURLTrack::UpdatePodcastFeed ()

Update the podcast feed for this track. This is equivalent to the user

choosing Update Podcast from the contextual menu for the podcast feed that contains this track.

Note:

This method is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Return values:

S_*OK* The operation was

successful.

E INVALIDARG This track is not a podcast.

ITUNES_E_PODCASTSDISABLED Access to podcasts is

disabled in the preferences.

E FAIL An unexpected error

occurred.

HRESULT IITURLTrack::URL ([in] BSTR url)

Set the URL of the stream represented by this track.

Parameters:

url The URL of the stream represented by this track.

Return values:

S_OK The operation was successful.

E_POINTER url is NULL.

ITUNES_E_OBJECTLOCKED The URL cannot be modified.
ITUNES_E_OBJECTDELETED This track has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITURLTrack::URL ([out, retval] BSTR * url)

Returns the URL of the stream represented by this track.

Parameters:

url Returns the URL of the stream represented by this track.

Return values:

S_OK The operation was successful.

E_POINTER url is NULL.

 $ITUNES_E_OBJECTDELETED$ This track has been deleted.

E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITUserPlaylist Interface Reference

Inheritance diagram for IITUserPlaylist:



List of all members.

Detailed Description

Represents a user-defined playlist.

A user playlist includes both smart and manual user-defined playlists.

Public Member Functions

Methods

HRESULT	AddFile ([in] BSTR filePath,[out, retval] IITOperationStatus **iStatus) Add a file or files inside a folder to the playlist.
HRESULT	AddFiles ([in] VARIANT *filePaths,[out, retval] IITOperationStatus **iStatus) Add an array of files and/or folders to the playlist.
HRESULT	AddURL ([in] BSTR url,[out, retval] IITURLTrack **iURLTrack) Add a streaming audio URL to the playlist.
HRESULT	AddTrack ([in] VARIANT *iTrackToAdd,[out, retval] IITTrack **iAddedTrack) Add an existing track to the playlist.
HRESULT	CreatePlaylist ([in] BSTR playlistName,[out, retval] IITPlaylist **iPlaylist) Creates a new playlist in a folder playlist.
HRESULT	CreateFolder ([in] BSTR folderName,[out, retval] IITPlaylist **iFolder) Creates a new folder in a folder playlist.
HRESULT	Reveal () Reveals the user playlist in the main browser window.

Properties

HRESULT	Shared ([out, retval] VARIANT_BOOL *isShared) Returns true if this playlist is being shared.
HRESULT	Shared ([in] VARIANT_BOOL shouldBeShared) Control whether this playlist should be shared.
HRESULT	Smart ([out, retval] VARIANT_BOOL *isSmart) Returns true if this playlist is a smart playlist.
HRESULT	SpecialKind ([out, retval] ITUserPlaylistSpecialKind *specialKind) Returns the special kind of the playlist.
HRESULT	Parent ([out, retval] IITUserPlaylist **iParentPlayList) Returns the parent folder of the playlist. If the playlist is not inside a folder, it has no parent.
HRESULT	Parent ([in] VARIANT *iParent) Sets the parent folder of the playlist.

Member Function Documentation

Add a file or files inside a folder to the playlist.

You cannot use this method to add a file that requires conversion to be added (e.g. a CD track), use **liTunes::ConvertFile()** or **liTunes::ConvertFile2()** instead. If you add a folder that contains files that require conversion, they will be skipped.

Parameters:

filePath The full path to the file or folder to add.

iStatus Returns an IITOperationStatus object corresponding to

the asynchronous operation. If an error occurs, or no files

were added, this will be set to NULL.

Return values:

S_OK The operation was successful.

E_POINTER filePath or iStatus is NULL.

E_INVALIDARG filePath is empty or too long, or

refers to a file that would require

conversion.

ITUNES_E_OBJECTLOCKED This playlist is read-only.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITUserPlaylist::AddFiles ([in] VARIANT *

[out, retval] IITOperationStatu

)

Add an array of files and/or folders to the playlist.

You cannot use this method to add a file that requires conversion to be added (e.g. a CD track), use **liTunes::ConvertFile()** or **liTunes::ConvertFile2()** instead. If you add a folder that contains files that require conversion, they will be skipped.

Parameters:

filePaths The single-dimensioned array of full paths to each file or

folder. filePaths can be of type

VT_ARRAY|VT_VARIANT, where each entry is a

VT_BSTR, or VT_ARRAY|VT_BSTR. You can also pass

a JScript Array object.

iStatus Returns an IITOperationStatus object corresponding to

the asynchronous operation. If an error occurs, or no

files were added, this will be set to NULL.

Return values:

S_OK The operation was successful.

E POINTER filePaths or iStatus is NULL.

E INVALIDARG filePaths is the wrong VARIANT

type, or includes a file that would

require conversion.

ITUNES_E_OBJECTLOCKED This playlist is read-only.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITUserPlaylist::AddTrack ([in] VARIANT * iTracl [out, retval] IITTrack ** iAdd&
)

Add an existing track to the playlist.

You cannot use this method to add a CD track (ITTrackKindCD) to another playlist, use IiTunes::ConvertTrack() or IiTunes::ConvertTrack2() instead.

You cannot add a shared library track (ITTrackKindSharedLibrary) to another playlist.

Parameters:

iTrackToAdd The track to add. This is a VARIANT of type VT DISPATCH that points to an **IITTrack**.

iAddedTrack Returns an **IITTrack** object corresponding to the new track.

Return values:

S_OK The operation was successful.

E POINTER iTrackToAdd or iAddedTrack is

NULL.

E INVALIDARG iTrackToAdd is the wrong

VARIANT type, or is a CD or

shared track.

ITUNES_E_OBJECTLOCKED This playlist is read-only.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

Add a streaming audio URL to the playlist.

Parameters:

url The URL to add. The length of the URL can be 255

characters or less.

iURLTrack Returns an **IITURLTrack** object corresponding to the new track.

Return values:

S_OK The operation was successful.

E_POINTER url or iURLTrack is NULL.

E_INVALIDARG url is empty or too long.

ITUNES_E_OBJECTLOCKED This playlist is read-only.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITUserPlaylist::CreateFolder ([in] BSTR [out, retval] IITPlaylist **
)

Creates a new folder in a folder playlist.

This playlist must be a folder playlist. If folderName is NULL or empty, the folder will be created with a default name.

Note:

This method is available in iTunes 6.0.2 and later (iTunes type library 1.7 and later).

Parameters:

folderName The name of the new folder (may be NULL or

empty).

iFolder Returns an **IITPlaylist** object corresponding to the

new folder.

Return values:

 S_OK

The operation was successful.

E_POINTER iFolder is NULL.

E_INVALIDARG This playlist is not a folder

playlist.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITUserPlaylist::CreatePlaylist ([in] BSTR

[out, retval] IITPlaylist **

)

Creates a new playlist in a folder playlist.

This playlist must be a folder playlist. If playlistName is NULL or empty, the playlist will be created with a default name.

Note:

This method is available in iTunes 6.0.2 and later (iTunes type library 1.7 and later).

Parameters:

playlistName The name of the new playlist (may be NULL or

empty).

iPlaylist Returns an **IITPlaylist** object corresponding to the

new playlist.

Return values:

S_OK The operation was successful.

E_POINTER iPlaylist is NULL.

E_INVALIDARG This playlist is not a folder

playlist.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITUserPlaylist::Parent ([in] VARIANT * iParent)

Sets the parent folder of the playlist.

iParent is a VARIANT of type VT_DISPATCH that points to a folder **IITUserPlaylist**.

Specifying an empty or NULL iParent will move the playlist outside of all folders.

Note:

This method is available in iTunes 6.0.2 and later (iTunes type library 1.7 and later).

Parameters:

iParent The new parent folder. This is a VARIANT of type VT DISPATCH that points to a folder **IITUserPlaylist**.

Return values:

S_OK	The operation was successful.
E_INVALIDARG	iParent is the wrong VARIANT
	type, or is not a folder playlist.
ITUNES_E_OBJECTDELETED	This playlist has been deleted.
E_FAIL	An unexpected error occurred.

HRESULT IITUserPlaylist::Parent ([out, retval] IITUserPlaylist ** iP

Returns the parent folder of the playlist. If the playlist is not inside a folder, it has no parent.

Note:

This property is available in iTunes 5.0 and later (iTunes type library 1.5 and later).

Parameters:

iParentPlayList Returns an **IITUserPlaylist** object corresponding to the parent folder. Will be set to NULL if this playlist is not inside a folder.

Return values:

S_OK The operation was successful.
S_FALSE This playlist is not inside a folder.

E_POINTER iParentPlayList is NULL.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITUserPlaylist::Reveal ()

Reveals the user playlist in the main browser window.

Note:

This method is available in iTunes 7.4 and later (iTunes type library 1.10 and later).

Return values:

 S_OK The operation was successful. ITUNES_E_OBJECTDELETED This playlist has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITUserPlaylist::Shared ([in] VARIANT_BOOL shouldBe

Control whether this playlist should be shared.

Parameters:

shouldBeShared True if this playlist should be shared.

Return values:

S_*OK* The operation was successful.

E_POINTER shouldBeShared is NULL.

ITUNES_E_OBJECTLOCKED Sharing is disabled, or is

enabled for all playlists (not selected playlists), or playlist

can't be shared.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITUserPlaylist::Shared ([out, retval] VARIANT_BOOL *

Returns true if this playlist is being shared.

Parameters:

isShared Returns true if this playlist is being shared.

Return values:

S_OK The operation was successful.

E_POINTER isShared is NULL.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITUserPlaylist::Smart ([out, retval] VARIANT_BOOL * is

Returns true if this playlist is a smart playlist.

Note that smart playlists cannot be modified.

Parameters:

isSmart Returns true if this playlist is a smart playlist.

Return values:

S_OK The operation was successful.

E_POINTER is Smart is NULL.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITUserPlaylist::SpecialKind ([out, retval] ITUserPlaylist:

Returns the special kind of the playlist.

Note:

This property is available in iTunes 4.9 and later (iTunes type library 1.4 and later).

Parameters:

specialKind Returns the special kind of the playlist.

Return values:

S_OK The operation was successful.

E_POINTER specialKind is NULL.

ITUNES_E_OBJECTDELETED This playlist has been deleted.

E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITVisual Interface Reference

List of all members.

Detailed Description

Represents an iTunes visual plug-in.

You can retrieve or set the currently selected visual plug-in using the **liTunes::CurrentVisual** property.

Public Member Functions

Properties

HRESULT Name ([out, retval] BSTR *name)

Returns the name of the visual plug-in (e.g. "iTunes Visualizer").

Member Function Documentation

HRESULT IITVisual::Name ([out, retval] BSTR * name)

Returns the name of the visual plug-in (e.g. "iTunes Visualizer").

Parameters:

name Returns the name of the visual plug-in (e.g. "iTunes Visualizer").

Return values:

S_OK The operation was successful.

E_POINTER name is NULL.

ITUNES_E_OBJECTDELETED This visual plug-in is not

available.

E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITVisualCollection Interface Reference

List of all members.

Detailed Description

Represents a collection of visual plug-in objects.

The **liTunes::Visuals** property returns the collection of visual plug-ins.

Public Member Functions

Properties

HRESULT	Count ([out, retval] long *count)
	Returns the number of visual plug-ins in the collection.
HRESULT	Item ([in] long index,[out, retval] IITVisual **iVisual)
	Returns an IITVisual object corresponding to the given index (1-based).
HRESULT	ItemByName ([in] BSTR name,[out, retval] IITVisual **iVisual)
	Returns an IITVisual object with the specified name.
HRESULT	_NewEnum ([out, retval] IUnknown **iEnumerator)
	Returns an IEnumVARIANT object which can enumerate the collection.

Member Function Documentation

HRESULT IITVisualCollection::_NewEnum ([out, retval] IUnknown

Returns an IEnumVARIANT object which can enumerate the collection.

Note:

This property is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Parameters:

iEnumerator Returns an IEnumVARIANT object which can enumerate the collection.

Return values:

S_OK The operation was successful.

E POINTER iEnumerator is NULL.

E_FAIL An unexpected error occurred.

HRESULT IITVisualCollection::Count ([out, retval] long * count)

Returns the number of visual plug-ins in the collection.

Parameters:

count Returns the number of visual plug-ins in the collection.

Return values:

S_OK The operation was successful.

E_POINTER count is NULL.

E_FAIL An unexpected error occurred.

```
HRESULT IITVisualCollection::Item ( [in] long index [out, retval] IITVisual ** iVisual ** iVisual **
```

Returns an **IITVisual** object corresponding to the given index (1-based).

Parameters:

index 1-based index of **IITVisual** to retrieve.

iVisual Returns an **IITVisual** object corresponding to the specified index.

Return values:

S *OK* The operation was successful.

S FALSE index is too big.

E INVALIDARG index is less than one.

E POINTER iVisual is NULL.

E_FAIL An unexpected error occurred.

```
HRESULT IITVisualCollection::ItemByName ( [in] BSTR [out, retval] IITVisual )
```

Returns an IITVisual object with the specified name.

Parameters:

name 1-based index of **IITVisual** to retrieve.

iVisual Returns an IITVisual object with the specified name.

Return values:

S_OK The operation was successful.

S_FALSE A visual plug-in with the specified name could not

be found.

E_POINTER name or iVisual is NULL.

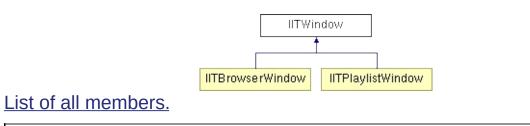
E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITWindow Interface Reference

Inheritance diagram for IITWindow:



Detailed Description

Represents an iTunes window.

You can retrieve all the windows using liTunes::Windows().

Public Member Functions

Properties

HRESULI	Name ([out, retval] BSTR *name) Returns the name (title) of the window.
HRESULT	Kind ([out, retval] ITWindowKind *kind) Returns the kind of the window.
HRESULT	Visible ([out, retval] VARIANT_BOOL *isVisible) Returns true if the window is visible.
HRESULT	Visible ([in] VARIANT_BOOL shouldBeVisible) Hide or show the window.
HRESULT	Resizable ([out, retval] VARIANT_BOOL *isResizable) Returns true if the window is resizable.
HRESULT	Minimized ([out, retval] VARIANT_BOOL *isMinimized) Returns true if the window is minimized.
HRESULT	Minimized ([in] VARIANT_BOOL shouldBeMinimized) Minimize or restore the window.
HRESULT	Maximizable ([out, retval] VARIANT_BOOL *isMaximizable) Returns true if the window is maximizable.
HRESULT	Maximized ([out, retval] VARIANT_BOOL *isMaximized) Returns true if the window is maximized.
HRESULT	Maximized ([in] VARIANT_BOOL shouldBeMaximized) <i>Maximize or restore the window.</i>
HRESULT	Zoomable ([out, retval] VARIANT_BOOL *isZoomable) Returns true if the window is zoomable.
HRESULT	Zoomed ([out, retval] VARIANT_BOOL *isZoomed) Returns true if the window is zoomed.
HRESULT	Zoomed ([in] VARIANT_BOOL shouldBeZoomed) Zoom or unzoom the window.
HRESULT	Top ([out, retval] long *top) Returns the screen coordinate of the top edge of the window.
HRESULT	Top ([in] long top) Set the screen coordinate of the top edge of the window.
HRESULT	Left ([out, retval] long *left) Returns the screen coordinate of the left edge of the window.

HRESULT	Left ([in] long left) Set the screen coordinate of the left edge of the window.
HRESULT	Bottom ([out, retval] long *bottom) Returns the screen coordinate of the bottom edge of the window.
HRESULT	Bottom ([in] long bottom) Set the screen coordinate of the bottom edge of the window.
HRESULT	Right ([out, retval] long *right) Returns the screen coordinate of the right edge of the window.
HRESULT	Right ([in] long right) Set the screen coordinate of the right edge of the window.
HRESULT	Width ([out, retval] long *width) Returns the width of the window.
HRESULT	Width ([in] long width) Set the width of the window.
HRESULT	Height ([out, retval] long *height) Returns the height of the window.
HRESULT	Height ([in] long height) Set the height of the window.

Member Function Documentation

HRESULT IITWindow::Bottom ([in] long bottom)

Set the screen coordinate of the bottom edge of the window.

This will resize, not move, the window. Currently, only the main browser and playlist windows can be resized.

Parameters:

bottom The new screen coordinate of the bottom edge of the window.

Return values:

S_OK

ITUNES_E_OBJECTLOCKED
Window is maximized, or is not a browser or playlist window.

ITUNES_E_OBJECTDELETED
This window has been deleted.

E_FAIL

An unexpected error occurred.

HRESULT IITWindow::Bottom ([out, retval] long * bottom)

Returns the screen coordinate of the bottom edge of the window.

Parameters:

bottom Returns the screen coordinate of the bottom edge of the window.

Return values:

S_OK The operation was successful.

E_POINTER bottom is NULL.

ITUNES E OBJECTDELETED This window has been deleted.

HRESULT IITWindow::Height ([in] long height)

Set the height of the window.

This will resize, not move, the window. It is the equivalent of setting the **IITWindow::Bottom** property to **IITWindow::Top** plus height. Currently, only the main browser and playlist windows can be resized.

Parameters:

height The height of the window.

Return values:

S_OK

ITUNES_E_OBJECTLOCKED
Window is maximized, or is not a browser or playlist window.

ITUNES_E_OBJECTDELETED
This window has been deleted.

E_FAIL

An unexpected error occurred.

HRESULT IITWindow::Height ([out, retval] long * height)

Returns the height of the window.

Parameters:

height Returns the height of the window.

Return values:

S_OK The operation was successful.

E_POINTER height is NULL.

ITUNES_E_OBJECTDELETED This window has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITWindow::Kind ([out, retval] ITWindowKind * kind)

Returns the kind of the window.

Parameters:

kind Returns the kind of the window.

Return values:

S_*OK* The operation was successful.

E_POINTER kind is NULL.

ITUNES_E_OBJECTDELETED This window has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITWindow::Left ([in] long left)

Set the screen coordinate of the left edge of the window.

This will move, not resize, the window.

Parameters:

left The new screen coordinate of the left edge of the window.

Return values:

S_*OK* The operation was successful.

ITUNES_E_OBJECTLOCKED Window is maximized.

ITUNES_E_OBJECTDELETED This window has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITWindow::Left ([out, retval] long * left)

Returns the screen coordinate of the left edge of the window.

Parameters:

left Returns the screen coordinate of the left edge of the window.

Return values:

S_*OK* The operation was successful.

E_POINTER left is NULL.

ITUNES_E_OBJECTDELETED This window has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITWindow::Maximizable ([out, retval] VARIANT_BOOL *

Returns true if the window is maximizable.

Parameters:

isMaximizable Returns true if the window is maximizable.

Return values:

S_OK The operation was successful.

E_POINTER isMaximizable is NULL.

 $ITUNES_E_OBJECTDELETED$ This window has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITWindow::Maximized ([in] VARIANT_BOOL shouldBel

Maximize or restore the window.

Parameters:

shouldBeMaximized True if window should be maximized.

Return values:

S OK ITUNES_E_OBJECTLOCKED Window is not maximizable. ITUNES E OBJECTDELETED This window has been deleted. E FAIL

The operation was successful. An unexpected error occurred.

HRESULT IITWindow::Maximized ([out, retval] VARIANT_BOOL * i

Returns true if the window is maximized.

Parameters:

is Maximized Returns true if the window is maximized.

Return values:

The operation was successful. S OK E POINTER isMaximized is NULL. ITUNES E OBJECTDELETED This window has been deleted. E FAIL An unexpected error occurred.

HRESULT IITWindow::Minimized ([in] VARIANT_BOOL shouldBell

Minimize or restore the window.

Parameters:

shouldBeMinimized True if window should be minimized.

Return values:

The operation was successful. S OK ITUNES E OBJECTDELETED This window has been deleted. E FAIL An unexpected error occurred.

HRESULT IITWindow::Minimized ([out, retval] VARIANT_BOOL * is

Returns true if the window is minimized.

Parameters:

isMinimized Returns true if the window is minimized.

Return values:

S_OK The operation was successful.

E POINTER isMinimized is NULL.

ITUNES_E_OBJECTDELETED This window has been deleted.

E FAIL An unexpected error occurred.

HRESULT IITWindow::Name ([out, retval] BSTR * name)

Returns the name (title) of the window.

Parameters:

name Returns the name (title) of the window.

Return values:

S_OK The operation was successful.

E_POINTER name is NULL.

 $ITUNES_E_OBJECTDELETED$ This window has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITWindow::Resizable ([out, retval] VARIANT_BOOL * is

Returns true if the window is resizable.

Parameters:

isResizable Returns true if the window is resizable.

Return values:

S_OK The operation was successful.

E_POINTER isResizable is NULL.

ITUNES_E_OBJECTDELETED This window has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITWindow::Right ([in] long right)

Set the screen coordinate of the right edge of the window.

This will resize, not move, the window. Currently, only the main browser and playlist windows can be resized.

Parameters:

right The new screen coordinate of the right edge of the window.

Return values:

S_OK

ITUNES_E_OBJECTLOCKED
Window is maximized, or is not a browser or playlist window.

ITUNES_E_OBJECTDELETED
This window has been deleted.

E FAIL

An unexpected error occurred.

HRESULT IITWindow::Right ([out, retval] long * right)

Returns the screen coordinate of the right edge of the window.

Parameters:

right Returns the screen coordinate of the right edge of the window.

Return values:

S_OK The operation was successful.

E_POINTER right is NULL.

ITUNES_E_OBJECTDELETED This window has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITWindow::Top ([in] long top)

Set the screen coordinate of the top edge of the window.

This will move, not resize, the window.

Parameters:

top The new screen coordinate of the top edge of the window.

Return values:

S_OK The operation was successful.

ITUNES_E_OBJECTLOCKED Window is maximized.

ITUNES_E_OBJECTDELETED This window has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITWindow::Top ([out, retval] long * top)

Returns the screen coordinate of the top edge of the window.

Parameters:

top Returns the screen coordinate of the top edge of the window.

Return values:

S_OK The operation was successful.

E_POINTER top is NULL.

 $ITUNES_E_OBJECTDELETED$ This window has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITWindow::Visible ([in] VARIANT_BOOL shouldBeVisib

Hide or show the window.

Note that the main browser window cannot be hidden using COM. However, if the user has minimized the main browser window to the system tray, it will appear to be hidden (as well as minimized). In this case, the main browser window can be restored by setting either the IITWindow::Visible property to true or the IITWindow::Minimized property to false.

Parameters:

shouldBeVisible True if the window should be shown.

Return values:

 S_OK The operation was successful. $E_NOINTERFACE$ This is the main browser window. $ITUNES_E_OBJECTDELETED$ This window has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITWindow::Visible ([out, retval] VARIANT_BOOL * isVis

Returns true if the window is visible.

Parameters:

is Visible Returns true if the window is visible.

Return values:

S OK

The operation was successful.

E_POINTER is Visible is NULL.ITUNES_E_OBJECTDELETED This window has been deleted.E_FAIL An unexpected error occurred.

HRESULT IITWindow::Width ([in] long width)

Set the width of the window.

This will resize, not move, the window. It is the equivalent of setting the **IITWindow::Right** property to **IITWindow::Left** plus width. Currently, only the main browser and playlist windows can be resized.

Parameters:

width The width of the window.

Return values:

S_OK

ITUNES_E_OBJECTLOCKED
Window is maximized, or is not a browser or playlist window.

ITUNES_E_OBJECTDELETED
This window has been deleted.

E FAIL

An unexpected error occurred.

HRESULT IITWindow::Width ([out, retval] long * width)

Returns the width of the window.

Parameters:

width Returns the width of the window.

Return values:

S_OK The operation was successful.

E_POINTER width is NULL.

ITUNES_E_OBJECTDELETED This window has been deleted. E_FAIL An unexpected error occurred.

HRESULT IITWindow::Zoomable ([out, retval] VARIANT_BOOL * is

Returns true if the window is zoomable.

Parameters:

isZoomable Returns true if the window is zoomable.

Return values:

S_OK The operation was successful.

E_POINTER isZoomable is NULL.

ITUNES_E_OBJECTDELETED This window has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITWindow::Zoomed ([in] VARIANT_BOOL shouldBeZoo

Zoom or unzoom the window.

Parameters:

shouldBeZoomed True if window should be zoomed.

Return values:

S_OK The operation was successful.

ITUNES_E_OBJECTLOCKED Window is not zoomable.

ITUNES_E_OBJECTDELETED This window has been deleted.

E_FAIL An unexpected error occurred.

HRESULT IITWindow::Zoomed ([out, retval] VARIANT_BOOL * isZ

Returns true if the window is zoomed.

Parameters:

isZoomed Returns true if the window is zoomed.

Return values:

S_OK The operation was successful.

E POINTER isZoomed is NULL.

ITUNES_E_OBJECTDELETED This window has been deleted.

E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

IITWindowCollection Interface Reference

List of all members.

Detailed Description

Represents a collection of window objects.

Note that collection indicies are always 1-based.

You can retrieve all the windows using IiTunes::Windows().

Public Member Functions

Properties

HRESULT	Count ([out, retval] long *count) Returns the number of windows in the collection.
HRESULT	Item ([in] long index,[out, retval] IITWindow **iWindow) Returns an IITWindow object corresponding to the given index (1-based).
HRESULT	ItemByName ([in] BSTR name,[out, retval] IITWindow **iWindow) Returns an IITWindow object with the specified name.
HRESULT	_NewEnum ([out, retval] IUnknown **iEnumerator) Returns an IEnumVARIANT object which can enumerate the collection.

Member Function Documentation

HRESULT IITWindowCollection::_NewEnum ([out, retval] IUnknow

Returns an IEnumVARIANT object which can enumerate the collection.

Note:

This property is available in iTunes 4.7 and later (iTunes type library 1.2 and later).

Parameters:

iEnumerator Returns an IEnumVARIANT object which can enumerate the collection.

Return values:

S_OK The operation was successful.

E POINTER iEnumerator is NULL.

E_FAIL An unexpected error occurred.

HRESULT IITWindowCollection::Count ([out, retval] long * count)

Returns the number of windows in the collection.

Parameters:

count Returns the number of windows in the collection.

Return values:

S_OK The operation was successful.

E_POINTER count is NULL.

E_FAIL An unexpected error occurred.

```
HRESULT IITWindowCollection::Item ( [in] long in [out, retval] IITWindow ** iV )
```

Returns an **IITWindow** object corresponding to the given index (1-based).

Parameters:

index Index of the window to retrieve, must be less than or

equal to IITWindowCollection::Count().

iWindow Returns an **IITWindow** object corresponding to the

given index. Will be set to NULL if no window could be

retrieved.

Return values:

S *OK* The operation was successful.

S_FALSE index is too high.

E INVALIDARG index is less than one.

E_POINTER iWindow is NULL.

E_FAIL An unexpected error occurred.

```
HRESULT IITWindowCollection::ItemByName ( [in] BSTR [out, retval] IITWind )
```

Returns an **IITWindow** object with the specified name.

Parameters:

name The name of the window to retrieve.

iWindow Returns an **IITWindow** object with the specified name.

Will be set to NULL if no window could be retrieved.

Return values:

S_*OK* The operation was successful.

 S_FALSE A Window with the specified name could not be

found.

E_INVALIDARG name is empty.

E_POINTER name or iWindow is NULL.

E_FAIL An unexpected error occurred.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

iTunesApp Class Reference

Detailed Description

Represents the iTunes application.

This is the only externally creatable iTunes COM object. All other iTunes interfaces are accessed through this class.

See the **liTunes** interface for more details.

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 Apple Computer, Inc.

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

All | Functions

_|a|b|c|d|e|f|g|h|i|k|l|m|n|o|p|q|r|s|t|u|v|w|y z

Here is a list of all class members with links to the classes they belong to:

-_-

 _NewEnum(): IITWindowCollection, IITVisualCollection, IITTrackCollection, IITSourceCollection, IITPlaylistCollection, IITEQPresetCollection, IITEncoderCollection, IITArtworkCollection

- a -

- AddArtworkFromFile(): IITTrack
- AddFile(): IITUserPlaylist, IITLibraryPlaylist
- AddFiles(): IITUserPlaylist, IITLibraryPlaylist
- AddTrack(): IITUserPlaylist, IITLibraryPlaylist
- AddURL(): IITUserPlaylist, IITLibraryPlaylist
- Album(): IITTrack
- AlbumArtist(): IITFileOrCDTrack
- AlbumRating(): IITURLTrack, IITFileOrCDTrack
- AlbumRatingKind(): IITURLTrack, IITFileOrCDTrack
- AppCommandMessageProcessingEnabled(): IiTunes
- Artist(): IITTrack, IITAudioCDPlaylist
- Artwork(): IITTrack

- b -

BackTrack(): liTunesBand1(): lITEQPreset

- Band10(): IITEQPreset
- Band2(): IITEQPreset
- Band3(): IITEQPreset
- Band4(): IITEQPreset
- Band5(): IITEQPreset
- Band6(): IITEQPreset
- Band7(): IITEQPreset
- Band8(): IITEQPreset
- Band9(): IITEQPreset
- BitRate(): IITTrack
- BookmarkTime(): IITFileOrCDTrack
- Bottom(): IITWindow
- BPM(): IITTrack
- BrowserWindow(): liTunes

- C -

- CanSetShuffle(): IiTunes
- CanSetSongRepeat(): IiTunes
- Capacity(): IITSource
- Category(): IITURLTrack, IITFileOrCDTrack
- CheckVersion(): IiTunes
- Comment(): IITTrack
- Compilation(): IITTrack, IITAudioCDPlaylist
- Composer(): IITTrack, IITAudioCDPlaylist
- ConvertFile(): liTunes
- ConvertFile2(): IiTunes
- ConvertFiles(): liTunes
- ConvertFiles2(): liTunes
- ConvertOperationStatus(): IiTunes
- ConvertTrack(): liTunes
- ConvertTrack2(): IiTunes
- ConvertTracks(): IiTunes
- ConvertTracks2(): IiTunes
- Count(): IITWindowCollection, IITVisualCollection, IITTrackCollection, IITSourceCollection, IITPlaylistCollection, IITEQPresetCollection, IITEncoderCollection,

IITArtworkCollection

- CreateEQPreset(): IiTunes
- CreateFolder(): IITUserPlaylist, IiTunes
- CreateFolderInSource(): IiTunes
- CreatePlaylist(): IITUserPlaylist, IiTunes
- CreatePlaylistInSource(): IiTunes
- CurrentEncoder(): liTunes
- CurrentEQPreset(): IiTunes
- CurrentPlaylist(): liTunes
- CurrentStreamTitle(): IiTunes
- CurrentStreamURL(): liTunes
- CurrentTrack(): liTunes
- CurrentVisual(): IiTunes

- d -

- DateAdded(): IITTrack
- Delete(): IITTrack, IITPlaylist, IITEQPreset, IITArtwork
- Description(): IITURLTrack, IITFileOrCDTrack, IITArtwork
- DiscCount(): IITTrack, IITAudioCDPlaylist
- DiscNumber(): IITTrack, IITAudioCDPlaylist
- DownloadPodcastEpisode(): IITURLTrack
- Duration(): IITTrack, IITPlaylist

- e -

- EjectlPod(): IITIPodSource
- Enabled(): IITTrack
- Encoders(): liTunes
- EpisodeID(): IITFileOrCDTrack
- EpisodeNumber(): IITFileOrCDTrack
- EQ(): IITTrack
- EQEnabled(): liTunes
- EQPresets(): liTunes
- EQWindow(): IiTunes
- ExcludeFromShuffle(): IITFileOrCDTrack

- f -

- FastForward(): liTunes
- Finish(): IITTrack
- ForceToForegroundOnDialog(): liTunes
- Format(): IITEncoder, IITArtwork
- FreeSpace(): IITSource
- FullScreenVisuals(): IiTunes

- g -

- Genre(): IITTrack, IITAudioCDPlaylist
- GetConversionStatus(): IITConvertOperationStatus
- GetITObjectByID(): IiTunes
- GetITObjectIDs(): IITObject
- GetITObjectPersistentIDs(): IiTunes
- GetPlayerButtonsState(): IiTunes
- GotoMusicStoreHomePage(): liTunes
- Grouping(): IITTrack

- h -

• Height(): IITWindow

- j -

- Index(): IITObject
- InProgress(): IITOperationStatus
- IsDownloadedArtwork(): IITArtwork
- Item(): IITWindowCollection, IITVisualCollection, IITTrackCollection, IITSourceCollection, IITPlaylistCollection, IITEQPresetCollection, IITEncoderCollection, IITArtworkCollection
- ItemByName(): IITWindowCollection, IITVisualCollection, IITTrackCollection, IITSourceCollection, IITPlaylistCollection, IITEQPresetCollection, IITEncoderCollection
- ItemByPersistentID(): IITTrackCollection, IITSourceCollection,

IITPlaylistCollection

- ItemByPlayOrder(): IITTrackCollection
- ITObjectPersistentIDHigh(): IiTunes
- ITObjectPersistentIDLow(): IiTunes

- k -

- Kind(): IITWindow, IITTrack, IITSource, IITPlaylist
- KindAsString(): IITTrack

- | -

- Left(): IITWindow
- LibraryPlaylist(): IiTunes
- LibrarySource(): IiTunes
- LibraryXMLPath(): IiTunes
- Location(): IITFileOrCDTrack
- LongDescription(): IITURLTrack, IITFileOrCDTrack
- Lyrics(): IITFileOrCDTrack

- m -

- Maximizable(): IITWindow
- Maximized(): IITWindow
- MaxProgressValue(): IITConvertOperationStatus
- Minimized(): IITWindow
- MiniPlayer(): IITBrowserWindow
- Modifiable(): IITEQPreset
- ModificationDate(): IITTrack
- Mute(): IiTunes

- n -

- Name(): IITWindow, IITVisual, IITObject, IITEQPreset, IITEncoder
- NextTrack(): IiTunes

- OnAboutToPromptUserToQuitEvent(): _liTunesEvents
- OnCOMCallsDisabledEvent(): _liTunesEvents
- OnCOMCallsEnabledEvent(): _liTunesEvents
- OnConvertOperationCompleteEvent() : IITConvertOperationStatusEvents
- OnConvertOperationStatusChangedEvent(): IITConvertOperationStatusEvents
- OnDatabaseChangedEvent(): _liTunesEvents
- OnPlayerPlayEvent(): _liTunesEvents
- OnPlayerPlayingTrackChangedEvent(): _liTunesEvents
- OnPlayerStopEvent(): _liTunesEvents
- OnQuittingEvent(): _liTunesEvents
- OnSoundVolumeChangedEvent(): _liTunesEvents
- OpenURL(): liTunes

- p -

- Parent(): IITUserPlaylist
- PartOfGaplessAlbum(): IITFileOrCDTrack
- Pause(): liTunes
- Play(): IITTrack, IiTunes
- PlayedCount(): IITTrack
- PlayedDate(): IITTrack
- PlayerButtonClicked(): IiTunes
- PlayerPosition(): IiTunes
- PlayerState(): IiTunes
- PlayFile(): IiTunes
- PlayFirstTrack(): IITPlaylist
- Playlist(): IITPlaylistWindow, IITTrack
- PlaylistID(): IITObject
- Playlists(): IITURLTrack, IITFileOrCDTrack, IITSource
- PlayOrderIndex(): IITTrack
- PlayPause(): liTunes
- Podcast(): IITURLTrack, IITFileOrCDTrack
- Preamp(): IITEQPreset

- PreviousTrack(): liTunes
- Print(): IITPlaylist
- ProgressValue(): IITConvertOperationStatus

- q -

• Quit(): IiTunes

- r -

- Rating(): IITTrack
- RatingKind(): IITURLTrack, IITFileOrCDTrack
- ReleaseDate(): IITFileOrCDTrack
- RememberBookmark(): IITFileOrCDTrack
- Rename(): IITEQPreset
- Resizable(): IITWindow
- Resume(): liTunes
- Reveal(): IITURLTrack, IITFileOrCDTrack, IITAudioCDPlaylist, IITUserPlaylist
- Rewind(): IiTunes
- Right(): IITWindow

- S -

- SampleRate(): IITTrack
- SaveArtworkToFile(): IITArtwork
- Search(): IITPlaylist
- SeasonNumber(): IITFileOrCDTrack
- SelectedPlaylist(): IITBrowserWindow
- SelectedTracks(): IITPlaylistWindow, IITBrowserWindow, IiTunes
- SetArtworkFromFile(): IITArtwork
- Shared(): IITUserPlaylist
- Show(): IITFileOrCDTrack
- Shuffle(): IITPlaylist
- Size(): IITTrack, IITPlaylist
- Size64High(): IITFileOrCDTrack
- Size64Low(): IITFileOrCDTrack

- SkippedCount(): IITFileOrCDTrack
- SkippedDate(): IITFileOrCDTrack
- Smart(): IITUserPlaylist
- SoftwareVersion(): IITIPodSource
- SongRepeat(): IITPlaylist
- SortAlbum(): IITFileOrCDTrack
- SortAlbumArtist(): IITFileOrCDTrack
- SortArtist(): IITFileOrCDTrack
- SortComposer() : IITFileOrCDTrack
- SortName(): IITFileOrCDTrack
- SortShow(): IITFileOrCDTrack
- SoundVolume(): IiTunes
- SoundVolumeControlEnabled(): IiTunes
- Source(): IITPlaylist
- SourceID(): IITObject
- Sources(): IiTunes
- SpecialKind(): IITUserPlaylist
- Start(): IITTrack
- Stop(): liTunes
- StopConversion(): IITConvertOperationStatus
- SubscribeToPodcast(): IiTunes

- t -

- Time(): IITTrack, IITPlaylist
- Top(): IITWindow
- TrackCount(): IITTrack
- TrackDatabaseID(): IITObject
- TrackID(): IITObject
- TrackName(): IITConvertOperationStatus
- TrackNumber() : IITTrack
- Tracks(): IITPlaylist, IITOperationStatus

- u -

- Unplayed(): IITFileOrCDTrack
- UpdateInfoFromFile(): IITFileOrCDTrack

- UpdatelPod(): IITIPodSource, IiTunes
- UpdatePodcastFeed(): IITURLTrack, IITFileOrCDTrack
- UpdatePodcastFeeds(): IiTunes
- URL(): IITURLTrack

- V -

- Version(): IiTunes
- VideoKind(): IITFileOrCDTrackVisible(): IITWindow, IITPlaylist
- Visuals(): IiTunes
- VisualsEnabled(): IiTunes
- VisualSize(): IiTunes
- VolumeAdjustment(): IITTrack

- W -

- Width(): IITWindowWindows(): IiTunes
- y -
 - Year(): IITTrack, IITAudioCDPlaylist

- Z -

Zoomable(): IITWindowZoomed(): IITWindow

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

iTunesLib Namespace Reference

Functions

importlib ("stdole2.tlb")

Function Documentation

importlib ("stdole2.tlb")

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List | Class List | File List | Namespace Members | Class Members | File Members | Related Pages

All | Functions | Enumerations | Enumeration values

i | k

Here is a list of all file members with links to the files they belong to:

- j -

- importlib(): iTunesCOMInterface.idl
- ITArtworkFormat : iTunesArtworkCOM.idl
- ITArtworkFormatBMP: iTunesArtworkCOM.idl
- ITArtworkFormatJPEG: iTunesArtworkCOM.idl
- ITArtworkFormatPNG: iTunesArtworkCOM.idl
- ITArtworkFormatUnknown: iTunesArtworkCOM.idl
- ITCOMDisabledReason : iTunesAppCOM.idl
- ITCOMDisabledReasonDialog: iTunesAppCOM.idl
- ITCOMDisabledReasonOther: iTunesAppCOM.idl
- ITCOMDisabledReasonQuitting : iTunesAppCOM.idl
- ITConvertOperationComplete : iTunesAppCOM.idl
- ITConvertOperationStatusChanged : iTunesAppCOM.idl
- ITConvertOperationStatusEvent : iTunesAppCOM.idl
- ITErrors : iTunesCOMInterface.idl
- ITEvent : iTunesAppCOM.idl
- ITEventAboutToPromptUserToQuit: iTunesAppCOM.idl
- ITEventCOMCallsDisabled : iTunesAppCOM.idl
- ITEventCOMCallsEnabled : iTunesAppCOM.idl
- ITEventDatabaseChanged : iTunesAppCOM.idl
- ITEventPlayerPlay : iTunesAppCOM.idl
- ITEventPlayerPlayingTrackChanged : iTunesAppCOM.idl
- ITEventPlayerStop : iTunesAppCOM.idl
- ITEventQuitting: iTunesAppCOM.idl
- ITEventSoundVolumeChanged: iTunesAppCOM.idl
- ITPlayButtonState : iTunesAppCOM.idl

- ITPlayButtonStatePauseDisabled : iTunesAppCOM.idl
- ITPlayButtonStatePauseEnabled : iTunesAppCOM.idl
- ITPlayButtonStatePlayDisabled : iTunesAppCOM.idl
- ITPlayButtonStatePlayEnabled : iTunesAppCOM.idl
- ITPlayButtonStateStopDisabled : iTunesAppCOM.idl
- ITPlayButtonStateStopEnabled : iTunesAppCOM.idl
- ITPlayerButton: iTunesAppCOM.idl
- ITPlayerButtonModifierKey : iTunesAppCOM.idl
- ITPlayerButtonModifierKeyAlt : iTunesAppCOM.idl
- ITPlayerButtonModifierKeyCapsLock : iTunesAppCOM.idl
- ITPlayerButtonModifierKeyControl : iTunesAppCOM.idl
- ITPlayerButtonModifierKeyNone : iTunesAppCOM.idl
- ITPlayerButtonModifierKeyShift : iTunesAppCOM.idl
- ITPlayerButtonNext : iTunesAppCOM.idl
- ITPlayerButtonPlay : iTunesAppCOM.idl
- ITPlayerButtonPrevious : iTunesAppCOM.idl
- ITPlayerState: iTunesAppCOM.idl
- ITPlayerStateFastForward: iTunesAppCOM.idl
- ITPlayerStatePlaying : iTunesAppCOM.idl
- ITPlayerStateRewind : iTunesAppCOM.idl
- ITPlayerStateStopped : iTunesAppCOM.idl
- ITPlaylistKind : iTunesPlaylistCOM.idl
- ITPlaylistKindCD: iTunesPlaylistCOM.idl
- ITPlaylistKindDevice : iTunesPlaylistCOM.idl
- ITPlaylistKindLibrary: iTunesPlaylistCOM.idl
- ITPlaylistKindRadioTuner : iTunesPlaylistCOM.idl
- ITPlaylistKindUnknown: iTunesPlaylistCOM.idl
- ITPlaylistKindUser : iTunesPlaylistCOM.idl
- ITPlaylistPrintKind : iTunesPlaylistCOM.idl
- ITPlaylistPrintKindAlbumlist : iTunesPlaylistCOM.idl
- ITPlaylistPrintKindInsert : iTunesPlaylistCOM.idl
- ITPlaylistPrintKindPlaylist: iTunesPlaylistCOM.idl
- ITPlaylistRepeatMode : iTunesPlaylistCOM.idl
- ITPlaylistRepeatModeAll: iTunesPlaylistCOM.idl
- ITPlaylistRepeatModeOff : iTunesPlaylistCOM.idl
- ITPlaylistRepeatModeOne : iTunesPlaylistCOM.idl
- ITPlaylistSearchField: iTunesPlaylistCOM.idl

- ITPlaylistSearchFieldAlbums : iTunesPlaylistCOM.idl
- ITPlaylistSearchFieldAll : iTunesPlaylistCOM.idl
- ITPlaylistSearchFieldArtists: iTunesPlaylistCOM.idl
- ITPlaylistSearchFieldComposers : iTunesPlaylistCOM.idl
- ITPlaylistSearchFieldSongNames : iTunesPlaylistCOM.idl
- ITPlaylistSearchFieldVisible : iTunesPlaylistCOM.idl
- ITRatingKind : iTunesTrackCOM.idl
- ITRatingKindComputed : iTunesTrackCOM.idl
- ITRatingKindUser : iTunesTrackCOM.idl
- ITSourceKind : iTunesSourceCOM.idl
- ITSourceKindAudioCD: iTunesSourceCOM.idl
- ITSourceKindDevice: iTunesSourceCOM.idl
- ITSourceKindlPod: iTunesSourceCOM.idl
- ITSourceKindLibrary : iTunesSourceCOM.idl
- ITSourceKindMP3CD: iTunesSourceCOM.idl
- ITSourceKindRadioTuner: iTunesSourceCOM.idl
- ITSourceKindSharedLibrary : iTunesSourceCOM.idl
- ITSourceKindUnknown: iTunesSourceCOM.idl
- ITTrackKind: iTunesTrackCOM.idl
- ITTrackKindCD: iTunesTrackCOM.idl
- ITTrackKindDevice : iTunesTrackCOM.idl
- ITTrackKindFile: iTunesTrackCOM.idl
- ITTrackKindSharedLibrary: iTunesTrackCOM.idl
- ITTrackKindUnknown: iTunesTrackCOM.idl
- ITTrackKindURL: iTunesTrackCOM.idl
- ITUNES_E_CONVERSIONINPROGRESS:
 iTunesCOMInterface.idl
- ITUNES_E_MUSICSTOREDISABLED: iTunesCOMInterface.idl
- ITUNES_E_OBJECTDELETED : iTunesCOMInterface.idl
- ITUNES E OBJECTEXISTS: iTunesCOMInterface.idl
- ITUNES_E_OBJECTLOCKED: iTunesCOMInterface.idl
- ITUNES E PODCASTSDISABLED: iTunesCOMInterface.idl
- ITUNES_E_USERCANCEL: iTunesCOMInterface.idl
- ITUserPlaylistSpecialKind : iTunesPlaylistCOM.idl
- ITUserPlaylistSpecialKindAudiobooks : iTunesPlaylistCOM.idl
- ITUserPlaylistSpecialKindFolder : iTunesPlaylistCOM.idl
- ITUserPlaylistSpecialKindMovies : iTunesPlaylistCOM.idl

- ITUserPlaylistSpecialKindMusic : iTunesPlaylistCOM.idl
- ITUserPlaylistSpecialKindNone : iTunesPlaylistCOM.idl
- ITUserPlaylistSpecialKindPartyShuffle : iTunesPlaylistCOM.idl
- ITUserPlaylistSpecialKindPodcasts : iTunesPlaylistCOM.idl
- ITUserPlaylistSpecialKindPurchasedMusic: iTunesPlaylistCOM.idl
- ITUserPlaylistSpecialKindTVShows: iTunesPlaylistCOM.idl
- ITUserPlaylistSpecialKindVideos : iTunesPlaylistCOM.idl
- ITVersion: iTunesCOMInterface.idl
- ITVideoKind: iTunesTrackCOM.idl
- ITVideoKindMovie: iTunesTrackCOM.idl
- ITVideoKindMusicVideo: iTunesTrackCOM.idl
- ITVideoKindNone: iTunesTrackCOM.idl
- ITVideoKindTVShow: iTunesTrackCOM.idl
- ITVisualSize: iTunesAppCOM.idl
- ITVisualSizeLarge : iTunesAppCOM.idl
- ITVisualSizeMedium : iTunesAppCOM.idl
- ITVisualSizeSmall : iTunesAppCOM.idl
- ITWindowKind: iTunesWindowCOM.idl
- ITWindowKindArtwork : iTunesWindowCOM.idl
- ITWindowKindBrowser : iTunesWindowCOM.idl
- ITWindowKindEQ : iTunesWindowCOM.idl
- ITWindowKindNowPlaying : iTunesWindowCOM.idl
- ITWindowKindPlaylist : iTunesWindowCOM.idl
- ITWindowKindUnknown: iTunesWindowCOM.idl

- k -

- kITTypeLibrary_MajorVersion: iTunesCOMInterface.idl
- kITTypeLibrary_MinorVersion : iTunesCOMInterface.idl

Main Page | Namespace List | Class Hierarchy | Alphabetical List | Class List | File List | Namespace Members | Class Members | File Members | Related Pages

All | Functions

Here is a list of all namespace members with links to the namespace documentation for each member:

• importlib(): iTunesLib

Generated on Fri Mar 13 12:50:54 2009 for iTunes 8.1.0.52 ©2004-2007 <u>Apple Computer, Inc.</u>

Main Page | Namespace List | Class Hierarchy | Alphabetical List |
Class List | File List | Namespace Members | Class Members |
File Members | Related Pages

iTunes COM Interface Changes

The iTunes COM interface was first released with iTunes 4.5, which included version 1.0 of the iTunes type library.

Use **liTunes::CheckVersion()** with the major and minor version of the type library to determine if the version of iTunes currently installed supports the interface you need.

Version 1.12 Changes

iTunes 8.1 included version 1.12 of the iTunes type library.

- The IITFileOrCDTrack::Location() property can now be set (it was previously read-only).
- Added IITFileOrCDTrack::ReleaseDate() read-only property.

Version 1.11 Changes

iTunes 7.7 included version 1.11 of the iTunes type library.

liTunes

- Added IiTunes::ITObjectPersistentIDHigh() and IiTunes::ITObjectPersistentIDLow() properties.
- Added IiTunes::GetITObjectPersistentIDs() method.

IITSourceCollection

• Added IITSourceCollection::ItemByPersistentID() property.

IITPlaylistCollection

Added IITPlaylistCollection::ItemByPersistentID() property.

IITTrackCollection

• Added IITTrackCollection::ItemByPersistentID() property.

IITArtwork

• Added IITArtwork::Description() property.

IITFileOrCDTrack

• Added IITFileOrCDTrack::Playlists() property.

IITURLTrack

• Added IITURLTrack::Playlists() property.

Version 1.10 Changes

iTunes 7.4 included version 1.10 of the iTunes type library.

IITUserPlaylist

• Added IITUserPlaylist::Reveal() method.

IITAudioCDPlaylist

• Added IITAudioCDPlaylist::Reveal() method.

IITFileOrCDTrack

- Added IITFileOrCDTrack::Reveal() method.
- Added IITFileOrCDTrack::AlbumRating(), IITFileOrCDTrack::AlbumRatingKind(), and IITFileOrCDTrack::RatingKind() properties.

IITURLTrack

- Added IITURLTrack::Reveal() method.
- Added IITURLTrack::AlbumRating(), IITURLTrack::AlbumRatingKind(), and IITURLTrack::RatingKind() properties.

Version 1.9 Changes

iTunes 7.1 included version 1.9 of the iTunes type library.

IITFileOrCDTrack

- Added IITFileOrCDTrack::Unplayed() property.
- Added IITFileOrCDTrack::Size64High() and IITFileOrCDTrack::Size64Low() properties.
- Added IITFileOrCDTrack::SortAlbum(),

IITFileOrCDTrack::SortAlbumArtist(),

IITFileOrCDTrack::SortArtist(),

IITFileOrCDTrack::SortComposer(),

IITFileOrCDTrack::SortName(), and

IITFileOrCDTrack::SortShow() properties.

Version 1.8 Changes

iTunes 7.0 included version 1.8 of the iTunes type library.

liTunes

Added IiTunes::LibraryXMLPath() property.

IITArtwork

Added IITArtwork::IsDownloadedArtwork() property.

IITFileOrCDTrack

- Added IITFileOrCDTrack::AlbumArtist() property.
- Added IITFileOrCDTrack::PartOfGaplessAlbum() property.
- Added IITFileOrCDTrack::SkippedCount() property.
- Added IITFileOrCDTrack::SkippedDate() property.
- Added IITFileOrCDTrack::VideoKind() property and ITVideoKind enumeration.
- Added IITFileOrCDTrack::Show() property.
- Added IITFileOrCDTrack::SeasonNumber() property.
- Added IITFileOrCDTrack::EpisodeID() property.
- Added IITFileOrCDTrack::EpisodeNumber() property.

IITUserPlaylist

 Added ITUserPlaylistSpecialKindMusic, ITUserPlaylistSpecialKindMovies, ITUserPlaylistSpecialKindTVShows, and ITUserPlaylistSpecialKindAudiobooks to ITUserPlaylistSpecialKind enumeration.

Version 1.7 Changes

iTunes 6.0.2 included version 1.7 of the iTunes type library.

liTunes

- Added IiTunes::CreateFolder method.
- Added IiTunes::CreateFolderInSource method.
- Added IiTunes::SoundVolumeControlEnabled() property.

IITUserPlaylist

- Added IITUserPlaylist::CreatePlaylist() method.
- Added IITUserPlaylist::CreateFolder() method.
- The IITUserPlaylist::Parent() property can now be written.

IITFileOrCDTrack

• Added IITFileOrCDTrack::BookmarkTime() property.

Version 1.6 Changes

iTunes 6.0 included version 1.6 of the iTunes type library.

IITUserPlaylist

• Added ITUserPlaylistSpecialKindVideos to ITUserPlaylistSpecialKind enumeration.

Version 1.5 Changes

iTunes 5.0 included version 1.5 of the iTunes type library.

iTunes Errors

 Added ITUNES_E_PODCASTSDISABLED, which can be returned by IiTunes::SubscribeToPodcast().

liTunes

Added IiTunes::UpdatePodcastFeeds() method.

IITFileOrCDTrack

- Added IITFileOrCDTrack::UpdatePodcastFeed() method.
- Added IITFileOrCDTrack::ExcludeFromShuffle() property.
- Added IITFileOrCDTrack::Lyrics() property.
- Added IITFileOrCDTrack::RememberBookmark() property.
- Added IITFileOrCDTrack::Category() property.
- Added IITFileOrCDTrack::Description() property.
- Added IITFileOrCDTrack::LongDescription() property.

IITURLTrack

- Added IITURLTrack::DownloadPodcastEpisode() method.
- Added IITURLTrack::UpdatePodcastFeed() method.
- Added IITURLTrack::Category() property.
- Added IITURLTrack::Description() property.
- Added IITURLTrack::LongDescription() property.

IITUserPlaylist

- Added IITUserPlaylist::Parent() property.
- Added ITUserPlaylistSpecialKindFolder to ITUserPlaylistSpecialKind enumeration.

Version 1.4 Changes

iTunes 4.9 included version 1.4 of the iTunes type library.

IiTunes

• Added IiTunes::SubscribeToPodcast() method.

IITUserPlaylist

• Added IITUserPlaylist::SpecialKind() property and ITUserPlaylistSpecialKind enumeration.

IITFileOrCDTrack

Added IITFileOrCDTrack::Podcast() property.

IITURLTrack

• Added IITURLTrack::Podcast() property.

Version 1.3 Changes

iTunes 4.8 and 4.8.1 included version 1.3 of the iTunes type library.

liTunes

• Added IiTunes::ConvertOperationStatus() property.

IITIPodSource

• Added IITIPodSource::SoftwareVersion() property.

_IITConvertOperationStatusEvents

Added

_IITConvertOperationStatusEvents::OnConvertOperationComple outgoing event.

Version 1.2 Changes

iTunes 4.7 and 4.7.1 included version 1.2 of the iTunes type library.

iTunes Errors

 Added ITUNES_E_OBJECTEXISTS, which can be returned by the new methods IiTunes::CreateEQPreset() and IITEQPreset::Rename().

_liTunesEvents

- Added _liTunesEvents::OnAboutToPromptUserToQuitEvent() outgoing event.
- Added _liTunesEvents::OnSoundVolumeChangedEvent() outgoing event.

liTunes

- Added IiTunes::CreateEQPreset() method.
- Added IiTunes::CreatePlaylistInSource() method.
- Added IiTunes::GetPlayerButtonsState() method (and ITPlayButtonState enumeration).
- Added IiTunes::PlayerButtonClicked() method (and ITPlayerButton and ITPlayerButtonModifierKey enumerations).
- Added IiTunes::CanSetShuffle() property.
- Added IiTunes::CanSetSongRepeat() property.
- Changed the IiTunes::BackTrack(), IiTunes::PreviousTrack(), and IiTunes::NextTrack() methods to suppress the authorization dialog if invoking the method results in an attempt to start playing a purchased track that has not been authorized for playback on the current machine. The unauthorized track is skipped in this case.

IITEQPreset

Added IITEQPreset::Delete() and IITEQPreset::Rename()

methods.

The IITEQPreset::Modifiable property always returns true for all EQ presets. Previously, only the Manual preset was modifiable. As a result of this change, setting the Preamp or Band properties can no longer return ITUNES_E_OBJECTLOCKED.

IITFileOrCDTrack

• Added IITFileOrCDTrack::UpdateInfoFromFile() method.

IITIPodSource

• Added IITIPodSource::EjectIPod() method.

IEnumVARIANT

 Added <u>IEnumVARIANT</u> support to iTunes collection objects to allow clients to use enumeration semantics to iterate collections. For example, you can now use *foreach* in C#:

• You can also use For Each in VBScript:

```
Set iTunesApp = WScript.CreateObject("iTunes.Application")

Dim aPlaylist

For Each aPlaylist In iTunesApp.LibrarySource.Playlists

WScript.Echo(aPlaylist.Name)

Next
```

Specifically, the following properties were added:

- IITArtworkCollection::_NewEnum
- IITEncoderCollection:: NewEnum
- IITEQPresetCollection::_NewEnum
- IITPlaylistCollection::_NewEnum
- IITSourceCollection:: NewEnum
- IITTrackCollection:: NewEnum
- IITVisualCollection:: NewEnum
- IITWindowCollection::_NewEnum

Version 1.1 Changes

iTunes 4.6 included version 1.1 of the iTunes type library.

liTunes

- Added IiTunes::AppCommandMessageProcessingEnabled property.
- Added IiTunes::ForceToForegroundOnDialog property.
- Added IiTunes::ConvertFile2(), IiTunes::ConvertFiles2(), IiTunes::ConvertTrack2(), and IiTunes::ConvertTracks2() methods which return an IITConvertOperationStatus. These methods supercede IiTunes::ConvertFile(), IiTunes::ConvertFiles(), IiTunes::ConvertTrack(), and IiTunes::ConvertTracks(), although the old methods are still available for backwards compatibility.

IiTunesEvents

- Added ITCOMDisabledReason enumeration.
- Added _liTunesEvents::OnCOMCallsDisabledEvent() and _liTunesEvents::OnCOMCallsEnabledEvent() outgoing events.
- Added _liTunesEvents::OnQuittingEvent() outgoing event.

IITConvertOperationStatus

 Added new interface IITConvertOperationStatus, which is returned by IiTunes::ConvertFile2(), IiTunes::ConvertFiles2(), IiTunes::ConvertTrack2(), and IiTunes::ConvertTracks2().

_IITConvertOperationStatusEvents

 Added new _IITConvertOperationStatusEvents interface, which is supported by the IITConvertOperationStatus interface. ©2004-2007 Apple Computer, Inc.

iTunes COM Interface Namespace List

Here is a list of all namespaces with brief descriptions:

iTunesLib

iTunes COM Interface Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

- IITConvertOperationStatusEvents
- liTunesEvents
- IITArtwork
- IITArtworkCollection
- IITEncoder
- IITEncoderCollection
- IITEQPreset
- IITEQPresetCollection
- IITObject
 - IITPlaylist
 - IITAudioCDPlaylist
 - IITLibraryPlaylist
 - IITUserPlaylist
 - IITSource
 - IITIPodSource
 - IITTrack
 - IITFileOrCDTrack
 - IITURLTrack
- IITOperationStatus
 - IITConvertOperationStatus
- IITPlaylistCollection
- IITSourceCollection
- IITTrackCollection
- liTunes
- IITVisual
- IITVisualCollection
- IITWindow
 - IITBrowserWindow
 - IITPlaylistWindow

- IITWindowCollection
- iTunesApp

iTunes COM Interface Class Index

_ [1





IITArtwork
IITArtworkCollection
IITAudioCDPlaylist
IITBrowserWindow
IITConvertOperationStatus
IITEncoder
IITEncoderCollection
IITEQPreset
IITEQPresetCollection
IITFileOrCDTrack
IITIPodSource
IITLibraryPlaylist
IITObject

IITOperationStatus

IITPlaylist IITPlaylistCollection

IITPlaylistWindow

IITSource

IITSourceCollection

IITTrack

IITTrackCollection

IiTunes

IITURLTrack

IITUserPlaylist

IITVisual

IITVisualCollection

IITWindow

IITWindowCollection

iTunesApp

_ [1

iTunes COM Interface Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

_IITConvertOperationStatusEvents	Defines the outgoing event interface of the IITConvertOperationStatus interface
_liTunesEvents	Defines the outgoing event interface of the liTunes interface
IITArtwork	Defines a single piece of artwork
IITArtworkCollection	Represents a collection of iTunes artwork objects
IITAudioCDPlaylist	Represents an audio CD playlist
IITBrowserWindow	Represents the main browser window
IITConvertOperationStatus	Represents the status of an asynchronous convert operation
IITEncoder	Represents an iTunes encoder
IITEncoderCollection	Represents a collection of encoder objects
IITEQPreset	Represents an equalizer preset
IITEQPresetCollection	Represents a collection of EQ preset objects
IITFileOrCDTrack	

	Represents a file or CD track
IITIPodSource	Represents an iPod source
IITLibraryPlaylist	Represents a library playlist
IITObject	Defines a source, playlist or track
IITOperationStatus	Represents the status of an asynchronous add or convert operation
IITPlaylist	Represents a playlist
IITPlaylistCollection	Represents a collection of playlist objects
IITPlaylistWindow	Represents a playlist window
IITSource	Represents an entry in the Source list (music library, CD, device, etc.)
IITSourceCollection	Represents a collection of source objects
IITTrack	Represents a track
IITTrackCollection	Represents a collection of track objects
liTunes	Defines the top-level iTunes application object
IITURLTrack	Represents a URL track
IITUserPlaylist	Represents a user-defined playlist
IITVisual	Represents an iTunes visual plug-in
IITVisualCollection	Represents a collection of visual plug-in objects
IITWindow	Represents an iTunes window
	Represents a collection of

IITWindowCollection	window objects
iTunesApp	Represents the iTunes application

iTunes COM Interface File List

Here is a list of all files with brief descriptions:

iTunesAppCOM.idl	
iTunesArtworkCOM.idl	
iTunesCOMInterface.idl	
iTunesEncoderCOM.idl	
iTunesEQPresetCOM.idl	
iTunesObjectCOM.idl	
iTunesPlaylistCOM.idl	
iTunesSourceCOM.idl	
iTunesTrackCOM.idl	
iTunesVisualCOM.idl	
iTunesWindowCOM.idl	

iTunes COM Interface Related Pages

Here is a list of all related documentation pages:

• iTunes COM Interface Changes

_IITConvertOperationStatusEvents Member List

This is the complete list of members for __IITConvertOperationStatusEvents, including all inherited members.

OnConvertOperationCompleteEvent()	_IITConvertOperation\$
OnConvertOperationStatusChangedEvent([in]	
BSTR trackName,[in] long progressValue,[in]	_IITConvertOperationS
long maxProgressValue)	

_liTunesEvents Member List

This is the complete list of members for _liTunesEvents, including all inherited members.

OnAboutToPromptUserToQuitEvent()	_liTunesEvents
OnCOMCallsDisabledEvent([in] ITCOMDisabledReason reason)	_liTunesEvents
OnCOMCallsEnabledEvent()	_liTunesEvents
OnDatabaseChangedEvent([in] VARIANT deletedObjectIDs,[in] VARIANT changedObjectIDs)	_liTunesEvents
OnPlayerPlayEvent([in] VARIANT iTrack)	_liTunesEvents
OnPlayerPlayingTrackChangedEvent([in] VARIANT iTrack)	_liTunesEvents
OnPlayerStopEvent([in] VARIANT iTrack)	_liTunesEvents
OnQuittingEvent()	_liTunesEvents
OnSoundVolumeChangedEvent([in] long newVolume)	_liTunesEvents

IITArtwork Member List

This is the complete list of members for **IITArtwork**, including all inherited members.

Delete()	IITArtwork
Description([out, retval] BSTR *description)	IITArtwork
Description([in] BSTR description)	IITArtwork
Format([out, retval] ITArtworkFormat *format)	IITArtwork
IsDownloadedArtwork([out, retval] VARIANT_BOOL *isDownloadedArtwork)	IITArtwork
SaveArtworkToFile([in] BSTR filePath)	IITArtwork
SetArtworkFromFile([in] BSTR filePath)	IITArtwork

IITArtworkCollection Member List

This is the complete list of members for **IITArtworkCollection**, including all inherited members.

_ NewEnum ([out, retval] IUnknown **iEnumerator)	IITArtworkCollection
Count([out, retval] long *count)	IITArtworkCollection
Item([in] long index,[out, retval] IITArtwork **iArtwork)	IITArtworkCollection

IITAudioCDPlaylist Member List

This is the complete list of members for **IITAudioCDPlaylist**, including all inherited members.

Artist([out, retval] BSTR *artist)	IITAudioCDPlaylist
Compilation([out, retval] VARIANT_BOOL *isCompiliation)	IITAudioCDPlaylist
Composer([out, retval] BSTR *composer)	IITAudioCDPlaylist
Delete()	IITPlaylist
DiscCount([out, retval] long *discCount)	IITAudioCDPlaylist
DiscNumber([out, retval] long *discNumber)	IITAudioCDPlaylist
Duration([out, retval] long *duration)	IITPlaylist
Genre([out, retval] BSTR *genre)	IITAudioCDPlaylist
GetITObjectIDs([out] long *sourceID,[out] long *playlistID,[out] long *trackID,[out] long *databaseID)	IITObject
Index([out, retval] long *index)	IITObject
Kind([out, retval] ITPlaylistKind *kind)	IITPlaylist
Name([out, retval] BSTR *name)	IITObject
Name([in] BSTR name)	IITObject
PlayFirstTrack()	IITPlaylist
PlaylistID([out, retval] long *playlistID)	IITObject
<pre>Print([in] VARIANT_BOOL showPrintDialog,[in] ITPlaylistPrintKind printKind,[in] BSTR theme)</pre>	IITPlaylist
Reveal()	IITAudioCDPlaylist
Search([in] BSTR searchText,[in] ITPlaylistSearchField searchFields,[out, retval] IITTrackCollection **iTrackCollection)	IITPlaylist
Shuffle([out, retval] VARIANT_BOOL *isShuffle)	IITPlaylist
Shuffle([in] VARIANT_BOOL shouldShuffle)	IITPlaylist

Size([out, retval] double *size)	IITPlaylist
SongRepeat([out, retval] ITPlaylistRepeatMode *repeatMode)	IITPlaylist
SongRepeat([in] ITPlaylistRepeatMode repeatMode)	IITPlaylist
Source([out, retval] IITSource **iSource)	IITPlaylist
SourceID([out, retval] long *sourceID)	IITObject
Time([out, retval] BSTR *time)	IITPlaylist
TrackDatabaseID([out, retval] long *databaseID)	IITObject
TrackID([out, retval] long *trackID)	IITObject
Tracks([out, retval] IITTrackCollection **iTrackCollection)	IITPlaylist
Visible([out, retval] VARIANT_BOOL *isVisible)	IITPlaylist
Year([out, retval] long *year)	IITAudioCDPlaylist

IITBrowserWindow Member List

This is the complete list of members for **IITBrowserWindow**, including all inherited members.

Bottom([out, retval] long *bottom)	IITWindow
Bottom([in] long bottom)	IITWindow
Height([out, retval] long *height)	IITWindow
Height([in] long height)	IITWindow
Kind([out, retval] ITWindowKind *kind)	IITWindow
Left([out, retval] long *left)	IITWindow
Left([in] long left)	IITWindow
Maximizable([out, retval] VARIANT_BOOL *isMaximizable)	IITWindow
Maximized([out, retval] VARIANT_BOOL *isMaximized)	IITWindow
Maximized([in] VARIANT_BOOL shouldBeMaximized)	IITWindow
Minimized([out, retval] VARIANT_BOOL *isMinimized)	IITWindow
Minimized([in] VARIANT_BOOL shouldBeMinimized)	IITWindow
MiniPlayer([out, retval] VARIANT_BOOL *isMiniPlayer)	IITBrowserWindow
MiniPlayer([in] VARIANT_BOOL shouldBeMiniPlayer)	IITBrowserWindow
Name([out, retval] BSTR *name)	IITWindow
Resizable([out, retval] VARIANT_BOOL *isResizable)	IITWindow
Right([out, retval] long *right)	IITWindow
Right([in] long right)	IITWindow
	IITBrowserWindow

SelectedPlaylist([out, retval] IITPlaylist **iPlaylist)	
SelectedPlaylist([in] VARIANT *iPlaylist)	IITBrowserWindow
SelectedTracks([out, retval] IITTrackCollection **iTrackCollection)	IITBrowserWindow
Top([out, retval] long *top)	IITWindow
Top([in] long top)	IITWindow
Visible([out, retval] VARIANT_BOOL *isVisible)	IITWindow
Visible([in] VARIANT_BOOL shouldBeVisible)	IITWindow
Width([out, retval] long *width)	IITWindow
Width([in] long width)	IITWindow
Zoomable ([out, retval] VARIANT_BOOL *isZoomable)	IITWindow
<pre>Zoomed([out, retval] VARIANT_BOOL *isZoomed)</pre>	IITWindow
Zoomed ([in] VARIANT_BOOL shouldBeZoomed)	IITWindow

IITConvertOperationStatus Member List

This is the complete list of members for **IITConvertOperationStatus**, including all inherited members.

GetConversionStatus ([out] BSTR *trackName,[out] long *progressValue,[out] long *maxProgressValue)	IITConvertOperationStatus
<pre>InProgress([out, retval] VARIANT_BOOL *isInProgress)</pre>	IITOperationStatus
MaxProgressValue([out, retval] long *maxProgressValue)	IITConvertOperationStatus
ProgressValue([out, retval] long *progressValue)	IITConvertOperationStatus
StopConversion()	IITConvertOperationStatus
TrackName ([out, retval] BSTR *trackName)	IITConvertOperationStatus
Tracks ([out, retval] IITTrackCollection **iTrackCollection)	IITOperationStatus

IITEncoder Member List

This is the complete list of members for **IITEncoder**, including all inherited members.

Format([out, retval] BSTR *format) IITEncoder
Name([out, retval] BSTR *name) IITEncoder

IITEncoderCollection Member List

This is the complete list of members for **IITEncoderCollection**, including all inherited members.

_ NewEnum ([out, retval] IUnknown **iEnumerator)	IITEncoderCollection
Count([out, retval] long *count)	IITEncoderCollection
<pre>Item([in] long index,[out, retval] IITEncoder **iEncoder)</pre>	IITEncoderCollection
ItemByName([in] BSTR name,[out, retval] IITEncoder **iEncoder)	IITEncoderCollection

IITEQPreset Member List

This is the complete list of members for **IITEQPreset**, including all inherited members.

Band1([out, retval] double *level)	IITEQPreset
Band1([in] double level)	IITEQPreset
Band10([out, retval] double *level)	IITEQPreset
Band10([in] double level)	IITEQPreset
Band2([out, retval] double *level)	IITEQPreset
Band2([in] double level)	IITEQPreset
Band3([out, retval] double *level)	IITEQPreset
Band3([in] double level)	IITEQPreset
Band4([out, retval] double *level)	IITEQPreset
Band4([in] double level)	IITEQPreset
Band5([out, retval] double *level)	IITEQPreset
Band5([in] double level)	IITEQPreset
Band6([out, retval] double *level)	IITEQPreset
Band6([in] double level)	IITEQPreset
Band7([out, retval] double *level)	IITEQPreset
Band7([in] double level)	IITEQPreset
Band8([out, retval] double *level)	IITEQPreset
Band8([in] double level)	IITEQPreset
Band9([out, retval] double *level)	IITEQPreset
Band9([in] double level)	IITEQPreset
Delete([in] VARIANT_BOOL updateAllTracks)	IITEQPreset
Modifiable([out, retval] VARIANT_BOOL *isModifiable)	IITEQPreset
Name([out, retval] BSTR *name)	IITEQPreset
Preamp([out, retval] double *level)	IITEQPreset
Preamp([in] double level)	IITEQPreset
Rename([in] BSTR newName,[in] VARIANT_BOOL	

IITEQPresetCollection Member List

This is the complete list of members for **IITEQPresetCollection**, including all inherited members.

_ NewEnum ([out, retval] IUnknown **iEnumerator)	IITEQPresetCollection
Count([out, retval] long *count)	IITEQPresetCollection
<pre>Item([in] long index,[out, retval] IITEQPreset **iEQPreset)</pre>	IITEQPresetCollection
ItemByName([in] BSTR name,[out, retval] IITEQPreset **iEQPreset)	IITEQPresetCollection

IITFileOrCDTrack Member List

This is the complete list of members for **IITFileOrCDTrack**, including all inherited members.

AddArtworkFromFile ([in] BSTR filePath,[out, retval] IITArtwork **iArtwork)	IITTrack
Album([out, retval] BSTR *album)	IITTrack
Album([in] BSTR album)	IITTrack
AlbumArtist([out, retval] BSTR *albumArtist)	IITFileOrCDTrack
AlbumArtist([in] BSTR albumArtist)	IITFileOrCDTrack
AlbumRating([out, retval] long *rating)	IITFileOrCDTrack
AlbumRating([in] long rating)	IITFileOrCDTrack
AlbumRatingKind([out, retval] ITRatingKind *ratingKind)	IITFileOrCDTrack
Artist([out, retval] BSTR *artist)	IITTrack
Artist([in] BSTR artist)	IITTrack
<pre>Artwork([out, retval] IITArtworkCollection **iArtworkCollection)</pre>	IITTrack
BitRate([out, retval] long *bitrate)	IITTrack
BookmarkTime([out, retval] long *bookmarkTime)	IITFileOrCDTrack
BookmarkTime([in] long bookmarkTime)	IITFileOrCDTrack
BPM([out, retval] long *beatsPerMinute)	IITTrack
BPM([in] long beatsPerMinute)	IITTrack
Category([out, retval] BSTR *category)	IITFileOrCDTrack
Category([in] BSTR category)	IITFileOrCDTrack
Comment([out, retval] BSTR *comment)	IITTrack
Comment([in] BSTR comment)	IITTrack
Compilation([out, retval] VARIANT_BOOL *isCompilation)	IITTrack
Compilation([in] VARIANT_BOOL	

shouldBeCompilation)	IITTrack
Composer([out, retval] BSTR *composer)	IITTrack
Composer([in] BSTR composer)	IITTrack
DateAdded([out, retval] DATE *dateAdded)	IITTrack
Delete()	IITTrack
Description([out, retval] BSTR *description)	IITFileOrCDTrack
Description([in] BSTR description)	IITFileOrCDTrack
DiscCount([out, retval] long *discCount)	IITTrack
DiscCount([in] long discCount)	IITTrack
DiscNumber([out, retval] long *discNumber)	IITTrack
DiscNumber([in] long discNumber)	IITTrack
Duration([out, retval] long *duration)	IITTrack
Enabled([out, retval] VARIANT_BOOL *isEnabled)	IITTrack
Enabled([in] VARIANT_BOOL shouldBeEnabled)	IITTrack
EpisodeID([out, retval] BSTR *episodeID)	IITFileOrCDTrack
EpisodeID([in] BSTR episodeID)	IITFileOrCDTrack
EpisodeNumber([out, retval] long *episodeNumber)	IITFileOrCDTrack
EpisodeNumber([in] long episodeNumber)	IITFileOrCDTrack
EQ([out, retval] BSTR *eq)	IITTrack
EQ([in] BSTR eq)	IITTrack
ExcludeFromShuffle ([out, retval] VARIANT_BOOL *excludeFromShuffle)	IITFileOrCDTrack
ExcludeFromShuffle([in] VARIANT_BOOL shouldExcludeFromShuffle)	IITFileOrCDTrack
Finish([in] long finish)	IITTrack
Finish([out, retval] long *finish)	IITTrack
Genre([out, retval] BSTR *genre)	IITTrack
Genre([in] BSTR genre)	IITTrack
GetITObjectIDs([out] long *sourceID,[out] long *playlistID,[out] long *trackID,[out] long *databaseID)	IITObject
Grouping([out, retval] BSTR *grouping)	IITTrack

Grouping([in] BSTR grouping)	IITTrack
Index([out, retval] long *index)	IITObject
Kind([out, retval] ITTrackKind *kind)	IITTrack
KindAsString([out, retval] BSTR *kind)	IITTrack
Location([out, retval] BSTR *location)	IITFileOrCDTrack
Location([in] BSTR location)	IITFileOrCDTrack
LongDescription([out, retval] BSTR *longDescription)	IITFileOrCDTrack
LongDescription([in] BSTR longDescription)	IITFileOrCDTrack
Lyrics([out, retval] BSTR *lyrics)	IITFileOrCDTrack
Lyrics([in] BSTR lyrics)	IITFileOrCDTrack
ModificationDate([out, retval] DATE *dateModified)	IITTrack
Name([out, retval] BSTR *name)	IITObject
Name([in] BSTR name)	IITObject
<pre>PartOfGaplessAlbum([out, retval] VARIANT_BOOL *partOfGaplessAlbum)</pre>	IITFileOrCDTrack
PartOfGaplessAlbum([in] VARIANT_BOOL shouldBePartOfGaplessAlbum)	IITFileOrCDTrack
Play()	IITTrack
PlayedCount([out, retval] long *playedCount)	IITTrack
PlayedCount([in] long playedCount)	IITTrack
PlayedDate([out, retval] DATE *playedDate)	IITTrack
PlayedDate([in] DATE playedDate)	IITTrack
Playlist([out, retval] IITPlaylist **iPlaylist)	IITTrack
PlaylistID([out, retval] long *playlistID)	IITObject
<pre>Playlists([out, retval] IITPlaylistCollection **iPlaylistCollection)</pre>	IITFileOrCDTrack
PlayOrderIndex([out, retval] long *index)	IITTrack
Podcast([out, retval] VARIANT_BOOL *isPodcast)	IITFileOrCDTrack
Rating([out, retval] long *rating)	IITTrack
Rating([in] long rating)	IITTrack
RatingKind([out, retval] ITRatingKind *ratingKind)	IITFileOrCDTrack

RememberBookmark([out, retval] VARIANT_BOOL *rememberBookmark) RememberBookmark([in] VARIANT_BOOL shouldRememberBookmark) Reveal() SampleRate([out, retval] long *sampleRate) SeasonNumber([out, retval] long *seasonNumber) IITFileOrCDTransparent IITTrack IITFileOrCDTransparent
shouldRememberBookmark) Reveal() SampleRate([out, retval] long *sampleRate) IITFileOrCDTra
SampleRate([out, retval] long *sampleRate) IITTrack
SeasonNumber([out, retval] long *seasonNumber) IITFileOrCDTra
\L ' \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
SeasonNumber([in] long seasonNumber) IITFileOrCDTra
Show([out, retval] BSTR *showName) IITFileOrCDTra
Show([in] BSTR showName) IITFileOrCDTra
Size([out, retval] long *size)
Size64High([out, retval] long *sizeHigh) IITFileOrCDTra
Size64Low([out, retval] long *sizeLow) IITFileOrCDTra
SkippedCount([out, retval] long *skippedCount) IITFileOrCDTra
SkippedCount([in] long skippedCount) IITFileOrCDTra
SkippedDate([out, retval] DATE *skippedDate) IITFileOrCDTra
SkippedDate([in] DATE skippedDate) IITFileOrCDTra
SortAlbum([out, retval] BSTR *album) IITFileOrCDTra
SortAlbum([in] BSTR album) IITFileOrCDTra
SortAlbumArtist([out, retval] BSTR *albumArtist) IITFileOrCDTra
SortAlbumArtist([in] BSTR albumArtist) IITFileOrCDTra
SortArtist([out, retval] BSTR *artist) IITFileOrCDTra
SortArtist([in] BSTR artist) IITFileOrCDTra
SortComposer([out, retval] BSTR *composer) IITFileOrCDTra
SortComposer([in] BSTR composer) IITFileOrCDTra
SortName([out, retval] BSTR *name) IITFileOrCDTra
SortName([in] BSTR name) IITFileOrCDTra
SortShow([out, retval] BSTR *showName) IITFileOrCDTra
SortShow([in] BSTR showName) IITFileOrCDTra
SourceID([out, retval] long *sourceID) IITObject
Start([out, retval] long *start) IITTrack

Start([in] long start)	IITTrack
Time([out, retval] BSTR *time)	IITTrack
TrackCount([out, retval] long *trackCount)	IITTrack
TrackCount([in] long trackCount)	IITTrack
TrackDatabaseID([out, retval] long *databaseID)	IITObject
TrackID([out, retval] long *trackID)	IITObject
TrackNumber([out, retval] long *trackNumber)	IITTrack
TrackNumber([in] long trackNumber)	IITTrack
<pre>Unplayed([out, retval] VARIANT_BOOL *isUnplayed)</pre>	IITFileOrCDTrack
Unplayed([in] VARIANT_BOOL shouldBeUnplayed)	IITFileOrCDTrack
UpdateInfoFromFile()	IITFileOrCDTrack
UpdatePodcastFeed()	IITFileOrCDTrack
VideoKind([out, retval] ITVideoKind *videoKind)	IITFileOrCDTrack
VideoKind([in] ITVideoKind videoKind)	IITFileOrCDTrack
VolumeAdjustment([out, retval] long *volumeAdjustment)	IITTrack
VolumeAdjustment([in] long volumeAdjustment)	IITTrack
Year([out, retval] long *year)	IITTrack
Year([in] long year)	IITTrack

IITIPodSource Member List

This is the complete list of members for **IITIPodSource**, including all inherited members.

Capacity([out, retval] double *capacity)	IITSource
EjectlPod()	IITIPodSource
FreeSpace([out, retval] double *freespace)	IITSource
GetITObjectIDs([out] long *sourceID,[out] long *playlistID,[out] long *trackID,[out] long *databaseID)	IITObject
Index([out, retval] long *index)	IITObject
Kind([out, retval] ITSourceKind *kind)	IITSource
Name([out, retval] BSTR *name)	IITObject
Name([in] BSTR name)	IITObject
PlaylistID([out, retval] long *playlistID)	IITObject
<pre>Playlists([out, retval] IITPlaylistCollection **iPlaylistCollection)</pre>	IITSource
SoftwareVersion ([out, retval] BSTR *softwareVersion)	IITIPodSource
SourceID([out, retval] long *sourceID)	IITObject
TrackDatabaseID([out, retval] long *databaseID)	IITObject
TrackID([out, retval] long *trackID)	IITObject
UpdatelPod()	IITIPodSource

IITLibraryPlaylist Member List

This is the complete list of members for **IITLibraryPlaylist**, including all inherited members.

AddFile([in] BSTR filePath,[out, retval] IITOperationStatus **iStatus)	IITLibraryPlaylist
AddFiles([in] VARIANT *filePaths,[out, retval] IITOperationStatus **iStatus)	IITLibraryPlaylist
AddTrack([in] VARIANT *iTrackToAdd,[out, retval] IITTrack **iAddedTrack)	IITLibraryPlaylist
AddURL([in] BSTR url,[out, retval] IITURLTrack **iURLTrack)	IITLibraryPlaylist
Delete()	IITPlaylist
Duration([out, retval] long *duration)	IITPlaylist
GetITObjectIDs([out] long *sourceID,[out] long *playlistID,[out] long *trackID,[out] long *databaseID)	IITObject
Index([out, retval] long *index)	IITObject
Kind([out, retval] ITPlaylistKind *kind)	IITPlaylist
Name([out, retval] BSTR *name)	IITObject
Name([in] BSTR name)	IITObject
PlayFirstTrack()	IITPlaylist
PlaylistID([out, retval] long *playlistID)	IITObject
<pre>Print([in] VARIANT_BOOL showPrintDialog,[in] ITPlaylistPrintKind printKind,[in] BSTR theme)</pre>	IITPlaylist
Search([in] BSTR searchText,[in] ITPlaylistSearchField searchFields,[out, retval] IITTrackCollection **iTrackCollection)	IITPlaylist
Shuffle([out, retval] VARIANT_BOOL *isShuffle)	IITPlaylist
Shuffle([in] VARIANT_BOOL shouldShuffle)	IITPlaylist
Size([out, retval] double *size)	IITPlaylist
SongRepeat([out, retval] ITPlaylistRepeatMode	

*repeatMode)	IITPlaylist
<pre>SongRepeat([in] ITPlaylistRepeatMode repeatMode)</pre>	IITPlaylist
Source([out, retval] IITSource **iSource)	IITPlaylist
SourceID([out, retval] long *sourceID)	IITObject
Time([out, retval] BSTR *time)	IITPlaylist
TrackDatabaseID([out, retval] long *databaseID)	IITObject
TrackID([out, retval] long *trackID)	IITObject
Tracks([out, retval] IITTrackCollection **iTrackCollection)	IITPlaylist
Visible([out, retval] VARIANT_BOOL *isVisible)	IITPlaylist

IITObject Member List

This is the complete list of members for **IITObject**, including all inherited members.

GetITObjectIDs([out] long *sourceID,[out] long *playlistID, [out] long *trackID,[out] long *databaseID)	IITObject
Index([out, retval] long *index)	IITObject
Name([out, retval] BSTR *name)	IITObject
Name([in] BSTR name)	IITObject
PlaylistID([out, retval] long *playlistID)	IITObject
SourceID([out, retval] long *sourceID)	IITObject
TrackDatabaseID([out, retval] long *databaseID)	IITObject
TrackID([out, retval] long *trackID)	IITObject

IITOperationStatus Member List

This is the complete list of members for **IITOperationStatus**, including all inherited members.

InProgress([out, retval] VARIANT_BOOL *isInProgress)	IITOperationStatus
Tracks([out, retval] IITTrackCollection **iTrackCollection)	IITOperationStatus

IITPlaylist Member List

This is the complete list of members for **IITPlaylist**, including all inherited members.

Delete()	IITPlaylist
Duration([out, retval] long *duration)	IITPlaylist
GetITObjectIDs([out] long *sourceID,[out] long *playlistID, [out] long *trackID,[out] long *databaseID)	IITObject
Index([out, retval] long *index)	IITObject
Kind([out, retval] ITPlaylistKind *kind)	IITPlaylist
Name([out, retval] BSTR *name)	IITObject
Name([in] BSTR name)	IITObject
PlayFirstTrack()	IITPlaylist
PlaylistID([out, retval] long *playlistID)	IITObject
<pre>Print([in] VARIANT_BOOL showPrintDialog,[in] ITPlaylistPrintKind printKind,[in] BSTR theme)</pre>	IITPlaylist
Search([in] BSTR searchText,[in] ITPlaylistSearchField searchFields,[out, retval] IITTrackCollection **iTrackCollection)	IITPlaylist
Shuffle([out, retval] VARIANT_BOOL *isShuffle)	IITPlaylist
Shuffle([in] VARIANT_BOOL shouldShuffle)	IITPlaylist
Size([out, retval] double *size)	IITPlaylist
SongRepeat([out, retval] ITPlaylistRepeatMode *repeatMode)	IITPlaylist
SongRepeat([in] ITPlaylistRepeatMode repeatMode)	IITPlaylist
Source([out, retval] IITSource **iSource)	IITPlaylist
SourceID([out, retval] long *sourceID)	IITObject
Time([out, retval] BSTR *time)	IITPlaylist
TrackDatabaseID([out, retval] long *databaseID)	IITObject
TrackID([out, retval] long *trackID)	IITObject

Tracks([out, retval] IITTrackCollection **iTrackCollection)	IITPlaylist
Visible([out, retval] VARIANT_BOOL *isVisible)	IITPlaylist

IITPlaylistCollection Member List

This is the complete list of members for **IITPlaylistCollection**, including all inherited members.

_NewEnum([out, retval] IUnknown **iEnumerator)	IITPlaylistCollection
Count([out, retval] long *count)	IITPlaylistCollection
<pre>Item([in] long index,[out, retval] IITPlaylist **iPlaylist)</pre>	IITPlaylistCollection
ItemByName([in] BSTR name,[out, retval] IITPlaylist **iPlaylist)	IITPlaylistCollection
ItemByPersistentID([in] long highID,[in] long lowID,[out, retval] IITPlaylist **iPlaylist)	IITPlaylistCollection

IITPlaylistWindow Member List

This is the complete list of members for **IITPlaylistWindow**, including all inherited members.

Bottom([out, retval] long *bottom)	IITWindow
Bottom([in] long bottom)	IITWindow
Height([out, retval] long *height)	IITWindow
Height([in] long height)	IITWindow
Kind([out, retval] ITWindowKind *kind)	IITWindow
Left([out, retval] long *left)	IITWindow
Left([in] long left)	IITWindow
Maximizable([out, retval] VARIANT_BOOL *isMaximizable)	IITWindow
Maximized([out, retval] VARIANT_BOOL *isMaximized)	IITWindow
Maximized([in] VARIANT_BOOL shouldBeMaximized)	IITWindow
Minimized([out, retval] VARIANT_BOOL *isMinimized)	IITWindow
Minimized([in] VARIANT_BOOL shouldBeMinimized)	IITWindow
Name([out, retval] BSTR *name)	IITWindow
Playlist([out, retval] IITPlaylist **iPlaylist)	IITPlaylistWindow
Resizable([out, retval] VARIANT_BOOL *isResizable)	IITWindow
Right([out, retval] long *right)	IITWindow
Right([in] long right)	IITWindow
SelectedTracks([out, retval] IITTrackCollection **iTrackCollection)	IITPlaylistWindow
Top([out, retval] long *top)	IITWindow
Top([in] long top)	IITWindow

Visible([out, retval] VARIANT_BOOL *isVisible)	IITWindow
Visible([in] VARIANT_BOOL shouldBeVisible)	IITWindow
Width([out, retval] long *width)	IITWindow
Width([in] long width)	IITWindow
Zoomable ([out, retval] VARIANT_BOOL *isZoomable)	IITWindow
Zoomed([out, retval] VARIANT_BOOL *isZoomed)	IITWindow
Zoomed ([in] VARIANT_BOOL shouldBeZoomed)	IITWindow

IITSource Member List

This is the complete list of members for **IITSource**, including all inherited members.

Capacity([out, retval] double *capacity)	IITSource
FreeSpace([out, retval] double *freespace)	IITSource
GetITObjectIDs([out] long *sourceID,[out] long *playlistID, [out] long *trackID,[out] long *databaseID)	IITObject
Index([out, retval] long *index)	IITObject
Kind([out, retval] ITSourceKind *kind)	IITSource
Name([out, retval] BSTR *name)	IITObject
Name([in] BSTR name)	IITObject
PlaylistID([out, retval] long *playlistID)	IITObject
Playlists([out, retval] IITPlaylistCollection **iPlaylistCollection)	IITSource
SourceID([out, retval] long *sourceID)	IITObject
TrackDatabaseID([out, retval] long *databaseID)	IITObject
TrackID([out, retval] long *trackID)	IITObject

IITSourceCollection Member List

This is the complete list of members for **IITSourceCollection**, including all inherited members.

_NewEnum([out, retval] IUnknown **iEnumerator)	IITSourceCollection
Count([out, retval] long *count)	IITSourceCollection
<pre>Item([in] long index,[out, retval] IITSource **iSource)</pre>	IITSourceCollection
ItemByName([in] BSTR name,[out, retval] IITSource **iSource)	IITSourceCollection
ItemByPersistentID([in] long highID,[in] long lowID,[out, retval] IITSource **iSource)	IITSourceCollection

IITTrack Member List

This is the complete list of members for **IITTrack**, including all inherited members.

AddArtworkFromFile([in] BSTR filePath,[out, retval] IITArtwork **iArtwork)	IITTrack
Album([out, retval] BSTR *album)	IITTrack
Album([in] BSTR album)	IITTrack
Artist([out, retval] BSTR *artist)	IITTrack
Artist([in] BSTR artist)	IITTrack
Artwork([out, retval] IITArtworkCollection **iArtworkCollection)	IITTrack
BitRate([out, retval] long *bitrate)	IITTrack
BPM([out, retval] long *beatsPerMinute)	IITTrack
BPM([in] long beatsPerMinute)	IITTrack
Comment([out, retval] BSTR *comment)	IITTrack
Comment([in] BSTR comment)	IITTrack
Compilation([out, retval] VARIANT_BOOL *isCompilation)	IITTrack
Compilation([in] VARIANT_BOOL shouldBeCompilation)	IITTrack
Composer([out, retval] BSTR *composer)	IITTrack
Composer([in] BSTR composer)	IITTrack
DateAdded([out, retval] DATE *dateAdded)	IITTrack
Delete()	IITTrack
DiscCount([out, retval] long *discCount)	IITTrack
DiscCount([in] long discCount)	IITTrack
DiscNumber([out, retval] long *discNumber)	IITTrack
DiscNumber([in] long discNumber)	IITTrack
Duration([out, retval] long *duration)	IITTrack
Enabled([out, retval] VARIANT_BOOL *isEnabled)	IITTrack
Enabled([in] VARIANT_BOOL shouldBeEnabled)	IITTrack

EQ([out, retval] BSTR *eq)	IITTrack
EQ([in] BSTR eq)	IITTrack
Finish([in] long finish)	IITTrack
Finish([out, retval] long *finish)	IITTrack
Genre([out, retval] BSTR *genre)	IITTrack
Genre([in] BSTR genre)	IITTrack
GetITObjectIDs([out] long *sourceID,[out] long *playlistID, [out] long *trackID,[out] long *databaseID)	IITObject
Grouping([out, retval] BSTR *grouping)	IITTrack
Grouping([in] BSTR grouping)	IITTrack
Index([out, retval] long *index)	IITObject
Kind([out, retval] ITTrackKind *kind)	IITTrack
KindAsString([out, retval] BSTR *kind)	IITTrack
ModificationDate([out, retval] DATE *dateModified)	IITTrack
Name([out, retval] BSTR *name)	IITObject
Name([in] BSTR name)	IITObject
Play()	IITTrack
PlayedCount([out, retval] long *playedCount)	IITTrack
PlayedCount([in] long playedCount)	IITTrack
PlayedDate([out, retval] DATE *playedDate)	IITTrack
PlayedDate([in] DATE playedDate)	IITTrack
Playlist([out, retval] IITPlaylist **iPlaylist)	IITTrack
PlaylistID([out, retval] long *playlistID)	IITObject
PlayOrderIndex([out, retval] long *index)	IITTrack
Rating([out, retval] long *rating)	IITTrack
Rating([in] long rating)	IITTrack
SampleRate([out, retval] long *sampleRate)	IITTrack
Size([out, retval] long *size)	IITTrack
SourceID([out, retval] long *sourceID)	IITObject
Start([out, retval] long *start)	IITTrack
Start([in] long start)	IITTrack
Time([out, retval] BSTR *time)	IITTrack

TrackCount([out, retval] long *trackCount)	IITTrack
TrackCount([in] long trackCount)	IITTrack
TrackDatabaseID([out, retval] long *databaseID)	IITObject
TrackID([out, retval] long *trackID)	IITObject
TrackNumber([out, retval] long *trackNumber)	IITTrack
TrackNumber([in] long trackNumber)	IITTrack
VolumeAdjustment([out, retval] long *volumeAdjustment)	IITTrack
VolumeAdjustment([in] long volumeAdjustment)	IITTrack
Year([out, retval] long *year)	IITTrack
Year([in] long year)	IITTrack

IITTrackCollection Member List

This is the complete list of members for **IITTrackCollection**, including all inherited members.

_NewEnum([out, retval] IUnknown **iEnumerator)	IITTrackCollection
Count([out, retval] long *count)	IITTrackCollection
Item([in] long index,[out, retval] IITTrack **iTrack)	IITTrackCollection
<pre>ItemByName([in] BSTR name,[out, retval] IITTrack **iTrack)</pre>	IITTrackCollection
ItemByPersistentID([in] long highID,[in] long lowID, [out, retval] IITTrack **iTrack)	IITTrackCollection
ItemByPlayOrder([in] long index,[out, retval] IITTrack **iTrack)	IITTrackCollection

liTunes Member List

This is the complete list of members for **liTunes**, including all inherited members.

AppCommandMessageProcessingEnabled([out, retval] VARIANT_BOOL *isEnabled)	liTunes
AppCommandMessageProcessingEnabled([in] VARIANT_BOOL shouldEnable)	liTunes
BackTrack()	liTunes
BrowserWindow([out, retval] IITBrowserWindow **iBrowserWindow)	liTunes
CanSetShuffle([in] VARIANT *iPlaylist,[out, retval] VARIANT_BOOL *canSetShuffle)	liTunes
CanSetSongRepeat([in] VARIANT *iPlaylist,[out, retval] VARIANT_BOOL *canSetSongRepeat)	liTunes
CheckVersion([in] long majorVersion,[in] long minorVersion, [out, retval] VARIANT_BOOL *isCompatible)	liTunes
ConvertFile ([in] BSTR filePath,[out, retval] IITOperationStatus **iStatus)	liTunes
ConvertFile2([in] BSTR filePath,[out, retval] IITConvertOperationStatus **iStatus)	liTunes
ConvertFiles([in] VARIANT *filePaths,[out, retval] IITOperationStatus **iStatus)	liTunes
ConvertFiles2([in] VARIANT *filePaths,[out, retval] IITConvertOperationStatus **iStatus)	liTunes
ConvertOperationStatus([out, retval] IITConvertOperationStatus **iStatus)	liTunes
ConvertTrack([in] VARIANT *iTrackToConvert,[out, retval] IITOperationStatus **iStatus)	liTunes
ConvertTrack2([in] VARIANT *iTrackToConvert,[out, retval] IITConvertOperationStatus **iStatus)	liTunes

ConvertTracks([in] VARIANT *iTracksToConvert,[out, retval] IITOperationStatus **iStatus)	liTunes
ConvertTracks2([in] VARIANT *iTracksToConvert,[out, retval] IITConvertOperationStatus **iStatus)	liTunes
CreateEQPreset([in] BSTR eqPresetName,[out, retval] IITEQPreset **iEQPreset)	liTunes
CreateFolder ([in] BSTR folderName,[out, retval] IITPlaylist **iFolder)	liTunes
CreateFolderInSource ([in] BSTR folderName,[in] VARIANT *iSource,[out, retval] IITPlaylist **iFolder)	liTunes
<pre>CreatePlaylist([in] BSTR playlistName,[out, retval] IITPlaylist **iPlaylist)</pre>	liTunes
CreatePlaylistInSource ([in] BSTR playlistName,[in] VARIANT *iSource,[out, retval] IITPlaylist **iPlaylist)	liTunes
CurrentEncoder([out, retval] IITEncoder **iEncoder)	liTunes
CurrentEncoder([in] IITEncoder *iEncoder)	liTunes
CurrentEQPreset([out, retval] IITEQPreset **iEQPreset)	liTunes
CurrentEQPreset([in] IITEQPreset *iEQPreset)	liTunes
CurrentPlaylist([out, retval] IITPlaylist **iPlaylist)	liTunes
CurrentStreamTitle([out, retval] BSTR *streamTitle)	liTunes
CurrentStreamURL([out, retval] BSTR *streamURL)	liTunes
CurrentTrack([out, retval] IITTrack **iTrack)	liTunes
CurrentVisual([out, retval] IITVisual **iVisual)	liTunes
CurrentVisual([in] IITVisual *iVisual)	liTunes
Encoders([out, retval] IITEncoderCollection **iEncoderCollection)	liTunes
EQEnabled ([out, retval] VARIANT_BOOL *isEnabled)	liTunes
EQEnabled([in] VARIANT_BOOL shouldEnable)	liTunes
EQPresets ([out, retval] IITEQPresetCollection **iEQPresetCollection)	liTunes
EQWindow([out, retval] IITWindow **iEQWindow)	liTunes
FastForward()	liTunes
ForceToForegroundOnDialog([out, retval] VARIANT_BOOL	liTunes

*forceToForceroundOnDiolog)	
*forceToForegroundOnDialog)	
ForceToForegroundOnDialog([in] VARIANT_BOOL	liTunes
forceToForegroundOnDialog)	
FullScreenVisuals([out, retval] VARIANT_BOOL *isFullScreen)	liTunes
FullScreenVisuals([in] VARIANT_BOOL shouldUseFullScreen)	liTunes
GetITObjectByID([in] long sourceID,[in] long playlistID,[in] long	liTunes
trackID,[in] long databaseID,[out, retval] IITObject **iObject)	irrunes
GetITObjectPersistentIDs([in] VARIANT *iObject,[out] long	liTunes
*highID,[out] long *lowID)	IIIulies
GetPlayerButtonsState([out] VARIANT_BOOL	
*previousEnabled,[out] ITPlayButtonState	liTunes
*playPauseStopState,[out] VARIANT_BOOL *nextEnabled)	
GotoMusicStoreHomePage()	liTunes
ITObjectPersistentIDHigh([in] VARIANT *iObject,[out, retval]	liTunes
long *highID)	
ITObjectPersistentIDLow([in] VARIANT *iObject,[out, retval]	liTunes
long *lowID)	
LibraryPlaylist([out, retval] IITLibraryPlaylist **iLibraryPlaylist)	liTunes
LibrarySource([out, retval] IITSource **iLibrarySource)	liTunes
LibraryXMLPath([out, retval] BSTR *filePath)	liTunes
Mute([out, retval] VARIANT_BOOL *isMuted)	liTunes
Mute([in] VARIANT_BOOL shouldMute)	liTunes
NextTrack()	liTunes
OpenURL([in] BSTR url)	liTunes
Pause()	liTunes
Play()	liTunes
PlayerButtonClicked([in] ITPlayerButton playerButton,[in] long	
playerButtonModifierKeys)	liTunes
PlayerPosition([out, retval] long *playerPos)	liTunes
PlayerPosition([in] long playerPos)	liTunes
PlayerState([out, retval] ITPlayerState *playerState)	liTunes
PlayFile([in] BSTR filePath)	liTunes
PlayPause()	liTunes
- 10 , 1 110 0	

PreviousTrack()	liTunes
Quit()	liTunes
Resume()	liTunes
Rewind()	liTunes
SelectedTracks([out, retval] IITTrackCollection **iTrackCollection)	liTunes
SoundVolume([out, retval] long *volume)	liTunes
SoundVolume([in] long volume)	liTunes
SoundVolumeControlEnabled ([out, retval] VARIANT_BOOL *isEnabled)	liTunes
Sources([out, retval] IITSourceCollection **iSourceCollection)	liTunes
Stop()	liTunes
SubscribeToPodcast([in] BSTR url)	liTunes
UpdatelPod()	liTunes
UpdatePodcastFeeds()	liTunes
Version([out, retval] BSTR *version)	liTunes
Visuals([out, retval] IITVisualCollection **iVisualCollection)	liTunes
VisualsEnabled([out, retval] VARIANT_BOOL *isEnabled)	liTunes
VisualsEnabled([in] VARIANT_BOOL shouldEnable)	liTunes
VisualSize([out, retval] ITVisualSize *visualSize)	liTunes
VisualSize([in] ITVisualSize visualSize)	liTunes
Windows([out, retval] IITWindowCollection **iWindowCollection)	liTunes

IITURLTrack Member List

This is the complete list of members for **IITURLTrack**, including all inherited members.

AddArtworkFromFile([in] BSTR filePath,[out, retval] IITArtwork **iArtwork)	IITTrack
Album([out, retval] BSTR *album)	IITTrack
Album([in] BSTR album)	IITTrack
AlbumRating([out, retval] long *rating)	IITURLTrack
AlbumRating([in] long rating)	IITURLTrack
AlbumRatingKind([out, retval] ITRatingKind *ratingKind)	IITURLTrack
Artist([out, retval] BSTR *artist)	IITTrack
Artist([in] BSTR artist)	IITTrack
Artwork([out, retval] IITArtworkCollection **iArtworkCollection)	IITTrack
BitRate([out, retval] long *bitrate)	IITTrack
BPM([out, retval] long *beatsPerMinute)	IITTrack
BPM([in] long beatsPerMinute)	IITTrack
Category([out, retval] BSTR *category)	IITURLTrack
Category([in] BSTR category)	IITURLTrack
Comment([out, retval] BSTR *comment)	IITTrack
Comment([in] BSTR comment)	IITTrack
Compilation([out, retval] VARIANT_BOOL *isCompilation)	IITTrack
Compilation([in] VARIANT_BOOL shouldBeCompilation)	IITTrack
Composer([out, retval] BSTR *composer)	IITTrack
Composer([in] BSTR composer)	IITTrack
DateAdded([out, retval] DATE *dateAdded)	IITTrack
Delete()	IITTrack
Description([out, retval] BSTR *description)	IITURLTrack

Description([in] BSTR description)	IITURLTrack
DiscCount([out, retval] long *discCount)	IITTrack
DiscCount([in] long discCount)	IITTrack
DiscNumber([out, retval] long *discNumber)	IITTrack
DiscNumber([in] long discNumber)	IITTrack
DownloadPodcastEpisode()	IITURLTrack
Duration([out, retval] long *duration)	IITTrack
Enabled([out, retval] VARIANT_BOOL *isEnabled)	IITTrack
Enabled([in] VARIANT_BOOL shouldBeEnabled)	IITTrack
EQ([out, retval] BSTR *eq)	IITTrack
EQ([in] BSTR eq)	IITTrack
Finish([in] long finish)	IITTrack
Finish([out, retval] long *finish)	IITTrack
Genre([out, retval] BSTR *genre)	IITTrack
Genre([in] BSTR genre)	IITTrack
GetITObjectIDs ([out] long *sourceID,[out] long *playlistID, [out] long *trackID,[out] long *databaseID)	IITObject
Grouping([out, retval] BSTR *grouping)	IITTrack
Grouping([in] BSTR grouping)	IITTrack
Index([out, retval] long *index)	IITObject
Kind([out, retval] ITTrackKind *kind)	IITTrack
KindAsString([out, retval] BSTR *kind)	IITTrack
LongDescription([out, retval] BSTR *longDescription)	IITURLTrack
LongDescription([in] BSTR longDescription)	IITURLTrack
ModificationDate([out, retval] DATE *dateModified)	IITTrack
Name([out, retval] BSTR *name)	IITObject
Name([in] BSTR name)	IITObject
Play()	IITTrack
PlayedCount([out, retval] long *playedCount)	IITTrack
PlayedCount([in] long playedCount)	IITTrack
PlayedDate([out, retval] DATE *playedDate)	IITTrack
PlayedDate([in] DATE playedDate)	IITTrack

Playlist([out, retval] IITPlaylist **iPlaylist)	IITTrack
PlaylistID([out, retval] long *playlistID)	IITObject
<pre>Playlists([out, retval] IITPlaylistCollection **iPlaylistCollection)</pre>	IITURLTrack
PlayOrderIndex([out, retval] long *index)	IITTrack
Podcast([out, retval] VARIANT_BOOL *isPodcast)	IITURLTrack
Rating([out, retval] long *rating)	IITTrack
Rating([in] long rating)	IITTrack
RatingKind([out, retval] ITRatingKind *ratingKind)	IITURLTrack
Reveal()	IITURLTrack
SampleRate([out, retval] long *sampleRate)	IITTrack
Size([out, retval] long *size)	IITTrack
SourceID([out, retval] long *sourceID)	IITObject
Start([out, retval] long *start)	IITTrack
Start([in] long start)	IITTrack
Time([out, retval] BSTR *time)	IITTrack
TrackCount([out, retval] long *trackCount)	IITTrack
TrackCount([in] long trackCount)	IITTrack
TrackDatabaseID([out, retval] long *databaseID)	IITObject
TrackID([out, retval] long *trackID)	IITObject
TrackNumber([out, retval] long *trackNumber)	IITTrack
TrackNumber([in] long trackNumber)	IITTrack
UpdatePodcastFeed()	IITURLTrack
URL([out, retval] BSTR *url)	IITURLTrack
URL([in] BSTR url)	IITURLTrack
VolumeAdjustment([out, retval] long *volumeAdjustment)	IITTrack
VolumeAdjustment([in] long volumeAdjustment)	IITTrack
Year([out, retval] long *year)	IITTrack
Year([in] long year)	IITTrack

IITUserPlaylist Member List

This is the complete list of members for **IITUserPlaylist**, including all inherited members.

AddFile([in] BSTR filePath,[out, retval] IITOperationStatus **iStatus)	IITUserPlaylist
AddFiles([in] VARIANT *filePaths,[out, retval] IITOperationStatus **iStatus)	IITUserPlaylist
AddTrack([in] VARIANT *iTrackToAdd,[out, retval] IITTrack **iAddedTrack)	IITUserPlaylist
AddURL([in] BSTR url,[out, retval] IITURLTrack **iURLTrack)	IITUserPlaylist
CreateFolder([in] BSTR folderName,[out, retval] IITPlaylist **iFolder)	IITUserPlaylist
CreatePlaylist([in] BSTR playlistName,[out, retval] IITPlaylist **iPlaylist)	IITUserPlaylist
Delete()	IITPlaylist
Duration([out, retval] long *duration)	IITPlaylist
GetITObjectIDs([out] long *sourceID,[out] long *playlistID,[out] long *trackID,[out] long *databaseID)	IITObject
Index([out, retval] long *index)	IITObject
Kind([out, retval] ITPlaylistKind *kind)	IITPlaylist
Name([out, retval] BSTR *name)	IITObject
Name([in] BSTR name)	IITObject
Parent([out, retval] IITUserPlaylist **iParentPlayList)	IITUserPlaylist
Parent([in] VARIANT *iParent)	IITUserPlaylist
PlayFirstTrack()	IITPlaylist
PlaylistID([out, retval] long *playlistID)	IITObject
<pre>Print([in] VARIANT_BOOL showPrintDialog,[in] ITPlaylistPrintKind printKind,[in] BSTR theme)</pre>	IITPlaylist
Reveal()	IITUserPlaylist

Search ([in] BSTR searchText,[in] ITPlaylistSearchField searchFields,[out, retval] IITTrackCollection **iTrackCollection)	IITPlaylist
Shared([out, retval] VARIANT_BOOL *isShared)	IITUserPlaylist
Shared([in] VARIANT_BOOL shouldBeShared)	IITUserPlaylist
Shuffle([out, retval] VARIANT_BOOL *isShuffle)	IITPlaylist
Shuffle([in] VARIANT_BOOL shouldShuffle)	IITPlaylist
Size([out, retval] double *size)	IITPlaylist
Smart([out, retval] VARIANT_BOOL *isSmart)	IITUserPlaylist
SongRepeat([out, retval] ITPlaylistRepeatMode *repeatMode)	IITPlaylist
SongRepeat([in] ITPlaylistRepeatMode repeatMode)	IITPlaylist
Source([out, retval] IITSource **iSource)	IITPlaylist
SourceID([out, retval] long *sourceID)	IITObject
SpecialKind ([out, retval] ITUserPlaylistSpecialKind *specialKind)	IITUserPlaylist
Time([out, retval] BSTR *time)	IITPlaylist
TrackDatabaseID([out, retval] long *databaseID)	IITObject
TrackID([out, retval] long *trackID)	IITObject
Tracks ([out, retval] IITTrackCollection **iTrackCollection)	IITPlaylist
Visible([out, retval] VARIANT_BOOL *isVisible)	IITPlaylist

IITVisual Member List

This is the complete list of members for **IITVisual**, including all inherited members.

Name([out, retval] BSTR *name) IITVisual

IITVisualCollection Member List

This is the complete list of members for **IITVisualCollection**, including all inherited members.

_NewEnum([out, retval] IUnknown **iEnumerator)	IITVisualCollection
Count([out, retval] long *count)	IITVisualCollection
Item([in] long index,[out, retval] IITVisual **iVisual)	IITVisualCollection
ItemByName([in] BSTR name,[out, retval] IITVisual **iVisual)	IITVisualCollection

IITWindow Member List

This is the complete list of members for **IITWindow**, including all inherited members.

Bottom([out, retval] long *bottom)	IITWindow
Bottom([in] long bottom)	IITWindow
Height([out, retval] long *height)	IITWindow
Height([in] long height)	IITWindow
Kind([out, retval] ITWindowKind *kind)	IITWindow
Left([out, retval] long *left)	IITWindow
Left([in] long left)	IITWindow
Maximizable([out, retval] VARIANT_BOOL *isMaximizable)	IITWindow
Maximized([out, retval] VARIANT_BOOL *isMaximized)	IITWindow
Maximized([in] VARIANT_BOOL shouldBeMaximized)	IITWindow
Minimized([out, retval] VARIANT_BOOL *isMinimized)	IITWindow
Minimized([in] VARIANT_BOOL shouldBeMinimized)	IITWindow
Name([out, retval] BSTR *name)	IITWindow
Resizable([out, retval] VARIANT_BOOL *isResizable)	IITWindow
Right([out, retval] long *right)	IITWindow
Right([in] long right)	IITWindow
Top([out, retval] long *top)	IITWindow
Top([in] long top)	IITWindow
Visible([out, retval] VARIANT_BOOL *isVisible)	IITWindow
Visible([in] VARIANT_BOOL shouldBeVisible)	IITWindow
Width([out, retval] long *width)	IITWindow
Width([in] long width)	IITWindow
Zoomable([out, retval] VARIANT_BOOL *isZoomable)	IITWindow
Zoomed([out, retval] VARIANT_BOOL *isZoomed)	IITWindow
Zoomed([in] VARIANT_BOOL shouldBeZoomed)	IITWindow

IITWindowCollection Member List

This is the complete list of members for **IITWindowCollection**, including all inherited members.

_ NewEnum ([out, retval] IUnknown **iEnumerator)	IITWindowCollection
Count([out, retval] long *count)	IITWindowCollection
<pre>Item([in] long index,[out, retval] IITWindow **iWindow)</pre>	IITWindowCollection
ItemByName([in] BSTR name,[out, retval] IITWindow **iWindow)	IITWindowCollection

All | Functions

_|a|b|c|d|e|f|g|h|i|k|l|m|n|o|p|q|r|s|t|u|v|w|y z

-_-

 NewEnum(): IITWindowCollection, IITVisualCollection, IITTrackCollection, IITSourceCollection, IITPlaylistCollection, IITEQPresetCollection, IITEncoderCollection, IITArtworkCollection

- a -

- AddArtworkFromFile(): IITTrack
- AddFile(): IITUserPlaylist, IITLibraryPlaylist
- AddFiles(): IITUserPlaylist, IITLibraryPlaylist
- AddTrack(): IITUserPlaylist, IITLibraryPlaylist
- AddURL(): IITUserPlaylist, IITLibraryPlaylist
- Album(): IITTrack
- AlbumArtist(): IITFileOrCDTrack
- AlbumRating(): IITURLTrack, IITFileOrCDTrack
- AlbumRatingKind(): IITURLTrack, IITFileOrCDTrack
- AppCommandMessageProcessingEnabled(): IiTunes
- Artist(): IITTrack, IITAudioCDPlaylist
- Artwork(): IITTrack

- b -

BackTrack(): IiTunes
Band1(): IITEQPreset
Band10(): IITEQPreset
Band2(): IITEQPreset

- Band3(): IITEQPreset
- Band4(): IITEQPreset
- Band5(): IITEQPreset
- Band6(): IITEQPreset
- Band7(): IITEQPreset
- Band8(): IITEQPreset
- Band9(): IITEQPreset
- BitRate(): IITTrack
- BookmarkTime(): IITFileOrCDTrack
- Bottom(): IITWindow
- BPM(): IITTrack
- BrowserWindow(): IiTunes

- C -

- CanSetShuffle(): IiTunes
- CanSetSongRepeat(): liTunes
- Capacity(): IITSource
- Category(): IITURLTrack, IITFileOrCDTrack
- CheckVersion(): IiTunes
- Comment(): IITTrack
- Compilation(): IITTrack, IITAudioCDPlaylist
- Composer(): IITTrack, IITAudioCDPlaylist
- ConvertFile(): liTunes
- ConvertFile2(): IiTunes
- ConvertFiles(): liTunes
- ConvertFiles2(): IiTunes
- ConvertOperationStatus(): IiTunes
- ConvertTrack(): liTunes
- ConvertTrack2(): IiTunes
- ConvertTracks(): liTunes
- ConvertTracks2(): IiTunes
- Count(): IITWindowCollection, IITVisualCollection, IITTrackCollection, IITSourceCollection, IITPlaylistCollection, IITEQPresetCollection, IITEncoderCollection, IITArtworkCollection
- CreateEQPreset(): IiTunes

- CreateFolder(): IITUserPlaylist, IiTunes
- CreateFolderInSource(): IiTunes
- CreatePlaylist(): IITUserPlaylist, IiTunes
- CreatePlaylistInSource(): liTunes
- CurrentEncoder(): IiTunes
- CurrentEQPreset(): liTunes
- CurrentPlaylist(): IiTunes
- CurrentStreamTitle(): liTunes
- CurrentStreamURL(): liTunes
- CurrentTrack(): liTunes
- CurrentVisual(): IiTunes

- d -

- DateAdded(): IITTrack
- Delete(): IITTrack, IITPlaylist, IITEQPreset, IITArtwork
- Description(): IITURLTrack, IITFileOrCDTrack, IITArtwork
- DiscCount(): IITTrack, IITAudioCDPlaylist
- DiscNumber(): IITTrack, IITAudioCDPlaylist
- DownloadPodcastEpisode(): IITURLTrack
- Duration(): IITTrack, IITPlaylist

- e -

- EjectlPod(): IITIPodSource
- Enabled(): IITTrack
- Encoders(): IiTunes
- EpisodeID(): IITFileOrCDTrack
- EpisodeNumber(): IITFileOrCDTrack
- EQ(): IITTrack
- EQEnabled(): liTunes
- EQPresets(): liTunes
- EQWindow(): IiTunes
- ExcludeFromShuffle(): IITFileOrCDTrack

- FastForward(): IiTunes
- Finish(): IITTrack
- ForceToForegroundOnDialog(): IiTunes
- Format(): IITEncoder, IITArtwork
- FreeSpace(): IITSource
- FullScreenVisuals(): IiTunes

- g -

- Genre(): IITTrack, IITAudioCDPlaylist
- GetConversionStatus(): IITConvertOperationStatus
- GetITObjectByID(): IiTunes
- GetITObjectIDs(): IITObject
- GetITObjectPersistentIDs(): liTunes
- GetPlayerButtonsState(): IiTunes
- GotoMusicStoreHomePage(): liTunes
- Grouping(): IITTrack

- h -

Height(): IITWindow

- j -

- Index(): IITObject
- InProgress(): IITOperationStatus
- IsDownloadedArtwork(): IITArtwork
- Item(): IITWindowCollection, IITVisualCollection, IITTrackCollection, IITSourceCollection, IITPlaylistCollection, IITEQPresetCollection, IITEncoderCollection, IITArtworkCollection
- ItemByName(): IITWindowCollection, IITVisualCollection, IITTrackCollection, IITSourceCollection, IITPlaylistCollection, IITEQPresetCollection, IITEncoderCollection
- ItemByPersistentID(): IITTrackCollection, IITSourceCollection, IITPlaylistCollection
- ItemByPlayOrder(): IITTrackCollection

- ITObjectPersistentIDHigh(): IiTunes
- ITObjectPersistentIDLow(): IiTunes

- k -

- Kind(): IITWindow, IITTrack, IITSource, IITPlaylist
- KindAsString(): IITTrack

- | -

- Left(): IITWindow
- LibraryPlaylist(): IiTunes
- LibrarySource(): liTunes
- LibraryXMLPath(): IiTunes
- Location(): IITFileOrCDTrack
- LongDescription(): IITURLTrack, IITFileOrCDTrack
- Lyrics(): IITFileOrCDTrack

- m -

- Maximizable(): IITWindow
- Maximized(): IITWindow
- MaxProgressValue(): IITConvertOperationStatus
- Minimized(): IITWindow
- MiniPlayer(): IITBrowserWindow
- Modifiable(): IITEQPreset
- ModificationDate(): IITTrack
- Mute(): IiTunes

- n -

- Name(): IITWindow, IITVisual, IITObject, IITEQPreset, IITEncoder
- NextTrack(): liTunes

- 0 -

- OnAboutToPromptUserToQuitEvent(): _liTunesEvents
- OnCOMCallsDisabledEvent(): _liTunesEvents
- OnCOMCallsEnabledEvent(): _liTunesEvents
- OnConvertOperationCompleteEvent():
 IITConvertOperationStatusEvents
- OnConvertOperationStatusChangedEvent() : _IITConvertOperationStatusEvents
- OnDatabaseChangedEvent(): _liTunesEvents
- OnPlayerPlayEvent(): _liTunesEvents
- OnPlayerPlayingTrackChangedEvent(): _liTunesEvents
- OnPlayerStopEvent(): _liTunesEvents
- OnQuittingEvent(): _liTunesEvents
- OnSoundVolumeChangedEvent(): IiTunesEvents
- OpenURL(): liTunes

- p -

- Parent(): IITUserPlaylist
- PartOfGaplessAlbum(): IITFileOrCDTrack
- Pause(): IiTunes
- Play(): IITTrack, IiTunes
- PlayedCount(): IITTrack
- PlayedDate(): IITTrack
- PlayerButtonClicked(): IiTunes
- PlayerPosition(): IiTunes
- PlayerState(): liTunes
- PlayFile(): liTunes
- PlayFirstTrack(): IITPlaylist
- Playlist(): IITPlaylistWindow, IITTrack
- PlaylistID(): IITObject
- Playlists(): IITURLTrack, IITFileOrCDTrack, IITSource
- PlayOrderIndex(): IITTrack
- PlayPause(): liTunes
- Podcast(): IITURLTrack, IITFileOrCDTrack
- Preamp(): IITEQPreset
- PreviousTrack(): liTunes
- Print(): IITPlaylist

ProgressValue(): IITConvertOperationStatus

- q -

• Quit(): IiTunes

- r -

- Rating(): IITTrack
- RatingKind(): IITURLTrack, IITFileOrCDTrack
- ReleaseDate(): IITFileOrCDTrack
- RememberBookmark(): IITFileOrCDTrack
- Rename(): IITEQPreset
- Resizable(): IITWindow
- Resume(): liTunes
- Reveal(): IITURLTrack, IITFileOrCDTrack, IITAudioCDPlaylist, IITUserPlaylist
- Rewind(): IiTunesRight(): IITWindow

- S -

- SampleRate(): IITTrack
- SaveArtworkToFile(): IITArtwork
- Search(): IITPlaylist
- SeasonNumber(): IITFileOrCDTrack
- SelectedPlaylist(): IITBrowserWindow
- SelectedTracks(): IITPlaylistWindow, IITBrowserWindow, IiTunes
- SetArtworkFromFile(): IITArtwork
- Shared(): IITUserPlaylist
- Show(): IITFileOrCDTrack
- Shuffle(): IITPlaylist
- Size(): IITTrack, IITPlaylist
- Size64High(): IITFileOrCDTrack
- Size64Low(): IITFileOrCDTrack
- SkippedCount(): IITFileOrCDTrack
- SkippedDate(): IITFileOrCDTrack

- Smart(): IITUserPlaylist
- SoftwareVersion(): IITIPodSource
- SongRepeat(): IITPlaylist
- SortAlbum(): IITFileOrCDTrack
- SortAlbumArtist(): IITFileOrCDTrack
- SortArtist(): IITFileOrCDTrack
- SortComposer() : IITFileOrCDTrack
- SortName(): IITFileOrCDTrack
- SortShow(): IITFileOrCDTrack
- SoundVolume(): liTunes
- SoundVolumeControlEnabled(): IiTunes
- Source(): IITPlaylist
- SourceID(): IITObject
- Sources(): liTunes
- SpecialKind(): IITUserPlaylist
- Start(): IITTrack
- Stop(): IiTunes
- StopConversion(): IITConvertOperationStatus
- SubscribeToPodcast(): IiTunes

- t -

- Time(): IITTrack, IITPlaylist
- Top(): IITWindow
- TrackCount(): IITTrack
- TrackDatabaseID(): IITObject
- TrackID(): IITObject
- TrackName(): IITConvertOperationStatus
- TrackNumber() : IITTrack
- Tracks(): IITPlaylist, IITOperationStatus

- u -

- Unplayed(): IITFileOrCDTrack
- UpdateInfoFromFile(): IITFileOrCDTrack
- UpdatelPod(): IITIPodSource, IiTunes
- UpdatePodcastFeed(): IITURLTrack, IITFileOrCDTrack

- UpdatePodcastFeeds(): IiTunes
- URL(): IITURLTrack

- V -

- Version(): IiTunes
- VideoKind(): IITFileOrCDTrack
 Visible(): IITMindow, IITPleydia
- Visible(): IITWindow, IITPlaylist
- Visuals(): IiTunes
- VisualsEnabled(): IiTunes
- VisualSize(): IiTunes
- VolumeAdjustment(): IITTrack

- W -

- Width(): IITWindowWindows(): IiTunes
- **y** -
 - Year(): IITTrack, IITAudioCDPlaylist
- Z -
 - Zoomable() : IITWindow
 - Zoomed(): IITWindow

All | Functions | Enumerations | Enumeration values

• importlib(): iTunesCOMInterface.idl

All | Functions | Enumerations | Enumeration values

- ITArtworkFormat : iTunesArtworkCOM.idl
- ITCOMDisabledReason : iTunesAppCOM.idl
- ITConvertOperationStatusEvent : iTunesAppCOM.idl
- ITErrors : iTunesCOMInterface.idl
- ITEvent : iTunesAppCOM.idl
- ITPlayButtonState : iTunesAppCOM.idl
- ITPlayerButton : iTunesAppCOM.idl
- ITPlayerButtonModifierKey: iTunesAppCOM.idl
- ITPlayerState : iTunesAppCOM.idl
- ITPlaylistKind : iTunesPlaylistCOM.idl
- ITPlaylistPrintKind : iTunesPlaylistCOM.idl
- ITPlaylistRepeatMode : iTunesPlaylistCOM.idl
- ITPlaylistSearchField : iTunesPlaylistCOM.idl
- ITRatingKind : iTunesTrackCOM.idl
- ITSourceKind: iTunesSourceCOM.idl
- ITTrackKind: iTunesTrackCOM.idl
- ITUserPlaylistSpecialKind : iTunesPlaylistCOM.idl
- ITVersion : iTunesCOMInterface.idl
- ITVideoKind : iTunesTrackCOM.idl
- ITVisualSize : iTunesAppCOM.idl
- ITWindowKind: iTunesWindowCOM.idl

All | Functions | Enumerations | Enumeration values

i | k

- j -

- ITArtworkFormatBMP : iTunesArtworkCOM.idl
- ITArtworkFormatJPEG: iTunesArtworkCOM.idl
- ITArtworkFormatPNG: iTunesArtworkCOM.idl
- ITArtworkFormatUnknown: iTunesArtworkCOM.idl
- ITCOMDisabledReasonDialog: iTunesAppCOM.idl
- ITCOMDisabledReasonOther: iTunesAppCOM.idl
- ITCOMDisabledReasonQuitting: iTunesAppCOM.idl
- ITConvertOperationComplete : iTunesAppCOM.idl
- ITConvertOperationStatusChanged: iTunesAppCOM.idl
- ITEventAboutToPromptUserToQuit: iTunesAppCOM.idl
- ITEventCOMCallsDisabled : iTunesAppCOM.idl
- ITEventCOMCallsEnabled : iTunesAppCOM.idl
- ITEventDatabaseChanged : iTunesAppCOM.idl
- ITEventPlayerPlay : iTunesAppCOM.idl
- ITEventPlayerPlayingTrackChanged : iTunesAppCOM.idl
- ITEventPlayerStop : iTunesAppCOM.idl
- ITEventQuitting: iTunesAppCOM.idl
- ITEventSoundVolumeChanged : iTunesAppCOM.idl
- ITPlayButtonStatePauseDisabled : iTunesAppCOM.idl
- ITPlayButtonStatePauseEnabled : iTunesAppCOM.idl
- ITPlayButtonStatePlayDisabled : iTunesAppCOM.idl
- ITPlayButtonStatePlayEnabled : iTunesAppCOM.idl
- ITPlayButtonStateStopDisabled : iTunesAppCOM.idl
- ITPlayButtonStateStopEnabled : **iTunesAppCOM.idl**
- ITPlayerButtonModifierKeyAlt : iTunesAppCOM.idl
- ITPlayerButtonModifierKeyCapsLock : iTunesAppCOM.idl
- ITPlayerButtonModifierKeyControl: iTunesAppCOM.idl

- ITPlayerButtonModifierKeyNone : iTunesAppCOM.idl
- ITPlayerButtonModifierKeyShift : iTunesAppCOM.idl
- ITPlayerButtonNext : iTunesAppCOM.idl
- ITPlayerButtonPlay : iTunesAppCOM.idl
- ITPlayerButtonPrevious : iTunesAppCOM.idl
- ITPlayerStateFastForward : iTunesAppCOM.idl
- ITPlayerStatePlaying : iTunesAppCOM.idl
- ITPlayerStateRewind : iTunesAppCOM.idl
- ITPlayerStateStopped : iTunesAppCOM.idl
- ITPlaylistKindCD: iTunesPlaylistCOM.idl
- ITPlaylistKindDevice : iTunesPlaylistCOM.idl
- ITPlaylistKindLibrary : iTunesPlaylistCOM.idl
- ITPlaylistKindRadioTuner : iTunesPlaylistCOM.idl
- ITPlaylistKindUnknown: iTunesPlaylistCOM.idl
- ITPlaylistKindUser : iTunesPlaylistCOM.idl
- ITPlaylistPrintKindAlbumlist : iTunesPlaylistCOM.idl
- ITPlaylistPrintKindInsert : iTunesPlaylistCOM.idl
- ITPlaylistPrintKindPlaylist: iTunesPlaylistCOM.idl
- ITPlaylistRepeatModeAll: iTunesPlaylistCOM.idl
- ITPlaylistRepeatModeOff: iTunesPlaylistCOM.idl
- ITPlaylistRepeatModeOne : iTunesPlaylistCOM.idl
- ITPlaylistSearchFieldAlbums : **iTunesPlaylistCOM.idl**
- ITPlaylistSearchFieldAll: iTunesPlaylistCOM.idl
- ITPlaylistSearchFieldArtists : iTunesPlaylistCOM.idl
- ITPlaylistSearchFieldComposers : iTunesPlaylistCOM.idl
- ITPlaylistSearchFieldSongNames : iTunesPlaylistCOM.idl
- ITPlaylistSearchFieldVisible : iTunesPlaylistCOM.idl
- ITRatingKindComputed : iTunesTrackCOM.idl
- ITRatingKindUser: iTunesTrackCOM.idl
- ITSourceKindAudioCD : iTunesSourceCOM.idl
- ITSourceKindDevice : iTunesSourceCOM.idl
- ITSourceKindIPod : iTunesSourceCOM.idI
- ITSourceKindLibrary : iTunesSourceCOM.idl
- ITSourceKindMP3CD: iTunesSourceCOM.idl
- ITSourceKindRadioTuner : iTunesSourceCOM.idl
- ITSourceKindSharedLibrary: iTunesSourceCOM.idl
- ITSourceKindUnknown: iTunesSourceCOM.idl

- ITTrackKindCD: iTunesTrackCOM.idl
- ITTrackKindDevice : iTunesTrackCOM.idl
- ITTrackKindFile: iTunesTrackCOM.idl
- ITTrackKindSharedLibrary : iTunesTrackCOM.idl
- ITTrackKindUnknown: iTunesTrackCOM.idl
- ITTrackKindURL: iTunesTrackCOM.idl
- ITUNES_E_CONVERSIONINPROGRESS:
 iTunesCOMInterface.idl
- ITUNES_E_MUSICSTOREDISABLED: iTunesCOMInterface.idl
- ITUNES E OBJECTDELETED : iTunesCOMInterface.idl
- ITUNES E OBJECTEXISTS: iTunesCOMInterface.idl
- ITUNES E OBJECTLOCKED: iTunesCOMInterface.idl
- ITUNES E PODCASTSDISABLED: iTunesCOMInterface.idl
- ITUNES E USERCANCEL: iTunesCOMInterface.idl
- ITUserPlaylistSpecialKindAudiobooks : iTunesPlaylistCOM.idl
- ITUserPlaylistSpecialKindFolder: iTunesPlaylistCOM.idl
- ITUserPlaylistSpecialKindMovies : iTunesPlaylistCOM.idl
- ITUserPlaylistSpecialKindMusic: iTunesPlaylistCOM.idl
- ITUserPlaylistSpecialKindNone : iTunesPlaylistCOM.idl
- ITUserPlaylistSpecialKindPartyShuffle : iTunesPlaylistCOM.idl
- ITUserPlaylistSpecialKindPodcasts : iTunesPlaylistCOM.idl
- ITUserPlaylistSpecialKindPurchasedMusic : iTunesPlaylistCOM.idl
- ITUserPlaylistSpecialKindTVShows: iTunesPlaylistCOM.idl
- ITUserPlaylistSpecialKindVideos : iTunesPlaylistCOM.idl
- ITVideoKindMovie : iTunesTrackCOM.idl
- ITVideoKindMusicVideo : iTunesTrackCOM.idl
- ITVideoKindNone : iTunesTrackCOM.idl
- ITVideoKindTVShow : iTunesTrackCOM.idl
- ITVisualSizeLarge : iTunesAppCOM.idl
- ITVisualSizeMedium : iTunesAppCOM.idl
- ITVisualSizeSmall : iTunesAppCOM.idl
- ITWindowKindArtwork : iTunesWindowCOM.idl
- ITWindowKindBrowser : iTunesWindowCOM.idl
- ITWindowKindEQ: iTunesWindowCOM.idl
- ITWindowKindNowPlaying : iTunesWindowCOM.idl
- ITWindowKindPlaylist : iTunesWindowCOM.idl
- ITWindowKindUnknown: iTunesWindowCOM.idl

- k -

- kITTypeLibrary_MajorVersion : iTunesCOMInterface.idl
- kITTypeLibrary_MinorVersion : iTunesCOMInterface.idl

All | Functions

• importlib(): iTunesLib