```
- (SBElementArray<QuickTimePlayerWindow *> *) windows;
- (SBElementArray<QuickTimePlayerDocument *> *) documents;
@interface QuickTimePlayerApplication : SBApplication
// The application's top-level scripting object.
*\
* Standard Suite
\*
@end
- (void) moveTo:(SBObject *)to; // Move an object to a new location.
Copy an object.
- (void) duplicateTo:(SBObject *)to withProperties:(NSDictionary *)withProperties; //
- (void) delete; // Delete an object.
(BOOL)printDialog; // Print a document.
- (void) printWithProperties: (NSDictionary *)withProperties printDialog:
- (void) saveIn:(NSURL *)in_ as:(id)as; // Save a document.
Close a document.
- (void) closeSaving:(QuickTimePlayerSaveOptions)saving savingIn:(NSURL *)savingIn; //
@protocol QuickTimePlayerGenericMethods
typedef enum QuickTimePlayerPrintingErrorHandling QuickTimePlayerPrintingErrorHandling;
PostScript errors */
    QuickTimePlayerPrintingErrorHandlingDetailed = 'lwdt' /* print a detailed report of
handling */,
    QuickTimePlayerPrintingErrorHandlingStandard = 'lwst' /* Standard PostScript error
enum QuickTimePlayerPrintingErrorHandling {
typedef enum QuickTimePlayerSaveOptions QuickTimePlayerSaveOptions;
};
file. */
    QuickTimePlayerSaveOptionsAsk = 'ask ' /* Ask the user whether or not to save the
    QuickTimePlayerSaveOptionsNo = 'no ' /* Do not save the file. */,
    QuickTimePlayerSaveOptionsYes = 'yes ' /* Save the file. */,
enum QuickTimePlayerSaveOptions {
QuickTimePlayerScreenCompressionPreset;
QuickTimePlayerAudioCompressionPreset, QuickTimePlayerMovieCompressionPreset,
QuickTimePlayerVideoRecordingDevice, QuickTimePlayerAudioRecordingDevice,
@class QuickTimePlayerApplication, QuickTimePlayerDocument, QuickTimePlayerWindow,
#import <ScriptingBridge/ScriptingBridge.h>
#import <AppKit/AppKit.h>
*\
 * QuickTimePlayer.h
\*
```

```
@property (copy, readonly) NSString *name; // The name of the application. @property (readonly) BOOL frontmost; // Is this the active application? @property (copy, readonly) NSString *version; // The version number of the application.
```

- (id) open:(id)x; // Open a document.
- (void) print:(id)x withProperties:(NSDictionary *)withProperties printDialog:
 (B00L)printDialog; // Print a document.
- (void) quitSaving:(QuickTimePlayerSaveOptions)saving; // Quit the application.
- (BOOL) exists:(id)x; // Verify that an object exists.
- (void) openURL:(NSString *)x; // Open a URL.
- (QuickTimePlayerDocument *) newMovieRecording NS_RETURNS_NOT_RETAINED; // Create a new movie recording document.
- (QuickTimePlayerDocument *) newAudioRecording NS_RETURNS_NOT_RETAINED; // Create a new audio recording document.
- (void) newScreenRecording NS_RETURNS_NOT_RETAINED; // Create a new screen recording document.

@end

// A document.
@interface QuickTimePlayerDocument : SBObject <QuickTimePlayerGenericMethods>

@property (copy, readonly) NSString *name; // Its name.
@property (readonly) BOOL modified; // Has it been modified since the last save?
@property (copy, readonly) NSURL *file; // Its location on disk, if it has one.

- (void) play; // Play the movie.
- (void) start; // Start the movie recording.
- (void) pause; // Pause the recording.
- (void) resume; // Resume the recording.
- (void) stop; // Stop the movie or recording.
- (void) stepBackwardBy: (NSInteger)by; // Step the movie backward the specified number
 of steps (default is 1).
- (void) stepForwardBy:(NSInteger)by; // Step the movie forward the specified number of steps (default is 1).
- (void) trimFrom:(double)from to:(double)to; // Trim the movie.
- (void) present; // Present the document full screen.
- (void) exportIn:(NSURL *)in_ usingSettingsPreset; // Export a movie to another file

@end

// A window.
@interface QuickTimePlayerWindow : SBObject <QuickTimePlayerGenericMethods>

@property (copy, readonly) NSString *name; // The title of the window.
- (NSInteger) id; // The unique identifier of the window.

@property NSInteger index; // The index of the window, ordered front to back.

@property NSRect bounds; // The bounding rectangle of the window.

@property (readonly) BOOL closeable; // Does the window have a close button?

@property (readonly) BOOL miniaturizable; // Does the window have a minimize button?

@property BOOL miniaturized; // Is the window minimized right now?

@property (readonly) BOOL resizable; // Can the window be resized?
@property BOOL visible; // Is the window visible right now?

@property (readonly) BOOL zoomable; // Does the window have a zoom button?

```
<QuickTimePlayerGenericMethods>
@interface QuickTimePlayerMovieCompressionPreset : SBObject
// A movie recording compression preset
@end
- (NSString *) id; // The unique identifier of the preset.
@property (copy, readonly) NSString *name; // The name of the preset.
<QuickTimePlayerGenericMethods>
@interface QuickTimePlayerAudioCompressionPreset : SBObject
// An audio recording compression preset
@end
- (NSString *) id; // The unique identifier of the device.
@property (copy, readonly) NSString *name; // The name of the device.
@interface QuickTimePlayerAudioRecordingDevice : SBObject <QuickTimePlayerGenericMethods>
// An audio recording device
@end
- (NSString *) id; // The unique identifier of the device.
@property (copy, readonly) NSString *name; // The name of the device.
@interface QuickTimePlayerVideoRecordingDevice : SBObject <QuickTimePlayerGenericMethods>
// A video recording device
*\
 * QuickTime Player Suite
\*
@end
@interface QuickTimePlayerApplication (InternetSuite)
 * Internet Suite
\*
```

@property (copy, readonly) QuickTimePlayerDocument *document; // The document whose

@end

contents are displayed in the window.

@property BOOL zoomed; // Is the window zoomed right now?

- (NSString *) id; // The unique identifier of the preset. @property (copy, readonly) NSString *name; // The name of the preset.

@end

```
<QuickTimePlayerGenericMethods>
@interface QuickTimePlayerScreenCompressionPreset : SBObject
// A screen recording compression preset
```

- (NSString *) id; // The unique identifier of the preset. @property (copy, readonly) NSString *name; // The name of the preset.

@end

@interface QuickTimePlayerApplication (QuickTimePlayerSuite)

- (SBElementArray<QuickTimePlayerVideoRecordingDevice *> *) videoRecordingDevices;
- (SBElementArray<QuickTimePlayerAudioRecordingDevice *> *) audioRecordingDevices;
- (SBElementArray<QuickTimePlayerAudioCompressionPreset *> *) audioCompressionPresets;
- (SBElementArray<QuickTimePlayerMovieCompressionPreset *> *) movieCompressionPresets;
- (SBElementArray<QuickTimePlayerScreenCompressionPreset *> *) screenCompressionPresets;

@end

@interface QuickTimePlayerDocument (QuickTimePlayerSuite)

```
@property BOOL muted; // Is the movie muted?
@property BOOL looping; // Is the movie playing in a loop?
@property (readonly) double duration; // The duration of the movie in seconds.
@property (readonly) NSInteger dataSize; // The data size of the movie in bytes.
second.
@property (readonly) NSInteger dataRate; // The data rate of the movie in bytes per
@property double currentTime; // The current time of the movie in seconds.
@property double audioVolume; \ //\  The volume of the movie from 0 to 1, where 1 is 100%.
```

@property (readonly) NSPoint naturalDimensions; // The natural dimensions of the movie.

@property (readonly) BOOL playing; // Is the movie playing?

@property double rate; // The current rate of the movie.

@property BOOL presenting; // Is the movie presented in full screen?

@property (copy) QuickTimePlayerAudioRecordingDevice *currentMicrophone; // The

currently previewing audio device.

previewing video device. @property (copy) QuickTimePlayerVideoRecordingDevice *currentCamera; // The currently

current audio compression preset. @property (copy) QuickTimePlayerAudioCompressionPreset *currentAudioCompression;

current movie compression preset. @property (copy) QuickTimePlayerMovieCompressionPreset *currentMovieCompression; // The

The current screen compression preset. @property (copy) QuickTimePlayerScreenCompressionPreset *currentScreenCompression; //

@end