```
\*
 * Shortcuts.h
 \*

#import <AppKit\AppKit.h>
#import <Scripting\Scripting.h>


@class ShortcutsEvents, ShortcutsEventsApplication, ShortcutsEventsDocument, ShortcutsEventsWindow,
ShortcutsEventsShortcut, ShortcutsEventsTrigger;

enum ShortcutsEventsSavingOptions {
    ShortcutsEventsSavingOptionsYes = 'yes ' \* Save the file. *\,
    ShortcutsEventsSavingOptionsNo = 'no  ' \* Do not save the file. *\,
    ShortcutsEventsSavingOptionsAsk = 'ask ' \* Ask the user whether or not to save the
file. *\
};
typedef enum ShortcutsEventsSavingOptions ShortcutsEventsSavingOptions;

enum ShortcutsEventsPrintingErrorHandling {
    ShortcutsEventsPrintingErrorHandlingStandard = 'lwst' \* Standard PostScript error
handling *\,
    ShortcutsEventsPrintingErrorHandlingDetailed = 'lwdt' \* Print a detailed report of
PostScript errors *\
};
typedef enum ShortcutsEventsPrintingErrorHandling ShortcutsEventsPrintingErrorHandling;

@protocol ShortcutsEventsGenericMethods

- (void) closeSaving:(ShortcutsEventsSavingOptions)saving savingIn:(NSURL *)savingIn;  \\
Close a document.
- (void) saveIn:(NSURL *)in_ as:(id)as;  \\ Save a document.
- (void) printWithProperties:(NSDictionary *)withProperties printDialog:
(BOOL)printDialog;  \\ Print a document.
- (void) delete;  \\ Delete an object.
- (void) duplicateTo:(SBObject *)to withProperties:(NSDictionary *)withProperties;  \\
Copy an object.
- (void) moveTo:(SBObject *)to;  \\ Move an object to a new location.

@end

@protocol ShortcutsEventsApplication <SBApplicationProtocol>
- (SBElementArray<ShortcutsEventsDocument *> *) documents;
- (SBElementArray<ShortcutsEventsWindow *> *) windows;

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

@property (readonly) NSString *version;  \\ The version of the 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:(BOOL)printDialog;  // Print a document.
- (void) quitSaving:(ShortcutsEventsSaveOptions)saving;  // Quit the application.
- (BOOL) exists:(id)x;  // Verify that an object exists.

@end

// A document.
@interface ShortcutsEventsDocument : SBObject <ShortcutsEventsGenericMethods>

@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.


@end

// A window.
@interface ShortcutsEventsWindow : SBObject <ShortcutsEventsGenericMethods>

@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?
@property BOOL zoomed;  // Is the window zoomed right now?
@property (copy, readonly) ShortcutsEventsDocument *document;  // The document whose
contents are displayed in the window.


@end



/*
 * Shortcuts Suite
 */

@interface ShortcutsEventsApplication (ShortcutsSuite)

- (SBElementArray<ShortcutsEventsShortcut *> *) shortcuts;
- (SBElementArray<ShortcutsEventsFolder *> *) folders;

@end

// a shortcut in the Shortcuts application
@interface ShortcutsEventsShortcut : SBObject <ShortcutsEventsGenericMethods>
```

```
@property (copy, readonly) NSString *name;  // the name of the shortcut
@property (copy, readonly) NSString *subtitle;  // the shortcut's subtitle
- (NSString *) id;  // the unique identifier of the shortcut
@property (copy) ShortcutsEventsFolder *folder;  // the folder containing this shortcut
@property (copy, readonly) NSColor *color;  // the shortcut's color
@property (readonly) BOOL acceptsInput;  // indicates whether or not the shortcut accepts input data
@property (readonly) NSInteger actionCount;  // the number of actions in the shortcut

- (id) runWithInput:(id)withInput;  // Run a shortcut. To run a shortcut in the background, without opening the Shortcuts app, tell 'Shortcuts Events' instead of 'Shortcuts'.

@end

// a folder containing shortcuts
@interface ShortcutsEventsFolder : SBObject <ShortcutsEventsGenericMethods>

- (SBElementArray<ShortcutsEventsShortcut *> *) shortcuts;

@property (copy) NSString *name;  // the name of the folder
- (NSString *) id;  // the unique identifier of the folder


@end
```