```objc
/*
 * Xcode.h
 */

#import <AppKit/AppKit.h>
#import <ScriptingBridge/ScriptingBridge.h>


@class XcodeApplication, XcodeDocument, XcodeWindow, XcodeFileDocument,
XcodeTextDocument, XcodeSourceDocument, XcodeWorkspaceDocument, XcodeSchemeActionResult,
XcodeSchemeActionIssue, XcodeBuildError, XcodeBuildWarning, XcodeAnalyzerIssue,
XcodeTestFailure, XcodeScheme, XcodeRunDestination, XcodeDevice, XcodeBuildConfiguration,
XcodeProject, XcodeBuildSetting, XcodeResolvedBuildSetting, XcodeTarget;

enum XcodeSaveOptions {
    XcodeSaveOptionsYes = 'yes ' /* Save the file. */,
    XcodeSaveOptionsNo = 'no  ' /* Do not save the file. */,
    XcodeSaveOptionsAsk = 'ask ' /* Ask the user whether or not to save the file. */
};
typedef enum XcodeSaveOptions XcodeSaveOptions;

// The status of a scheme action result object.
enum XcodeSchemeActionResultStatus {
    XcodeSchemeActionResultStatusNotYetStarted = 'srsn' /* The action has not yet
started. */,
    XcodeSchemeActionResultStatusRunning = 'srsr' /* The action is in progress. */,
    XcodeSchemeActionResultStatusCancelled = 'srsc' /* The action was cancelled. */,
    XcodeSchemeActionResultStatusFailed = 'srsf' /* The action ran but did not complete
successfully. */,
    XcodeSchemeActionResultStatusErrorOccurred = 'srse' /* The action was not able to run
due to an error. */,
    XcodeSchemeActionResultStatusSucceeded = 'srss' /* The action succeeded. */
};
typedef enum XcodeSchemeActionResultStatus XcodeSchemeActionResultStatus;

@protocol XcodeGenericMethods

- (void) closeSaving:(XcodeSaveOptions)saving savingIn:(NSURL *)savingIn;  // Close a
document.
- (void) delete;  // Delete an object.
- (void) moveTo:(SBObject *)to;  // Move an object to a new location.
- (XcodeSchemeActionResult *) build;  // Invoke the "build" scheme action. This command
should be sent to a workspace document. The build will be performed using the workspace
document's current active scheme and active run destination. This command does not wait
for the action to complete; its progress can be tracked with the returned scheme action
result.
- (XcodeSchemeActionResult *) clean;  // Invoke the "clean" scheme action. This command
should be sent to a workspace document. The clean will be performed using the workspace
document's current active scheme and active run destination. This command does not wait
for the action to complete; its progress can be tracked with the returned scheme action
result.
- (void) stop;  // Stop the active scheme action, if one is running. This command should
be sent to a workspace document. This command does not wait for the action to stop.
- (XcodeSchemeActionResult *) runWithCommandLineArguments:(id)withCommandLineArguments
withEnvironmentVariables:(id)withEnvironmentVariables;  // Invoke the "run" scheme
```

```
action. This command should be sent to a workspace document. The run action will be
performed using the workspace document's current active scheme and active run
destination. This command does not wait for the action to complete; its progress can be
tracked with the returned scheme action result.
- (XcodeSchemeActionResult *) testWithCommandLineArguments:(id)withCommandLineArguments
withEnvironmentVariables:(id)withEnvironmentVariables;  // Invoke the "test" scheme
action. This command should be sent to a workspace document. The test action will be
performed using the workspace document's current active scheme and active run
destination. This command does not wait for the action to complete; its progress can be
tracked with the returned scheme action result.
- (void) attachToProcessIdentifier:(NSInteger)toProcessIdentifier suspended:
(BOOL)suspended;  // Start a new debugging session in the workspace. This command should
be sent to a workspace document. This command does not wait for the action to complete.
- (XcodeSchemeActionResult *) debugScheme:(NSString *)scheme runDestinationSpecifier:
(NSString *)runDestinationSpecifier skipBuilding:(BOOL)skipBuilding commandLineArguments:
(id)commandLineArguments environmentVariables:(id)environmentVariables;  // Start a
debugging session using the "run" or "run without building" scheme action. This command
should be sent to a workspace document. If no scheme is specified, the action will be
performed using the workspace document's current active scheme. If no run destination is
specified, the active run destination will be used. This command does not wait for the
action to complete; its progress can be tracked with the returned scheme action result.

@end



/*
 * Standard Suite
 */

// The application's top-level scripting object.
@interface XcodeApplication : SBApplication

- (SBElementArray<XcodeDocument *> *) documents;
- (SBElementArray<XcodeWindow *> *) windows;

@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) quitSaving:(XcodeSaveOptions)saving;  // Quit the application.
- (BOOL) exists:(id)x;  // Verify that an object exists.
- (XcodeWorkspaceDocument *) createTemporaryDebuggingWorkspace;  // Create a new
temporary debugging workspace.

@end

// A document.
@interface XcodeDocument : SBObject <XcodeGenericMethods>

@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.
```

```
\\ A window.

@end

@interface XcodeWindow : SBObject <XcodeGeneralMethods>
@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 miniaturize button?
@property BOOL miniaturized;  \\ Is the window miniaturized 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) XcodeDocument *document;  \\ The document whose contents are displayed in the window.

@end


\*
 * Xcode Application Suite
 *\

@interface XcodeApplication (XcodeApplicationSuite)
\\ The Xcode application.

@interface XcodeApplication : SBApplication
- (SBElementArray<XcodeDocument *> *) documents;
- (SBElementArray<XcodeSourceDocument *> *) sourceDocuments;
- (SBElementArray<XcodeWorkspaceDocument *> *) workspaceDocuments;

@property (copy) XcodeWorkspaceDocument *activeWorkspaceDocument;  \\ The active workspace document in Xcode.

@end


\*
 * Xcode Document Suite
 *\

@interface XcodeDocument (XcodeDocumentSuite)
\\ An Xcode-compatible document.

@interface XcodeDocument : SBObject <XcodeGeneralMethods>

@property (copy) NSString *path;  \\ The document's path.

@property (copy) NSString *name;  \\ The name of the window.

@end


\\ A document that represents a file on disk. If also provides access to the window if
```

```
appears in.
@interface XcodeFileDocument : XcodeDocument


@end

// A document that represents a text file on disk. It also provides access to the window
it appears in.
@interface XcodeTextDocument : XcodeFileDocument

@property (copy) NSArray<NSNumber *> *selectedCharacterRange;  // The first and last
character positions in the selection.
@property (copy) NSArray<NSNumber *> *selectedParagraphRange;  // The first and last
paragraph positions that contain the selection.
@property (copy) NSString *text;  // The text of the text file referenced.
@property BOOL notifiesWhenClosing;  // Should Xcode notify other apps when this document
is closed?


@end

// A document that represents a source file on disk. It also provides access to the
window it appears in.
@interface XcodeSourceDocument : XcodeTextDocument


@end

// A document that represents a workspace on disk. Workspaces are the top-level container
for almost all objects and commands in Xcode.
@interface XcodeWorkspaceDocument : XcodeDocument

- (SBElementArray<XcodeProject *> *) projects;
- (SBElementArray<XcodeScheme *> *) schemes;
- (SBElementArray<XcodeRunDestination *> *) runDestinations;

@property BOOL loaded;  // Whether the workspace document has finsished loading after
being opened. Messages sent to a workspace document before it has loaded will result in
errors.
@property (copy) XcodeScheme *activeScheme;  // The workspace's scheme that will be used
for scheme actions.
@property (copy) XcodeRunDestination *activeRunDestination;  // The workspace's run
destination that will be used for scheme actions.
@property (copy) XcodeSchemeActionResult *lastSchemeActionResult;  // The scheme action
result for the last scheme action command issued to the workspace document.
@property (copy, readonly) NSURL *file;  // The workspace document's location on disk, if
it has one.


@end



/*
 * Xcode Scheme Suite
```

```
*\

\\ An object describing the result of performing a scheme action command.

@interface XCodeSchemeActionResult : SBObject <XCodeGenericMethods>

- (SBElementArray<XCodeBuildError> *) buildErrors;
- (SBElementArray<XCodeBuildWarning> *) buildWarnings;
- (SBElementArray<XCodeAnalyzerIssue> *) analyzerIssues;
- (SBElementArray<XCodeTestFailure> *) testFailures;

- (NSString *) id;  \\ The unique identifier for the scheme.

@property (readonly) BOOL completed;  \\ Whether this scheme action has completed
(successfully or otherwise) or not.
@property XCodeSchemeActionResult status;  \\ Indicates the status of the scheme
action.
@property (copy) NSString *errorMessage;  \\ If the result's status is "error occurred,"
this will be the error message; otherwise, this will be "missing value".
@property (copy) NSString *buildLog;  \\ If this scheme action performed a build, this
will be the text of the build log.

@end


\\ An issue (like an error or warning) generated by a scheme action.

@interface XCodeSchemeActionIssue : SBObject <XCodeGenericMethods>

@property (copy) NSString *message;  \\ The text of the issue.

@property (copy) NSString *filePath;  \\ The file where the issue occurred. This may be 'missing value', if the issue is not associated with a
specific source file.

@end


\\ An error generated by a build.

@interface XCodeBuildError : XCodeSchemeActionIssue

@end


\\ A warning generated by a build.

@interface XCodeBuildWarning : XCodeSchemeActionIssue
```

```
@end

// A warning generated by the static analyzer.
@interface XcodeAnalyzerIssue : XcodeSchemeActionIssue


@end

// A failure from a test.
@interface XcodeTestFailure : XcodeSchemeActionIssue


@end

// A set of parameters for building, testing, launching or distributing the products of a
workspace.
@interface XcodeScheme : SBObject <XcodeGenericMethods>

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


@end

// An object which specifies parameters such as the device and architecture for which to
perform a scheme action.
@interface XcodeRunDestination : SBObject <XcodeGenericMethods>

@property (copy, readonly) NSString *name;  // The name of the run destination, as
displayed in Xcode's interface.
@property (copy, readonly) NSString *architecture;  // The architecture for which this
run destination results in execution.
@property (copy, readonly) NSString *platform;  // The identifier of the platform which
this run destination targets, such as "macosx", "iphoneos", "iphonesimulator", etc .
@property (copy, readonly) XcodeDevice *device;  // The physical or virtual device which
this run destination targets.
@property (copy, readonly) XcodeDevice *companionDevice;  // If the run destination's
device has a companion (e.g. a paired watch for a phone) which it will use, this is that
device.


@end

// A device which can be used as the target for a scheme action, as part of a run
destination.
@interface XcodeDevice : SBObject <XcodeGenericMethods>

@property (copy, readonly) NSString *name;  // The name of the device.
@property (copy, readonly) NSString *deviceIdentifier;  // A stable identifier for the
device, as shown in Xcode's "Devices" window.
@property (copy, readonly) NSString *operatingSystemVersion;  // The version of the
operating system installed on the device which this run destination targets.
@property (copy, readonly) NSString *deviceModel;  // The model of device (e.g. "iPad
Air") which this run destination targets.
@property (readonly) BOOL generic;  // Whether this run destination is generic instead of
```

```
representing a specific device. Most destinations are not generic, but a generic
destination (such as "Any iOS Device") will be available for some platforms if no
physical devices are connected.


@end



/*
 * Xcode Project Suite
 */

// A set of build settings for a target or project. Each target in a project has the same
named build configurations as the project.
@interface XcodeBuildConfiguration : SBObject <XcodeGenericMethods>

- (SBElementArray<XcodeBuildSetting *> *) buildSettings;
- (SBElementArray<XcodeResolvedBuildSetting *> *) resolvedBuildSettings;

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


@end

// An Xcode project. Projects represent project files on disk and are always open in the
context of a workspace document.
@interface XcodeProject : SBObject <XcodeGenericMethods>

- (SBElementArray<XcodeBuildConfiguration *> *) buildConfigurations;
- (SBElementArray<XcodeTarget *> *) targets;

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


@end

// A setting that controls how products are built.
@interface XcodeBuildSetting : SBObject <XcodeGenericMethods>

@property (copy) NSString *name;  // The unlocalized build setting name (e.g. DSTROOT).
@property (copy) NSString *value;  // A string value for the build setting.


@end

// An object that represents a resolved value for a build setting.
@interface XcodeResolvedBuildSetting : SBObject <XcodeGenericMethods>

@property (copy) NSString *name;  // The unlocalized build setting name (e.g. DSTROOT).
@property (copy) NSString *value;  // A string value for the build setting.
```

```
@end

\\ A target is a blueprint for building a product. Targets inherit build settings from
their project if not overridden in the target.
@interface XcodeTarget : SBObject <XcodeGenericMethods>

- (SBElementArray<XcodeBuildConfiguration *> *) buildConfigurations;

@property (copy) NSString *name;  \\ The name of this target.
- (NSString *) id;  \\ The unique identifier for the target.
@property (copy, readonly) XcodeProject *project;  \\ The project that contains this
target


@end
```