```
KeynoteMovieExportFormatsFormat720p = 'Kmf7' /* 720p */,
    KeynoteMovieExportFormatsFormat540p = 'Kmf5' /* 540p */,
    KeynoteMovieExportFormatsFormat360p = 'Kmf3' /* 360p */,
enum KeynoteMovieExportFormats {
typedef enum KeynoteImageExportFormats KeynoteImageExportFormats;
}:
    KeynoteImageExportFormatsTIFF = 'Kift' /* TIFF */
    KeynoteImageExportFormatsPNG = 'Kifp' /* PNG */,
    KeynoteImageExportFormatsJPEG = 'Kifj' /* JPEG */,
enum KeynoteImageExportFormats {
typedef enum KeynoteExportFormat KeynoteExportFormat;
};
    KeynoteExportFormatKeynote09 = 'Kkey' /* Keynote 09 */
    KeynoteExportFormatMicrosoftPowerPoint = 'Kppt' /* Microsoft PowerPoint */,
    KeynoteExportFormatSlideImages = 'Kimg' /* image */,
    KeynoteExportFormatPDF = 'Kpdf' /* PDF */,
    KeynoteExportFormatQuickTimeMovie = 'Kmov' /* QuickTime movie */,
    KeynoteExportFormatHTML = 'Khtm' /* HTML */,
enum KeynoteExportFormat {
typedef enum KeynoteSaveableFileFormat KeynoteSaveableFileFormat;
};
    KeynoteSaveableFileFormatKeynote = 'Knff' /* The Keynote native file format */
enum KeynoteSaveableFileFormat {
typedef enum KeynotePrintingErrorHandling KeynotePrintingErrorHandling;
PostScript errors */
    KeynotePrintingErrorHandlingDetailed = 'lwdt' /* print a detailed report of
    KeynotePrintingErrorHandlingStandard = 'lwst' /* Standard PostScript error handling
enum KeynotePrintingErrorHandling {
typedef enum KeynoteSaveOptions KeynoteSaveOptions;
};
    KeynoteSaveOptionsAsk = 'ask ' /* Ask the user whether or not to save the file. */
    KeynoteSaveOptionsNo = 'no ' /* Do not save the file. */,
    KeynoteSaveOptionsYes = 'yes ' /* Save the file, */,
enum KeynoteSaveOptions {
KeynoteRange, KeynoteCell, KeynoteRow, KeynoteColumn;
KeynoteImage, KeynoteGroup, KeynoteLine, KeynoteMovie, KeynoteTable, KeynoteTextItem,
KeynoteSlideLayout, KeynoteIWorkItem, KeynoteAudioClip, KeynoteShape, KeynoteChart,
KeynoteCharacter, KeynoteParagraph, KeynoteWord, KeynoteIWorkContainer, KeynoteSlide,
@class KeynoteApplication, KeynoteDocument, KeynoteWindow, KeynoteTheme, KeynoteRichText,
#import <ScriptingBridge/ScriptingBridge.h>
#import <AppKit/AppKit.h>
*\
 * Keynote.h
\*
```

```
KeynoteTransitionEffectsObjectPush = 'toph' /* */,
    KeynoteTransitionEffectsObjectPop = 'topp' /* */,
    KeynoteTransitionEffectsObjectFlip = 'tofp' /* */,
    KeynoteTransitionEffectsObjectCube = 'tocb' /* */,
    KeynoteTransitionEffectsSwing = 'tswg' /* */,
    KeynoteTransitionEffectsSparkle = 'tspk' /* */,
    KeynoteTransitionEffectsShimmer = 'tshm' /* */,
    KeynoteTransitionEffectsMagicMove = 'tmjv' /* */,
    KeynoteTransitionEffectsNoTransitionEffect = 'tnil' /* */,
enum KeynoteTransitionEffects {
typedef enum KeynotePDFImageQuality KeynotePDFImageQuality;
}:
    KeynotePDFImageQualityBest = 'KnP2' /* best quality */
    KeynotePDFImageQualityBetter = 'KnP1' /* better quality */,
    KeynotePDFImageQualityGood = 'KnP0' /* good quality */,
enum KeynotePDFImageQuality {
typedet enum KeynotePrintWhat KeynotePrintWhat;
};
    KeynotePrintWhatHandouts = 'Kpwh' /* handouts */
    KeynotePrintWhatSlideWithNotes = 'Kpwn' /* slides with notes */,
    KeynotePrintWhatIndividualSlides = 'Kpwi' /* individual slides */,
enum KeynotePrintWhat {
typedef enum KeynoteMovieFramerates KeynoteMovieFramerates;
}:
    KeynoteMovieFrameratesFPS60 = 'Kfr9' /* 60 FPS */
    KeynoteMovieFrameratesFPS5994 = 'Kfr8' /* 59.94 FPS */,
    KeynoteMovieFrameratesFPS50 = 'Kfr7' /* 50 FPS */,
    KeynoteMovieFrameratesFPS30 = 'Kfr6' /* 30 FPS */,
    KeynoteMovieFrameratesFPS2997 = 'Kfr5' /* 29.97 FPS */,
    KeynoteMovieFrameratesFPS25 = 'Kfr4' /* 25 FPS */,
    KeynoteMovieFrameratesFPS24 = 'Kfr3' /* 24 FPS */,
    KeynoteMovieFrameratesFPS2398 = 'Kfr2' /* 23.98 FPS */,
    KeynoteMovieFrameratesFPS12 = 'Kfr1' /* 12 FPS */,
enum KeynoteMovieFramerates {
typedef enum KeynoteMovieCodecs KeynoteMovieCodecs;
}:
    KeynoteMovieCodecsHEVC = 'Kmc7' /* HEVC */
    KeynoteMovieCodecsAppleProRes422Proxy = 'Kmc6' /* Apple ProRes 422Proxy */,
    KeynoteMovieCodecsAppleProRes422HQ = 'Kmc5' /* Apple ProRes 422HQ */,
    KeynoteMovieCodecsAppleProRes422LT = 'Kmc4' /* Apple ProRes 422LT */,
    KeynoteMovieCodecsAppleProRes4444 = 'Kmc3' /* Apple ProRes 4444 */,
    KeynoteMovieCodecsAppleProRes422 = 'Kmc2' /* Apple ProRes 422 */,
    KeynoteMovieCodecsH264 = 'Kmc1' /* H.264 */,
enum KeynoteMovieCodecs {
typedef enum KeynoteMovieExportFormats KeynoteMovieExportFormats;
}:
dimensions as the document, up to 4096x2160 */
    KeynoteMovieExportFormatsNativeSize = 'KmfN' /* Exported movie will have the same
    KeynoteMovieExportFormatsFormat2160p = 'Kmf4' /* DCI 4K (4096x2160) */,
    KeynoteMovieExportFormatsFormat1080p = 'Kmf8' /* 1080p */,
```

```
KeynoteNMSDAscending = 'ascn' /* Sort in increasing value order */,
enum KeynoteNMSD {
typedef enum KeynoteTAHT KeynoteTAHT;
   KeynoteTAHTRight = 'arit' /* Right-align content. */
   KeynoteTAHTLeft = 'alft' /* Left-align content. */,
   KeynoteTAHTJustify = 'ajst' /* Fully justify (left and right) content, */,
   KeynoteTAHTCenter = 'actr' /* Center-align content. */,
   KeynoteTAHTAutoAlign = 'aaut' /* Auto-align based on content type. */,
enum KeynoteTAHT {
typedef enum KeynoteTAVT KeynoteTAVT;
}:
   KeynoteTAVTTop = 'avtp' /* Top-align content. */
   KeynoteTAVTCenter = 'actr' /* Center-align content. */,
   KeynoteTAVTBottom = 'avbt' /* Right-align content. */,
enum KeynoteTAVT {
typedef enum KeynoteTransitionEffects KeynoteTransitionEffects;
   KeynoteTransitionEffectsFadeAndMove = 'tfad' /* */
   KeynoteTransitionEffectsTwist = 'ttwi' /* */,
   KeynoteTransitionEffectsTwirl = 'ttwl' /* */,
   KeynoteTransitionEffectsSwoosh = 'tsws' /* */,
   KeynoteTransitionEffectsSwap = 'tswp' /* */,
   KeynoteTransitionEffectsScale = 'tscl' /* */,
   KeynoteTransitionEffectsRevolvingDoor = 'trev' /* */,
   KeynoteTransitionEffectsReflection = 'trfl' /* */,
   KeynoteTransitionEffectsPivot = 'tpvt' /* */,
   KeynoteTransitionEffectsPageFlip = 'tpfl' /* */,
   KeynoteTransitionEffectsMosaic = 'tmsc' /* */,
   KeynoteTransitionEffectsFlop = 'tfop' /* */,
   KeynoteTransitionEffectsFlip = 'tfip' /* */,
   KeynoteTransitionEffectsFall = 'tfal' /* */,
   KeynoteTransitionEffectsDoorway = 'tdwy' /* */,
   KeynoteTransitionEffectsCube = 'tcub' /* */,
   KeynoteTransitionEffectsColorPlanes = 'tcpl' /* */,
   KeynoteTransitionEffectsBlinds = 'tbld' /* */,
   KeynoteTransitionEffectsWipe = 'twpe' /* */,
   KeynoteTransitionEffectsSwitch = 'tswi' /* */,
   KeynoteTransitionEffectsReveal = 'trvl' /* */,
   KeynoteTransitionEffectsPush = 'tpsh' /* */,
   KeynoteTransitionEffectsMoveIn = 'tmvi' /* */,
   KeynoteTransitionEffectsIris = 'tirs' /* */,
   KeynoteTransitionEffectsGrid = 'tgrd' /* */,
   KeynoteTransitionEffectsFadeThroughColor = 'tftc' /* */,
   KeynoteTransitionEffectsDroplet = 'tdpl' /* */,
   KeynoteTransitionEffectsDrop = 'tdrp' /* */,
   KeynoteTransitionEffectsDissolve = 'tdis' /* */,
   KeynoteTransitionEffectsConfetti = 'tcft' /* */,
   KeynoteTransitionEffectsClothesline = 'tclo' /* */,
   KeynoteTransitionEffectsPerspective = 'tprs' /* */,
   KeynoteTransitionEffectsObjectZoom = 'tozm' /* */,
   KeynoteTransitionEffectsObjectRevolve = 'torv' /* */,
```

```
KeynoteLegacyChartTypeStacked_vertical_bar_3d = 'svb3' /* three-dimensional stacked
chart */,
    KeynoteLegacyChartTypeVertical_bar_3d = 'vbr3' /* three-dimensional vertical bar
    KeynoteLegacyChartTypePie_3d = 'pie3' /* three-dimensional pie chart. */,
horizontal bar chart */,
    KeynoteLegacyChartTypeStacked_horizontal_bar_2d = 'shb2' /* two-dimensional stacked
chart */,
    KeynoteLegacyChartTypeHorizontal_bar_2d = 'hbr2' /* two-dimensional horizontal bar
vertical bar chart */,
    KeynoteLegacyChartTypeStacked_vertical_bar_2d = 'svb2' /* two-dimensional stacked
*/ '
    KeynoteLegacyChartTypeVertical_bar_2d = 'vbr2' /* two-dimensional vertical bar chart
    KeynoteLegacyChartTypePie_2d = 'pie2' /* two-dimensional pie chart */,
enum KeynoteLegacyChartType {
// Visual style of chart
typedef enum KeynotePlaybackRepetitionMethod KeynotePlaybackRepetitionMethod;
}:
    KeynotePlaybackRepetitionMethodLoopBackAndForth = 'mvbf' /* */
    KeynotePlaybackRepetitionMethodLoop = 'mvlp' /* */,
    KeynotePlaybackRepetitionMethodNone = 'mvrn' /* */,
enum KeynotePlaybackRepetitionMethod {
typedef enum KeynoteItemFillOptions KeynoteItemFillOptions;
};
    KeynoteItemFillOptionsAdvancedImageFill = 'fiai' /* */
    KeynoteItemFillOptionsImageFill = 'fiim' /* */,
    KeynoteItemFillOptionsAdvancedGradientFill = 'fiag' /* */,
    KeynoteItemFillOptionsGradientFill = 'figr' /* */,
    KeynoteItemFillOptionsColorFill = 'fico' /* */,
    KeynoteItemFillOptionsNoFill = 'fino' /* */,
enum KeynoteItemFillOptions {
typedef enum KeynoteNMCT KeynoteNMCT;
}:
    KeynoteNMCTNumeralSystem = 'fcns' /* Numeral System */
    KeynoteNMCTRating = 'frat' /* Rating format. (Numbers only) */,
    KeynoteNMCTDuration = 'fdur' /* Duration format */,
    KeynoteNMCTText = 'ctxt' /* Text format */,
    KeynoteNMCTStepper = 'fcst' /* Stepper control format (Numbers only) */,
    KeynoteNMCTSlider = 'fcsl' /* Slider control format (Numbers only) */,
    KeynoteNMCTScientific = 'fsci' /* Scientific notation format */,
    KeynoteNMCTPopUpMenu = 'fcpp' /* Pop-up menu control format (Numbers only) */,
    KeynoteNMCTPercent = 'fper' /* Percentage number format */,
    KeynoteNMCTNumber = 'nmbr' /* Decimal number format */,
    KeynoteNMCTFraction = 'ffra' /* Fraction number format */,
    KeynoteNMCTDateAndTime = 'fdtm' /* Date and time format */,
    KeynoteNMCTCurrency = 'fcur' /* Currency number format */,
    KeynoteNMCTCheckbox = 'fcch' /* Checkbox control format (Numbers only) */,
    KeynoteNMCTAutomatic = 'faut' /* Automatic format */,
enum KeynoteNMCT {
typedef enum KeynoteNMSD KeynoteNMSD;
}:
    KeynoteNMSDDescending = 'dscn' /* Sort in decreasing value order */
```

```
@property (readonly) BOOL frontmost; // Is this the active application?
@property (copy, readonly) NSString *name; // The name of the application.
- (SBElementArray<KeynoteWindow *> *) windows;
- (SBElementArray<KeynoteDocument *> *) documents;
@interface KeynoteApplication : SBApplication
// The application's top-level scripting object.
 *\
 * Standard Suite
\*
@end
- (void) delete; // Delete an object.
Copy an object.
- (void) duplicateTo:(SBObject *)to withProperties:(NSDictionary *)withProperties;
- (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:(KeynoteSaveableFileFormat)as; // Save a document.
document.
- (void) closeSaving:(KeynoteSaveOptions)saving savingIn:(NSURL *)savingIn; // Close a
@protocol KeynoteGenericMethods
typedef enum KeynoteLegacyChartGrouping KeynoteLegacyChartGrouping;
};
    KeynoteLegacyChartGroupingChartColumn = 'KCgc' /* group by column */
    KeynoteLegacyChartGroupingChartRow = 'KCgr' /* group by row */,
enum KeynoteLegacyChartGrouping {
// Grouping for chart data
typedef enum KeynoteLegacyChartType KeynoteLegacyChartType;
};
    KeynoteLegacyChartTypeScatterplot_2d = 'scp2' /* two-dimensional scatterplot chart */
chart */,
    KeynoteLegacyChartTypeStacked_area_3d = 'sar3' /* three-dimensional stacked area
    KeynoteLegacyChartTypeArea_3d = 'are3' /* three-dimensional area chart */,
    KeynoteLegacyChartTypeLine_3d = 'lin3' /* three-dimensional line chart */,
    KeynoteLegacyChartTypeLine_2d = 'lin2' /* two-dimensional line chart. */,
*/'
    KeynoteLegacyChartTypeStacked_area_2d = 'sar2' /* two-dimensional stacked area chart
    KeynoteLegacyChartTypeArea_2d = 'are2' /* two-dimensional area chart. */,
horizontal bar chart */,
    KeynoteLegacyChartTypeStacked_horizontal_bar_3d = 'shb3' /* three-dimensional stacked
chart */,
```

KeynoteLegacyChartTypeHorizontal_bar_3d = 'hbr3' /* three-dimensional horizontal bar

bar chart */,

@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: (KeynoteSaveOptions)saving; // Quit the application.
- (BOOL) exists:(id)x; // Verify that an object exists.
- (void) setPassword:(NSString *)x to:(KeynoteDocument *)to hint:(NSString *)hint savingInKeychain:(B00L)savingInKeychain; // Set a password to an unencrypted document. - (void) removePassword:(NSString *)x from:(KeynoteDocument *)from: // Remove the
- (void) removePassword:(NSString *)x from:(KeynoteDocument *)from; // Remove the
- password from the document.
- (void) showNext; // Advance one build or slide.
- (void) showPrevious; // Go to the previous slide.

@end

// A document.

```
@interface KeynoteDocument : SBObject <KeynoteGenericMethods>
@arceperty (eggy_readenly) NGString wasme. (/ Ite pame
```

@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) exportTo:(NSURL *)to as:(KeynoteExportFormat)as withProperties:(NSDictionary
 *)withProperties; // Export a slideshow to another file
- (void) startFrom: (KeynoteSlide *)from; // Start playing the presentation.
- (void) makeImageSlidesFiles:(NSArray<NSURL *> *)files setTitles:(B00L)setTitles slideLayout:(KeynoteSlideLayout *)slideLayout; // Make a series of slides from a list of files.
- (void) stop; // Stop the presentation.
- (void) showSlideSwitcher; // Show the slide switcher in play mode
- (void) hideSlideSwitcher; // Hide the slide switcher in play mode
- (void) moveSlideSwitcherForward; // Move the slide switcher forward one slide
- (void) moveSlideSwitcherBackward; // Move the slide switcher backward one slide
- (void) cancelSlideSwitcher; // Hide the slide switcher without changing slides
- (void) acceptSlideSwitcher; // Hide the slide switcher, going to the slide it has selected

@end

```
// A window.
@interface KeynoteWindow : SBObject <KeynoteGenericMethods>
```

@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) KeynoteDocument *document; // The document whose contents are

displayed in the window.

* iWork Text Suite

```
\*
@end
@property (copy, readonly) NSString *name;
- (NSString *) id; // The identifier used by the application.
@interface KeynoteTheme : SBObject <KeynoteGenericMethods>
// A collection of slide layouts, with shared design intents and elements.
@end
width = 1024. Wide slide width = 1920.
@property NSInteger width; // The width of the document (in points). Standard slide
height = 768. Wide slide height = 1080.
@property NSInteger height; // The height of the document (in points). Standard slide
slide that would display if the presentation was started.
@property (copy) KeynoteSlide *currentSlide; // The currently selected slide, or the
the specified time
@property NSInteger maximumIdleDuration; // Restart the slideshow if it's inactive for
time
@property BOOL autoRestart; // Restart the slideshow if it's inactive for the specified
@property BOOL autoPlay; // Automatically play the presentation when opening the file.
@property BOOL autoLoop; // Make the slideshow play repeatedly.
@property (copy) KeynoteTheme stdocumentTheme; // The theme assigned to the document.
@property BOOL slideNumbersShowing; // Are the slide numbers displayed?
- (NSString *) id; // Document ID.
- (SBElementArray<KeynoteSlideLayout *> *) slideLayouts;
- (SBElementArray<KeynoteSlide *> *) slides;
@interface KeynoteDocument (KeynoteSuite)
// The Keynote document.
@end
- (SBElementArray<KeynoteTheme *> *) themes;
@interface KeynoteApplication (KeynoteSuite)
// The Keynote application.
 * Keynote Suite
\*
```

```
// This provides the base rich text class for all iWork applications.
@interface KeynoteRichText : SBObject <KeynoteGenericMethods>
```

- (SBElementArray<KeynoteCharacter *> *) characters;
- (SBElementArray<KeynoteParagraph *> *) paragraphs;
- (SBElementArray<KeynoteWord *> *) words;

@property (copy) NSColor *color; // The color of the font. Expressed as an RGB value consisting of a list of three color values from 0 to 65535. ex: Blue = {0, 0, 65535}. @property (copy) NSString *font; // The name of the font. Can be the PostScript name, such as: "TimesNewRomanPS-ItalicMT", or display name: "Times New Roman Italic". TIP: Use the Font Book application get the information about a typeface. @property double size; // The size of the font.

```
@end
```

// One of some text's characters.
@interface KeynoteCharacter : KeynoteRichText

@end

 $\ensuremath{\mbox{//}}$ One of some text's paragraphs. Ginterface KeynoteParagraph : KeynoteRichText

- (SBElementArray<KeynoteCharacter *> *) characters;

- (SBElementArray<KeynoteWord *> *) words;

@end

// One of some text's words.
@interface KeynoteWord : KeynoteRichText

- (SBElementArray<KeynoteCharacter *> *) characters;

@end

```
/*
 * iWork Suite
 */
@interface KeynoteDocument (IWorkSuite)
```

@property (copy) NSArray<KeynotelWorkItem *> *selection; // A list of the currently
selected items.
@property (readonly) BOOL passwordProtected; // Whether the document is password
protected or not.

// A container for iWork items

```
@property (copy, readonly) NSString *name; // The name of the slide layout
@interface KeynoteSlideLayout : KeynoteSlide
// A slide layout in a theme or slideshow document
@end
(KeynoteLegacyChartGrouping)groupBy; // Add a chart to a slide
*> *)columnNames data:(NSArray<id> *)data type:(KeynoteLegacyChartType)type groupBy:

    (void) addChartRowNames:(NSArray<NSString *> *)rowNames columnNames:(NSArray<NSString</li>

to the slide.
@property (copy) NSDictionary *transitionProperties; // The transition settings to apply
@property (copy) KeynoteRichText *presenterNotes; // The presenter notes for the slide
container of the slide
@property (copy, readonly) KeynoteShape *defaultTitleItem; // The default title
of the slide
@property (copy, readonly) KeynoteShape *defaultBodyItem; // The default body container
@property BOOL titleShowing; // Is the default slide title displayed?
@property (readonly) NSInteger slideNumber; // index of the slide in the document
@property BOOL skipped; // Is the slide skipped?
@property BOOL bodyShowing; // Is the default body text displayed?
@property (copy) KeynoteSlideLayout *baseLayout; // The slide layout this slide is based
@interface KeynoteSlide : KeynoteIWorkContainer
// A slide in a slideshow document
*\
 * Keynote Suite
@end
- (SBElementArray<KeynoteTextItem *> *) textItems;
- (SBElementArray<KeynoteTable *> *) tables;
- (SBElementArray<KeynoteShape *> *) shapes;
- (SBElementArray<KeynoteMovie *> *) movies;
- (SBElementArray<KeynoteLine *> *) lines;
- (SBElementArray<KeynoteGroup *> *) groups;
- (SBElementArray<KeynoteIWorkItem *> *) iWorkItems;
- (SBElementArray<KeynoteImage *> *) images;
- (SBElementArray<KeynoteChart *> *) charts;
- (SBElementArray<KeynoteAudioClip *> *) audioClips;
@interface KeynotelWorkContainer : SBObject <KeynoteGenericMethods>
```

```
@interface KeynoteChart : KeynoteIWorkItem
// A chart
@end
@property NSInteger opacity; // The opacity of the object, in percent.
359.
@property NSInteger rotation; // The rotation of the iWork item, in degrees from 0 to
from 0 (none) to 100 (full).
@property NSInteger reflectionValue; // The percentage of reflection of the iWork item,
@property BOOL reflectionShowing; // Is the iWork item displaying a reflection?
@property (copy) KeynoteRichText *objectText; // The text contained within the shape.
any, for the shape.
@property (readonly) KeynoteItemFillOptions backgroundFillType; // The background, if
@interface KeynoteShape : KeynoteIWorkItem
// A shape container
@end
repeats.
@property KeynotePlaybackRepetitionMethod repetitionMethod; // If or how the audio clip
to 100 (full volume).
@property NSInteger clipVolume; // The volume setting for the audio clip, from 0 (none)
@property (copy) id fileName; // The name of the audio file.
@interface KeynoteAudioClip : KeynoteIWorkItem
// An audio clip
@end
@property NSInteger width; // The width of the iWork item.
point of the iWork item.
@property NSPoint position; // The horizontal and vertical coordinates of the top left
containing this iWork item.
@property (copy, readonly) KeynotelWorkContainer *parent; // The iWork container
@property BOOL locked; // Whether the object is locked.
@property NSInteger height; // The height of the iWork item.
@interface KeynotelWorkItem : SBObject <KeynoteGenericMethods>
// An item which supports formatting
 *\
 * iWork Suite
\*
```

```
// An image container
@interface KeynoteImage : KeynoteIWorkItem
@property (copy) NSString *objectDescription;
```

@property (copy) NSString *objectDescription; // Text associated with the image, read aloud by VoiceOver.

@property (copy, readonly) NSURL *file; // The image file.

@property (copy) id fileName; // The name of the image file.

@property NSInteger opacity; // The opacity of the object, in percent.

@property BOOL reflectionShowing; // Is the iWork item displaying a reflection?

@property NSInteger reflectionValue; $\//$ The percentage of reflection of the iWork item, from 0 (none) to 100 (full).

@property NSInteger rotation; $\mbox{\ensuremath{//}}$ The rotation of the iWork item, in degrees from 0 to 359.

@end

// A group container
@interface KeynoteGroup : KeynoteIWorkContainer

@property NSInteger height; // The height of the iWork item.

@property (copy, readonly) KeynotelWorkContainer *parent; // The iWork container containing this iWork item.

@property NSPoint position; // The horizontal and vertical coordinates of the top left point of the iWork item.

@property NSInteger width; // The width of the iWork item.

@property NSInteger rotation; $\//$ The rotation of the iWork item, in degrees from 0 to 359.

@end

// A line
@interface KeynoteLine : KeynoteIWorkItem

<code>@property NSPoint endPoint; $\//\$ A list of two numbers indicating the horizontal and vertical position of the line ending point.</code>

@property NSInteger reflectionValue; // The percentage of reflection of the iWork item, from 0 (none) to 100 (full).

@property NSInteger rotation; $\//\$ The rotation of the iWork item, in degrees from 0 to 359.

@property NSPoint startPoint; // A list of two numbers indicating the horizontal and vertical position of the line starting point.

@end

// A movie container
@interface KeynoteMovie : KeynoteIWorkItem

@property (copy) id fileName; // The name of the movie file.

<code>@property NSInteger movieVolume; // The volume setting for the movie, from 0 (none) to 100 (full volume).</code>

@property NSInteger opacity; // The opacity of the object, in percent.

@property BOOL reflectionShowing; // Is the iWork item displaying a reflection?
@property NSInteger reflectionValue; // The percentage of reflection of the iWork item,
from 0 (none) to 100 (full).
@property KeynotePlaybackRepetitionMethod repetitionMethod; // If or how the movie

@property KeynotePlaybackRepetitionMethod repetitionMethod; $\//\$ If or how the movie repeats.

@property NSInteger rotation; $\mbox{\ensuremath{//}}$ The rotation of the iWork item, in degrees from 0 to 359.

@end

// A table
@interface KeynoteTable : KeynoteIWorkItem

- (SBElementArray<KeynoteCell *> *) cells;
- (SBElementArray<KeynoteRow *> *) rows;
- (SBElementArray<KeynoteColumn *> *) columns;
- (SBElementArray<KeynoteRange *> *) ranges;

@property (copy) NSString *name; // The item's name.

@property (copy, readonly) KeynoteRange *cellRange; // The range describing every cell in the table.

@property (copy) KeynoteRange *selectionRange; // The cells currently selected in the table.

@property NSInteger rowCount; // The number of rows in the table.

@property NSInteger columnCount; // The number of columns in the table.

@property NSInteger headerRowCount; // The number of header rows in the table.

@property NSInteger headerColumnCount; // The number of header columns in the table.

@property NSInteger footerRowCount; // The number of footer rows in the table.

- (void) sortBy:(KeynoteColumn *)by direction:(KeynoteNMSD)direction inRows:(KeynoteRange *)inRows; // Sort the rows of the table.

@end

// A text container
@interface KeynoteTextItem : KeynoteIWorkItem

@property (readonly) KeynoteItemFillOptions backgroundFillType; // The background, if any, for the text item.

<code>@property</code> (copy) KeynoteRichText *objectText; // The text contained within the text item.

@property NSInteger opacity; // The opacity of the object, in percent.

@property BOOL reflectionShowing; // Is the iWork item displaying a reflection?

@property NSInteger reflectionValue; $\//$ The percentage of reflection of the iWork item, from 0 (none) to 100 (full).

@property NSInteger rotation; $\//\$ The rotation of the iWork item, in degrees from 0 to 359.

@end

// A range of cells in a table
@interface KeynoteRange : SBObject <KeynoteGenericMethods>

- (SBElementArray<KeynoteCell *> *) cells;
- (SBElementArray<KeynoteColumn *> *) columns;
- (SBElementArray<KeynoteRow *> *) rows;
- @property (copy) NSString *fontName; // The font of the range's cells.
 @property double fontSize; // The font size of the range's cells.
 @property KeynoteNMCT format; // The format of the range's cells.
 @property KeynoteTAHT alignment; // The horizontal alignment of content in the range's cells.
 @property (copy, readonly) NSString *name; // The range's coordinates.
 @property (copy) NSColor *textColor; // The text color of the range's cells.
- @property BOOL textWrap; // Whether text should wrap in the range's cells.
 @property (copy) NSColor *backgroundColor; // The background color of the range's cells.
 @property KeynoteTAVT verticalAlignment; // The vertical alignment of content in the range's cells.
- (void) clear; // Clear the contents of a specified range of cells, including formatting and style.
- (void) merge; // Merge a specified range of cells.
- (void) unmerge; // Unmerge all merged cells in a specified range.

@end

// A cell in a table

```
@interface KeynoteCell : KeynoteRange
@property (copy, readonly) KeynoteColumn *column; // The cell's column.
@property (copy, readonly) KeynoteRow *row; // The cell's row.
@property (copy) id value; // The actual value in the cell, or missing value if the cell is empty.
@property (copy, readonly) NSString *formattedValue; // The formatted value in the cell, or missing value if the cell is empty.
@property (copy, readonly) NSString *formula; // The formula in the cell, as text, e.g.
=SUM(40+2). If the cell does not contain a formula, returns missing value. To set the value of a cell to a formula as text, use the value property.
```

@end

// A row of cells in a table

```
@interface KeynoteRow : KeynoteRange
@property (readonly) NSInteger address; // The row's index in the table (e.g., the
second row has address 2).
@property double height; // The height of the row.
```

```
// A column of cells in a table 
@interface KeynoteColumn : KeynoteRange 
@property (readonly) NSInteger address; // The column's index in the table (e.g., the second column has address 2). 
@property double width; // The width of the column.
```

```
* Compatibility Suite

*/

*/

Deprecated Keynote application properties and verbs.

@interface KeynoteApplication (CompatibilitySuite)

@end

@interface KeynoteDocument (CompatibilitySuite)

@end

@interface KeynoteSlide (CompatibilitySuite)

@interface KeynoteSlide (CompatibilitySuite)
```