

---

# Application Services Reference Update

[Carbon](#) > [Graphics & Imaging](#)



2007-07-18



Apple Inc.  
© 2007 Apple Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
408-996-1010

Apple, the Apple logo, Carbon, ColorSync, eMac, FontSync, Inkwell, Mac, Mac OS, Macintosh, Objective-C, Quartz, QuickDraw, QuickTime, and TrueType are trademarks of Apple Inc., registered in the United States and other countries.

Aperture, Spotlight, and Switcher are trademarks of Apple Inc.

Adobe, Acrobat, and PostScript are trademarks or registered trademarks of Adobe Systems Incorporated in the U.S. and/or other countries.

Java and all Java-based trademarks are trademarks or registered trademarks of Sun

Microsystems, Inc. in the U.S. and other countries.

Mighty Mouse is a registered trademark of CBS Opertaions, Inc.

OpenGL is a registered trademark of Silicon Graphics, Inc.

Times is a registered trademark of Heidelberger Druckmaschinen AG, available from Linotype Library GmbH.

VMS is a trademark of Digital Equipment Corporation.

Simultaneously published in the United States and Canada.

**Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

**IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.**

**THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.**

**Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.**

# Contents

---

## Introduction to Application Services Reference Update 5

---

Organization of This Document 5

See Also 5

---

## 10.5 Symbol Changes 7

---

C Symbols 7

CoreText 7

SpeechSynthesis 28

HI Services 32

ATS 34

PrintCore 36

ImageIO 38

CoreGraphics 41

ColorSync 52

---

## 10.4 Symbol Changes 53

---

C Symbols 53

LaunchServices 53

ATS 58

QD 58

HI Services 59

PrintCore 65

CoreGraphics 68

AE 92

ImageIO 93

ColorSync 105

---

## 10.3 Symbol Changes 109

---

C Symbols 109

LaunchServices 109

HI Services 111

QD 115

ATS 118

PrintCore 119

CoreGraphics 122

AE 135

SpeechSynthesis 136

ColorSync 136

## 10.2 Symbol Changes 139

---

C Symbols 139  
LaunchServices 139  
AE 139  
FindByContent 141  
HIServices 143  
QD 146  
ATS 154  
PrintCore 159  
CoreGraphics 162  
ColorSync 171

## 10.1 Symbol Changes 173

---

C Symbols 173  
LaunchServices 173  
AE 174  
QD 175  
ATS 180  
PrintCore 181  
CoreGraphics 186  
ColorSync 189

## Document Revision History 193

---

# Introduction to Application Services Reference Update

---

This document summarizes the symbols that have been added to the Application Services framework. The full reference documentation notes in what version a symbol was introduced, but sometimes it's useful to see only the new symbols for a given release.

If you are not familiar with this framework you should refer to the complete framework reference documentation.

## Organization of This Document

Symbols are grouped by class or protocol for Objective-C and by header file for C. For each symbol there is a link to complete documentation, if available, and a brief description, if available.

## See Also

For reference documentation on this framework, see *Application Services Framework Reference*.



# 10.5 Symbol Changes

This article lists the symbols added to `ApplicationServices.framework` in Mac OS X v10.5.

## C Symbols

All of the header files with new symbols are listed alphabetically, with their new symbols described.

### CoreText

#### CoreText.h

##### Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CTGetCoreTextVersion</code>	Returns the version of the Core Text framework.
-----------------------------------	---

#### CTFont.h

##### Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CTFontCopyAttribute</code>	Returns the value associated with an arbitrary attribute of the given font.
<code>CTFontCopyAvailableTables</code>	Returns an array of font table tags.
<code>CTFontCopyCharacterSet</code>	Returns the Unicode character set of the font.
<code>CTFontCopyDisplayName</code>	Returns the display name of the given font.
<code>CTFontCopyFamilyName</code>	Returns the family name of the given font.
<code>CTFontCopyFeatures</code>	Returns an array of font features.
<code>CTFontCopyFeatureSettings</code>	Returns an array of font feature-setting tuples.

CTFontCopyFontDescriptor	Returns the normalized font descriptor for the given font reference.
CTFontCopyFullName	Returns the full name of the given font.
CTFontCopyGraphicsFont	Returns a Core Graphics font reference and attributes.
CTFontCopyLocalizedName	Returns a reference to a localized name for the given font.
CTFontCopyName	Returns a reference to the requested name of the given font.
CTFontCopyPostScriptName	Returns the PostScript name of the given font.
CTFontCopySupportedLanguages	Returns an array of languages supported by the font.
CTFontCopyTable	Returns a reference to the font table data.
CTFontCopyTraits	Returns the traits dictionary of the given font.
CTFontCopyVariation	Returns a variation dictionary from the font reference.
CTFontCopyVariationAxes	Returns an array of variation axes.
CTFontCreateCopyWithAttributes	Returns a new font with additional attributes based on the original font.
CTFontCreateCopyWithFamily	Returns a new font in the specified family based on the traits of the original font.
CTFontCreateCopyWithSymbolicTraits	Returns a new font in the same font family as the original with the specified symbolic traits.
CTFontCreateForString	Returns a new font reference that can best map the given string range based on the current font.
CTFontCreatePathForGlyph	Creates a path for the specified glyph.
CTFontCreateUIFontForLanguage	Returns the special user-interface font for the given language and user-interface type.
CTFontCreateWithFontDescriptor	Returns a new font reference that best matches the given font descriptor.
CTFontCreateWithGraphicsFont	Creates a new font reference from an existing Core Graphics font reference.
CTFontCreateWithName	Returns a new font reference for the given name.
CTFontCreateWithPlatformFont	Creates a new font reference from an ATS font reference.
CTFontCreateWithQuickdrawInstance	Returns a font reference for the given QuickDraw instance.



CTFontGetAdvancesForGlyphs	Calculates the advances for an array of glyphs and returns the summed advance.
CTFontGetAscent	Returns the scaled font-ascent metric of the given font.
CTFontGetBoundingBox	Returns the scaled bounding box of the given font.
CTFontGetBoundingRectsForGlyphs	Calculates the bounding rects for an array of glyphs and returns the overall bounding rectangle for the glyph run.
CTFontGetCapHeight	Returns the cap-height metric of the given font.
CTFontGetDescent	Returns the scaled font-descent metric of the given font.
CTFontGetGlyphCount	Returns the number of glyphs of the given font.
CTFontGetGlyphsForCharacters	Provides basic Unicode encoding for the given font, returning by reference an array of CGGlyph values corresponding to a given array of Unicode characters for the given font.
CTFontGetGlyphWithName	Returns the CGGlyph value for the specified glyph name in the given font.
CTFontGetLeading	Returns the scaled font-leading metric of the given font.
CTFontGetMatrix	Returns the transformation matrix of the given font.
CTFontGetPlatformFont	Returns an ATS font reference and attributes.
CTFontGetSize	Returns the point size of the given font.
CTFontGetSlantAngle	Returns the slant angle of the given font.
CTFontGetStringEncoding	Returns the best string encoding for legacy format support.
CTFontGetSymbolicTraits	Returns the symbolic traits of the given font.
CTFontGetTypeID	Returns the type identifier for Core Text font references.
CTFontGetUnderlinePosition	Returns the scaled underline position of the given font.
CTFontGetUnderlineThickness	Returns the scaled underline-thickness metric of the given font.
CTFontGetUnitsPerEm	Returns the units-per-em metric of the given font.
CTFontGetVerticalTranslationsForGlyphs	Calculates the offset from the default (horizontal) origin to the vertical origin for an array of glyphs.

CTFontGetXHeight	Returns the x-height metric of the given font.
------------------	--

### Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CTFontRef	A reference to a Core Text font object.
CTFontTableOptions	These constants describe font table options.
CTFontTableTag	Font table tags provide access to font table data.
CTFontUIFontType	These constants represent the specific user-interface purpose to specify for font creation.
kCTFontAlertHeaderFontType	The font used for alert headers.
kCTFontApplicationFontType	The default font for text documents.
kCTFontControlContentFontType	The font used for contents of user-interface controls.
kCTFontCopyrightNameKey	The name specifier for the copyright name.
kCTFontDescriptionNameKey	The name specifier for the description name.
kCTFontDesignerNameKey	The name specifier for the designer name.
kCTFontDesignerURLNameKey	The name specifier for the designer URL name.
kCTFontEmphasizedSystemDetailFontType	The system font used for emphasis in details.
kCTFontEmphasizedSystemFontType	The system font used for emphasis in alerts.
kCTFontFamilyNameKey	The name specifier for the family name.
kCTFontFeatureSelectorDefaultKey	Key to be used with a selector dictionary to get the default indicator for the selector.
kCTFontFeatureSelectorIdentifierKey	Key to be used with a selector dictionary corresponding to a feature type to obtain the selector identifier value as a CFNumberRef object.
kCTFontFeatureSelectorNameKey	Key to be used with a selector dictionary to get the localized name string for the selector as a CFStringRef object.
kCTFontFeatureSelectorSettingKey	Key to be used with a selector dictionary to get or specify the current setting for the selector.
kCTFontFeatureTypeExclusiveKey	Key to get the font feature exclusive setting of the feature as a CFBoolean object.

kCTFontFeatureTypeIdentifierKey	Key to get the font feature type value as a CFNumberRef object.
kCTFontFeatureTypeNameKey	Key to get the localized font feature type name as a CFString object.
kCTFontFeatureTypeSelectorsKey	Key to get the the array of font feature selectors as a CFArrayRef object.
kCTFontFullNameKey	The name specifier for the full name.
kCTFontLabelFontType	The font used for labels and tick marks on full-size sliders.
kCTFontLicenseNameKey	The name specifier for the license name.
kCTFontLicenseURLNameKey	The name specifier for the license URL name.
kCTFontManufacturerNameKey	The name specifier for the manufacturer name.
kCTFontMenuItemCmdKeyFontType	The font used for menu-item command-key equivalents.
kCTFontMenuItemFontType	The font used for menu items.
kCTFontMenuItemMarkFontType	The font used to draw menu item marks.
kCTFontMenuItemTitleFontType	The font used for menu titles.
kCTFontMessageFontType	The font used for standard interface items, such as button labels, menu items, and so on.
kCTFontMiniEmphasizedSystemFontType	The miniature system font used for emphasis.
kCTFontMiniSystemFontType	The standard miniature system font used for mini controls and utility window labels and text.
kCTFontNoFontType	The user-interface font type is not specified.
kCTFontPaletteFontType	The font used in tool palettes.
kCTFontPostScriptCIDNameKey	The name specifier for the PostScript character identifier (CID) font name.
kCTFontPostScriptNameKey	The name specifier for the PostScript name.
kCTFontPushButtonFontType	The font used for a push button (a rounded rectangular button with a text label on it).
kCTFontSampleTextNameKey	The name specifier for the sample text name string.
kCTFontSmallEmphasizedSystemFontType	The small system font used for emphasis.
kCTFontSmallSystemFontType	The standard small system font used for informative text in alerts, column headings in lists, help tags, and small controls.

kCTFontSmallToolbarFontType	The small font used for labels of toolbar items.
kCTFontStyleNameKey	The name specifier for the style name.
kCTFontSubFamilyNameKey	The name specifier for the subfamily name.
kCTFontSystemDetailFontType	The standard system font used for details.
kCTFontSystemFontType	The system font used for standard user-interface items such as button labels, menu items, and so on.
kCTFontTableAcnt	Font table tag for accent attachment.
kCTFontTableAvar	Font table tag for axis variation.
kCTFontTableBASE	Font table tag for the font baseline.
kCTFontTableBdat	Font table tag for bitmap data.
kCTFontTableBhed	Font table tag for bitmap font header.
kCTFontTableBloc	Font table tag for bitmap location.
kCTFontTableBsln	Font table tag for baseline.
kCTFontTableCFF	Font table tag for a PostScript font program.
kCTFontTableCmap	Font table tag for character-to-glyph mapping.
kCTFontTableCvar	Font table tag for control value variation, or CVT variation.
kCTFontTableCvt	Font table tag for control value table.
kCTFontTableDSIG	Font table tag for a digital signature.
kCTFontTableEBDT	Font table tag for an embedded bitmap.
kCTFontTableEBLC	Font table tag for the embedded bitmap location.
kCTFontTableEBSC	Font table tag for embedded bitmap scaling.
kCTFontTableFdsc	Font table tag for font descriptor.
kCTFontTableFeat	Font table tag for layout feature.
kCTFontTableFmtx	Font table tag for font metrics.
kCTFontTableFpgm	Font table tag for font program.
kCTFontTableFvar	Font table tag for font variation.
kCTFontTableGasp	Font table tag for grid-fitting/scan-conversion.
kCTFontTableGDEF	Font table tag for glyph definition.

kCTFontTableGlyf	Font table tag for glyph data.
kCTFontTableGPOS	Font table tag for glyph positioning.
kCTFontTableGSUB	Font table tag for glyph substitution.
kCTFontTableGvar	Font table tag for glyph variation.
kCTFontTableHdmx	Font table tag for horizontal device metrics.
kCTFontTableHead	Font table tag for font header.
kCTFontTableHhea	Font table tag for horizontal header.
kCTFontTableHmtx	Font table tag for horizontal metrics.
kCTFontTableHsty	Font table tag for horizontal style.
kCTFontTableJSTF	Font table tag for justification.
kCTFontTableJust	Font table tag for justification.
kCTFontTableKern	Font table tag for kerning.
kCTFontTableLcar	Font table tag for ligature caret.
kCTFontTableLoca	Font table tag for index to location.
kCTFontTableLTSH	Font table tag for linear threshold.
kCTFontTableMaxp	Font table tag for maximum profile.
kCTFontTableMort	Font table tag for morph.
kCTFontTableMorx	Font table tag for extended morph.
kCTFontTableName	Font table tag for naming table.
kCTFontTableOpbd	Font table tag for optical bounds.
kCTFontTableOptionExcludeSynthetic	The font table excludes synthetic font data.
kCTFontTableOptionNoOptions	No font table options are specified.
kCTFontTableOS2	Font table tag for OS/2 and Windows-specific metrics.
kCTFontTablePCLT	Font table tag for PCL 5 data.
kCTFontTablePost	Font table tag for PostScript information.
kCTFontTablePrep	Font table tag for control value program, 'prep' table.
kCTFontTableProp	Font table tag for properties.
kCTFontTableTrak	Font table tag for tracking.

kCTFontTableVDMX	Font table tag for vertical device metrics.
kCTFontTableVhea	Font table tag for vertical header.
kCTFontTableVmtx	Font table tag for vertical metrics.
kCTFontTableVORG	Font table tag for vertical origin.
kCTFontTableZapf	Font table tag for glyph reference.
kCTFontToolbarFontType	The font used for labels of toolbar items.
kCTFontToolTipFontType	The font used for tool tips.
kCTFontTrademarkNameKey	The name specifier for the trademark name.
kCTFontUniqueNameKey	The name specifier for the unique name.
kCTFontUserFixedPitchFontType	The font used by default for documents and other text under the user's control when that font is fixed-pitch.
kCTFontUserFontType	The font used by default for documents and other text under the user's control (that is, text whose font the user can normally change).
kCTFontUtilityWindowTitleFontType	The font used for utility window titles.
kCTFontVariationAxisDefaultValueKey	Key to get the variation axis default value as a CFNumberRef object.
kCTFontVariationAxisIdentifierKey	Key to get the variation axis identifier value as a CFNumberRef object.
kCTFontVariationAxisMaximumValueKey	Key to get the variation axis maximum value as a CFNumberRef object.
kCTFontVariationAxisMinimumValueKey	Key to get the variation axis minimum value as a CFNumberRef object.
kCTFontVariationAxisNameKey	Key to get the localized variation axis name string.
kCTFontVendorURLNameKey	The name specifier for the vendor URL name.
kCTFontVersionNameKey	The name specifier for the version name.
kCTFontViewsFontType	The view font used as the default font of text in lists and tables.
kCTFontWindowTitleFontType	The font used for window titles.

## CTFontCollection.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CTFontCollectionCreateCopyWithFontDescriptors	Returns a copy of the original collection augmented with the given new font descriptors.
CTFontCollectionCreateFromAvailableFonts	Returns a new font collection containing all available fonts.
CTFontCollectionCreateMatchingFontDescriptors	Returns an array of font descriptors matching the collection.
CTFontCollectionCreateMatchingFont- DescriptorsSortedWithCallback	Returns the array of matching font descriptors sorted with the callback function.
CTFontCollectionCreateWithFontDescriptors	Returns a new font collection based on the given array of font descriptors.
CTFontCollectionGetTypeID	Returns the type identifier for Core Text font collection references.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CTFontCollectionRef	A reference to a font collection.
CTFontCollectionSortDescriptorsCallback	
kCTFontCollectionRemoveDuplicatesOption	Option key to specify filtering of duplicates.

## CTFontDescriptor.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CTFontDescriptorCopyAttribute	Returns the value associated with an arbitrary attribute.
CTFontDescriptorCopyAttributes	Returns the attributes dictionary of the font descriptor.

CTFontDescriptorCopyLocalizedAttribute	Returns a localized value for the requested attribute, if available.
CTFontDescriptorCreateCopyWithAttributes	Creates a copy of the original font descriptor with new attributes.
CTFontDescriptorCreateCopyWithFeature	Copies a font descriptor with new feature settings.
CTFontDescriptorCreateCopyWithVariation	Creates a copy of the original font descriptor with a new variation instance.
CTFontDescriptorCreateMatchingFontDescriptor	Returns the single preferred matching font descriptor based on the original descriptor and system precedence.
CTFontDescriptorCreateMatchingFontDescriptors	Returns an array of normalized font descriptors matching the provided descriptor.
CTFontDescriptorCreateWithAttributes	Creates a new font descriptor reference from a dictionary of attributes.
CTFontDescriptorCreateWithNameAndSize	Creates a new font descriptor with the provided PostScript name and size.
CTFontDescriptorGetTypeID	Returns the type identifier for Core Text font descriptor references.

### Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CTFontDescriptorRef	A reference to a CTFontDescriptor object.
CTFontOrientation	Specifies the intended rendering orientation of the font for obtaining glyph metrics.
kCTFontBaselineAdjustAttribute	Key to specify or obtain the baseline adjustment for a font reference.
kCTFontCascadeListAttribute	Key to specify or obtain the cascade list used for a font reference.
kCTFontCharacterSetAttribute	Key to specify or obtain the Unicode character coverage set for a font reference.
kCTFontDefaultOrientation	The native orientation of the font.
kCTFontDisplayNameAttribute	Key for accessing the name used to display the font.
kCTFontFamilyNameAttribute	Key for accessing the font family name from the font descriptor.



kCTFontFeaturesAttribute	Key to specify or obtain the font features for a font reference.
kCTFontFeatureSettingsAttribute	Key to specify or obtain the font features settings for a font reference.
kCTFontFixedAdvanceAttribute	Key to specify a fixed advance to be used for a font reference.
kCTFontHorizontalOrientation	Specifies horizontal orientation.
kCTFontLanguagesAttribute	Key to specify or obtain a list of covered languages for a font reference.
kCTFontMacintoshEncodingsAttribute	Key to specify or obtain the Macintosh encodings for a font reference.
kCTFontMatrixAttribute	Key to specify the font transformation matrix when creating a font.
kCTFontNameAttribute	Key for accessing the PostScript name from the font descriptor.
kCTFontOrientationAttribute	Key to specify a particular orientation for the glyphs of the font.
kCTFontSizeAttribute	Key to obtain or specify the font point size.
kCTFontStyleNameAttribute	Key for accessing the style name of the font.
kCTFontTraitsAttribute	Key for accessing the dictionary of font traits for stylistic information.
kCTFontVariationAttribute	Key to obtain the font variation dictionary instance as a CFDictionaryRef object.
kCTFontVerticalOrientation	Specifies vertical orientation.

## CTFontTraits.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CTFontStylisticClass	These constants represent the stylistic class values of the font.
CTFontSymbolicTraits	These constants represent the symbolic representation of stylistic font attributes.
kCTFontBoldTrait	The font typestyle is boldface.
kCTFontClarendonSerifsClass	The font's style is a variation of the Oldstyle Serifs and the Transitional Serifs.

<code>kCTFontClassMaskShift</code>	Value used to shift the font class to the uppermost four bits of the symbolic traits
<code>kCTFontClassMaskTrait</code>	Mask for the font class.
<code>kCTFontCondensedTrait</code>	The font typestyle is condensed.
<code>kCTFontExpandedTrait</code>	The font typestyle is expanded.
<code>kCTFontFreeformSerifsClass</code>	The font's style includes serifs, but it expresses a design freedom that does not generally fit within the other serif design classifications.
<code>kCTFontItalicTrait</code>	The font typestyle is italic.
<code>kCTFontModernSerifsClass</code>	The font's style is based on the Latin printing style of the 20th century.
<code>kCTFontMonoSpaceTrait</code>	The font uses fixed-pitch glyphs if available.
<code>kCTFontOldStyleSerifsClass</code>	The font's style is based on the Latin printing style of the 15th to 17th century.
<code>kCTFontOrnamentalsClass</code>	The font's style includes highly decorated or stylized character shapes such as those typically used in headlines.
<code>kCTFontSansSerifClass</code>	The font's style includes most basic letter forms (excluding Scripts and Ornamentals) that do not have serifs on the strokes.
<code>kCTFontScriptsClass</code>	The font's style is among those typefaces designed to simulate handwriting.
<code>kCTFontSlabSerifsClass</code>	The font's style is characterized by serifs with a square transition between the strokes and the serifs (no brackets).
<code>kCTFontSlantTrait</code>	Key to access the normalized slant angle from the font traits dictionary.
<code>kCTFontSymbolicClass</code>	The font's style is generally design independent, making it suitable for special characters (icons, dingbats, technical symbols, and so on) that may be used equally well with any font.
<code>kCTFontSymbolicTrait</code>	Key to access the symbolic traits value from the font traits dictionary.
<code>kCTFontTransitionalSerifsClass</code>	The font's style is based on the Latin printing style of the 18th to 19th century.
<code>kCTFontUIOptimizedTrait</code>	The font synthesizes appropriate attributes for user interface rendering, such as control titles, if necessary.
<code>kCTFontUnknownClass</code>	The font has no design classification.
<code>kCTFontVerticalTrait</code>	The font uses vertical glyph variants and metrics.

<code>kCTFontWeightTrait</code>	Key to access the normalized weight trait from the font traits dictionary.
<code>kCTFontWidthTrait</code>	Key to access the normalized proportion (width condense or expand) trait from the font traits dictionary.

## CTFrame.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CTFrameDraw</code>	Draws an entire frame into a context.
<code>CTFrameGetFrameAttributes</code>	Returns the frame attributes used to create the frame.
<code>CTFrameGetLineOrigins</code>	Copies a range of line origins for a frame.
<code>CTFrameGetLines</code>	Returns an array of lines stored in the frame.
<code>CTFrameGetPath</code>	Returns the path used to create the frame.
<code>CTFrameGetStringRange</code>	Returns the range of characters originally requested to fill the frame.
<code>CTFrameGetTypeID</code>	Returns the type identifier for the CTFrame opaque type.
<code>CTFrameGetVisibleStringRange</code>	Returns the range of characters that actually fit in the frame.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CTFrameProgression</code>	These constants specify frame progression types.
<code>CTFrameRef</code>	A reference to a Core text frame object.
<code>kCTFrameProgressionAttributeName</code>	A CFNumberRef object containing a “CTFrameProgression” constant.
<code>kCTFrameProgressionRightToLeft</code>	Lines are stacked right to left for vertical text.
<code>kCTFrameProgressionTopToBottom</code>	Lines are stacked top to bottom for horizontal text.

## CTFramesetter.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CTFramesetterCreateFrame	Creates an immutable frame using a framesetter.
CTFramesetterCreateWithAttributedString	Creates an immutable framesetter object from an attributed string.
CTFramesetterGetTypeID	Returns the Core Foundation type identifier of the framesetter object.
CTFramesetterGetTypesetter	Returns the typesetter object being used by the framesetter.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CTFramesetterRef	A reference to a Core Foundation framesetter object.
------------------	--

## CTGlyphInfo.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CTGlyphInfoCreateWithCharacterIdentifier	Creates an immutable glyph info object with a character identifier.
CTGlyphInfoCreateWithGlyph	Creates an immutable glyph info object with a glyph index.
CTGlyphInfoCreateWithGlyphName	Creates an immutable glyph info object with a glyph name.
CTGlyphInfoGetCharacterCollection	Gets the character collection for a glyph info object.
CTGlyphInfoGetCharacterIdentifier	Gets the character identifier for a glyph info object.
CTGlyphInfoGetGlyphName	Gets the glyph name for a glyph info object if that object exists.
CTGlyphInfoGetTypeID	Returns the Core Foundation type identifier of the glyph info object

## Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CTCharacterCollection	These constants specify character collections.
CTGlyphInfoRef	A reference to a glyph info object.
kCTAdobeCNS1CharacterCollection	The Adobe-CNS1 mapping.
kCTAdobeGB1CharacterCollection	The Adobe-GB1 mapping.
kCTAdobeJapan1CharacterCollection	The Adobe-Japan1 mapping.
kCTAdobeJapan2CharacterCollection	The Adobe-Japan2 mapping.
kCTAdobeKorea1CharacterCollection	The Adobe-Korea1 mapping.
kCTIdentityMappingCharacterCollection	The character identifier is equal to the CGGlyph glyph index.

## CTLine.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CTLineCreateJustifiedLine	Creates a justified line from an existing line.
CTLineCreateTruncatedLine	Creates a truncated line from an existing line.
CTLineCreateWithAttributedString	Creates a single immutable line object directly from an attributed string.
CTLineDraw	Draws a complete line.
CTLineGetGlyphCount	Returns the total glyph count for the line object.
CTLineGetGlyphRuns	Returns the array of glyph runs that make up the line object.
CTLineGetImageBounds	Calculates the image bounds for a line.
CTLineGetOffsetForStringIndex	Determines the graphical offset or offsets for a string index.
CTLineGetPenOffsetForFlush	Gets the pen offset required to draw flush text.
CTLineGetStringIndexForPosition	Performs hit testing.
CTLineGetStringRange	Gets the range of characters that originally spawned the glyphs in the line.

CTLineGetTrailingWhitespaceWidth	Returns the trailing whitespace width for a line.
CTLineGetTypeID	Returns the Core Foundation type identifier of the line object.
CTLineGetTypographicBounds	Calculates the typographic bounds of a line.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CTLineRef	A reference to a line object.
CTLineTruncationType	Truncation types required by the CTLineCreateTruncatedLine function to tell the truncation engine which type of truncation is being requested.
kCTLineTruncationEnd	Truncate the end of the line, leaving the start portion visible.
kCTLineTruncationMiddle	Truncate the middle of the line, leaving both the start and the end portions visible.
kCTLineTruncationStart	Truncate the beginning of the line, leaving the end portion visible.

## CTParagraphStyle.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CTParagraphStyleCreate	Creates an immutable paragraph style.
CTParagraphStyleCreateCopy	Creates an immutable copy of a paragraph style.
CTParagraphStyleGetTypeID	Returns the Core Foundation type identifier of the paragraph style object.
CTParagraphStyleGetValueForSpecifier	Obtains the current value for a single setting specifier.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CTLineBreakMode	These constants specify what happens when a line is too long for its frame.
-----------------	---

CTParagraphStyleRef	A reference to a Core Text paragraph style.
CTParagraphStyleSpecifier	These constants are used to query and modify the CTParagraphStyle object.
CTTextAlignment	These constants specify text alignment.
CTWritingDirection	These constants specify the writing direction.
kCTCenterTextAlignment	Text is visually center aligned.
kCTJustifiedTextAlignment	Text is fully justified.
kCTLeftTextAlignment	Text is visually left aligned.
kCTLineBreakByCharWrapping	Wrapping occurs before the first character that doesn't fit.
kCTLineBreakByClipping	Lines are simply not drawn past the edge of the frame.
kCTLineBreakByTruncatingHead	Each line is displayed so that the end fits in the frame and the missing text is indicated by an ellipsis glyph.
kCTLineBreakByTruncatingMiddle	Each line is displayed so that the beginning and end fit in the container and the missing text is indicated by an ellipsis glyph in the middle.
kCTLineBreakByTruncatingTail	Each line is displayed so that the beginning fits in the container and the missing text is indicated by an ellipsis glyph.
kCTLineBreakByWordWrapping	Wrapping occurs at word boundaries unless the word itself doesn't fit on a single line.
kCTNaturalTextAlignment	Text uses the natural alignment of the text's script.
kCTParagraphStyleSpecifierAlignment	The text alignment.
kCTParagraphStyleSpecifierBaseWritingDirection	The base writing direction of the lines.
kCTParagraphStyleSpecifierCount	The number of style specifiers.
kCTParagraphStyleSpecifierDefaultTabInterval	The documentwide default tab interval.
kCTParagraphStyleSpecifierFirstLineHeadIndent	The distance, in points, from the leading margin of a frame to the beginning of the paragraph's first line.

kCTParagraphStyleSpecifierHeadIndent	The distance, in points, from the leading margin of a text container to the beginning of lines other than the first.
kCTParagraphStyleSpecifierLineBreakMode	The mode that should be used to break lines when laying out the paragraph's text.
kCTParagraphStyleSpecifierLineHeightMultiple	The line height multiple.
kCTParagraphStyleSpecifierLineSpacing	The space in points added between lines within the paragraph (commonly known as leading).
kCTParagraphStyleSpecifierMaximumLineHeight	The maximum height that any line in the frame will occupy, regardless of the font size or size of any attached graphic.
kCTParagraphStyleSpecifierMinimumLineHeight	The minimum height that any line in the frame will occupy, regardless of the font size or size of any attached graphic.
kCTParagraphStyleSpecifierParagraphSpacing	The space added at the end of the paragraph to separate it from the following paragraph.
kCTParagraphStyleSpecifierParagraphSpacingBefore	The distance between the paragraph's top and the beginning of its text content.
kCTParagraphStyleSpecifierTabStops	The CTextTab objects, sorted by location, that define the tab stops for the paragraph style.
kCTParagraphStyleSpecifierTailIndent	The distance, in points, from the margin of a frame to the end of lines.
kCTRightTextAlignment	Text is visually right aligned.
kCTWritingDirectionLeftToRight	The writing direction is left to right.
kCTWritingDirectionNatural	The writing direction is algorithmically determined using the Unicode Bidirectional Algorithm rules P2 and P3.
kCTWritingDirectionRightToLeft	The writing direction is right to left.
valueSize	



## CTRun.h

## Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CTRunDraw	Draws a complete run or part of one.
CTRunGetAttributes	Returns the attribute dictionary that was used to create the glyph run.
CTRunGetGlyphCount	Gets the glyph count for the run.
CTRunGetGlyphs	Copies a range of glyphs into a user-provided buffer.
CTRunGetGlyphsPtr	Returns a direct pointer for the glyph array stored in the run.
CTRunGetImageBounds	Calculates the image bounds for a glyph range.
CTRunGetPositions	Copies a range of glyph positions into a user-provided buffer.
CTRunGetPositionsPtr	Returns a direct pointer for the glyph position array stored in the run.
CTRunGetStatus	Returns the run's status.
CTRunGetStringIndices	Copies a range of string indices into a user-provided buffer.
CTRunGetStringIndicesPtr	Returns a direct pointer for the string indices stored in the run.
CTRunGetStringRange	Gets the range of characters that originally spawned the glyphs in the run.
CTRunGetTextMatrix	Returns the text matrix needed to draw this run.
CTRunGetTypeID	Returns the Core Foundation type identifier of the run object.
CTRunGetTypographicBounds	Gets the typographic bounds of the run.

## Data Types &amp; Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CTRunRef	A reference to a run object.
CTRunStatus	A bitfield passed back by the CTRunGetStatus function that is used to indicate the disposition of the run.
kCTRunStatusHasNonIdentityMatrix	The run requires a specific text matrix to be set in the current Core Graphics context for proper drawing.

kCTRunStatusNonMonotonic	The run has been reordered in some way such that the string indices associated with the glyphs are no longer strictly increasing (for left-to-right runs) or decreasing (for right-to-left runs).
kCTRunStatusNoStatus	The run has no special attributes.
kCTRunStatusRightToLeft	The run proceeds from right to left.

## CTStringAttributes.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CTUnderlineStyle	Underline style specifiers.
CTUnderlineStyleModifiers	Underline style modifiers.
kCTFontAttributeName	The font of the text to which this attribute applies.
kCTForegroundColorAttributeName	The foreground color of the text to which this attribute applies.
kCTGlyphInfoAttributeName	The glyph info object to apply to the text associated with this attribute.
kCTKernAttributeName	The amount to kern the next character.
kCTLigatureAttributeName	The type of ligatures to use.
kCTParagraphStyleAttributeName	The paragraph style of the text to which this attribute applies.
kCTUnderlinePatternDash	Draw an underline using a pattern of dashes.
kCTUnderlinePatternDashDot	Draw an underline using a pattern of alternating dashes and dots.
kCTUnderlinePatternDashDotDot	Draw an underline using a pattern of a dash followed by two dots.
kCTUnderlinePatternDot	Draw an underline using a pattern of dots.
kCTUnderlinePatternSolid	Draw a solid underline.
kCTUnderlineStyleAttributeName	The style of underlining, to be applied at render time, for the text to which this attribute applies.
kCTUnderlineStyleDouble	Draw an underline consisting of a double line.
kCTUnderlineStyleNone	Do not draw an underline.

<code>kCTUnderlineStyleSingle</code>	Draw an underline consisting of a single line.
<code>kCTUnderlineStyleThick</code>	Draw an underline consisting of a thick line.
<code>kCTVerticalFormsAttributeName</code>	The orientation of the glyphs in the text to which this attribute applies.

## CTTextTab.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CTTextTabCreate</code>	Creates and initializes a new text tab object.
<code>CTTextTabGetAlignment</code>	Returns the text alignment of the tab.
<code>CTTextTabGetLocation</code>	Returns the tab's ruler location.
<code>CTTextTabGetOptions</code>	Returns the dictionary of attributes associated with the tab.
<code>CTTextTabGetTypeID</code>	Returns the Core Foundation type identifier of the text tab object.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CTTextTabRef</code>	A reference to a text tab object.
<code>kCTTabColumnTerminatorsAttributeName</code>	Specifies the terminating character for a tab column.

## CTTypesetter.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CTTypesetterCreateLine</code>	Creates an immutable line from the typesetter.
<code>CTTypesetterCreateWithAttributedString</code>	Creates an immutable typesetter object using an attributed string.

<code>CTTypesetterCreateWithAttributedStringAndOptions</code>	Creates an immutable typesetter object using an attributed string and a dictionary of options.
<code>CTTypesetterGetTypeID</code>	Returns the Core Foundation type identifier of the typesetter object.
<code>CTTypesetterSuggestClusterBreak</code>	Suggests a cluster line breakpoint based on the width provided.
<code>CTTypesetterSuggestLineBreak</code>	Suggests a contextual line breakpoint based on the width provided.

### Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CTTypesetterRef</code>	A reference to a typesetter object.
<code>kCTTypesetterOptionDisableBidiProcessing</code>	Disables bidirectional processing.

## SpeechSynthesis

### SpeechSynthesis.h

#### Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CopyPhonemesFromText</code>	Converts the specified text string into its equivalent phonemic representation.
<code>CopySpeechProperty</code>	Gets the value associated with the specified property of a speech channel.
<code>SetSpeechProperty</code>	Sets the value of the specified speech-channel property.
<code>SpeakCFString</code>	Begins speaking a string represented as a CFString object.
<code>UseSpeechDictionary</code>	Registers a speech dictionary with a speech channel.

### Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

aString	
kAudioUnitProperty_SpeechChannel	
kAudioUnitProperty_Voice	
kAudioUnitSubType_SpeechSynthesis	
kSpeechCharacterModeProperty	Get or set the speech channel's current character-processing mode.
kSpeechCommandDelimiterProperty	Set the embedded speech command delimiter characters to be used for the speech channel.
kSpeechCommandPrefix	The command delimiter string that prefixes a command (by default, this is []).
kSpeechCommandSuffix	The command delimiter string that suffixes a command (by default, this is []).
kSpeechCurrentVoiceProperty	Set the current voice on the current speech channel to the specified voice.
kSpeechDictionaryAbbreviations	The set of custom pronunciations for abbreviations.
kSpeechDictionaryEntryPhonemes	The phonemic representation of an entry.
kSpeechDictionaryEntrySpelling	The spelling of an entry.
kSpeechDictionaryLocaleIdentifier	The locale associated with the pronunciation.
kSpeechDictionaryModificationDate	The date the dictionary was last modified.
kSpeechDictionaryPronunciations	The set of custom pronunciations.
kSpeechErrorCallbackCharacterOffset	
kSpeechErrorCallbackSpokenString	The string to speak to the user when an error occurs.
kSpeechErrorCFCallBack	Set the callback function to be called when an error is encountered during the processing of an embedded command.
kSpeechErrorCount	The number of errors that have occurred in processing the current text string, since the last call to the CopySpeechProperty function with the kSpeechErrorsProperty property.
kSpeechErrorNewest	The error code of the most recent error that occurred since the last call to the CopySpeechProperty function with the kSpeechErrorsProperty property.
kSpeechErrorNewestCharacterOffset	The position in the text string of the most recent error that occurred since the last call to the CopySpeechProperty function with the kSpeechErrorsProperty property.

kSpeechErrorOldest	The error code of the first error that occurred since the last call to the CopySpeechProperty function with the kSpeechErrorsProperty property.
kSpeechErrorOldestCharacterOffset	The position in the text string of the first error that occurred since the last call to the CopySpeechProperty function with the kSpeechErrorsProperty property.
kSpeechErrorsProperty	Get speech-error information for the speech channel.
kSpeechImmediateBoundary	
kSpeechInputModeProperty	Get or set the speech channel's current text-processing mode.
kSpeechModeLiteral	Indicates that each digit is spoken literally (so that "12" is spoken as "one, two").
kSpeechModeNormal	Indicates that the synthesizer assembles digits into numbers (so that "12" is spoken as "twelve").
kSpeechModePhoneme	Indicates that the speech channel is in phoneme-processing mode. When in phoneme-processing mode, a text buffer is interpreted to be a series of characters representing various phonemes and prosodic controls.
kSpeechModeText	Indicates that the speech channel is in text-processing mode.
kSpeechNoEndingProsody	Disable prosody at the end of sentences.
kSpeechNoSpeechInterrupt	Do not interrupt current speech.
kSpeechNumberModeProperty	Get or set the speech channel's current number-processing mode.
kSpeechOutputToFileURLProperty	Set the speech output destination to a file or to the computer's speakers.
kSpeechPhonemeCallback	Set the callback function to be called every time the Speech Synthesis Manager is about to generate a phoneme on the speech channel.
kSpeechPhonemeInfoExample	An example word that illustrates the use of the phoneme.
kSpeechPhonemeInfoHiliteEnd	The character offset into the example word that identifies the location of the end of the phoneme.
kSpeechPhonemeInfoHiliteStart	The character offset into the example word that identifies the location of the beginning of the phoneme.
kSpeechPhonemeInfoOpcode	The opcode of the phoneme.
kSpeechPhonemeInfoSymbol	The symbol used to represent the phoneme.

kSpeechPhonemeSymbolsProperty	Get a list of phoneme symbols and example words defined for the speech channel's synthesizer.
kSpeechPitchBaseProperty	Get or set the speech channel's baseline speech pitch.
kSpeechPitchModProperty	Get or set a speech channel's pitch modulation.
kSpeechPreflightThenPause	Compute speech without generating it.
kSpeechRateProperty	Get or set a speech channel's speech rate.
kSpeechRecentSyncProperty	Get the message code for the most recently encountered synchronization command.
kSpeechRefConProperty	Set a speech channel's reference constant value.
kSpeechResetProperty	Set a speech channel back to its default state.
kSpeechSentenceBoundary	
kSpeechSpeechDoneCallback	Set the callback function to be called when the Speech Synthesis Manager has finished generating speech on the speech channel.
kSpeechStatusNumberOfCharactersLeft	The number of characters left in the input string of text.
kSpeechStatusOutputBusy	Indicates whether the speech channel is currently producing speech.
kSpeechStatusOutputPaused	Indicates whether speech output in the speech channel has been paused by a call to the PauseSpeechAt function.
kSpeechStatusPhonemeCode	The opcode for the phoneme that the speech channel is currently processing.
kSpeechStatusProperty	Get speech-status information for the speech channel.
kSpeechSyncCallback	Set the callback function to be called when the Speech Synthesis Manager encounters a synchronization command within an embedded speech command in text being processed on the speech channel.
kSpeechSynthesizerInfoIdentifier	The identifier of the speech synthesizer.
kSpeechSynthesizerInfoManufacturer	
kSpeechSynthesizerInfoProperty	Get information about the speech synthesizer being used on the specified speech channel.
kSpeechSynthesizerInfoVersion	The version of the speech synthesizer.
kSpeechTextDoneCallback	Set the callback function to be called when the Speech Synthesis Manager has finished processing speech being generated on the speech channel.

<code>kSpeechVoiceCreator</code>	The synthesizer that is required to use the voice.
<code>kSpeechVoiceID</code>	The voice ID of the voice for the synthesizer (every voice on a synthesizer has a unique ID).
<code>kSpeechVolumeProperty</code>	Get or set the speech volume for a speech channel.
<code>kSpeechWordBoundary</code>	
<code>kSpeechWordCFCallback</code>	Set the callback function to be called every time the Speech Synthesis Manager is about to generate a word on the speech channel.
<code>SpeechBoundary</code>	
<code>wordRange</code>	

## HI Services

---

### AXError.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kAXErrorNotEnoughPrecision</code>	
---	--

### HIShape.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>HIShapeCreateMutableWithRect</code>	
---	--

<code>HIShapeCreateXor</code>	
-------------------------------	--

<code>HIShapeEnumerate</code>	
-------------------------------	--

<code>HIShapeInset</code>	
---------------------------	--

<code>HIShapeSetWithShape</code>	
----------------------------------	--

<code>HIShapeUnionWithRect</code>	
-----------------------------------	--

<code>HIShapeXor</code>	
-------------------------	--



## Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>inRect</code>	
<code>inRefcon</code>	
<code>inShape</code>	
<code>kHIShapeEnumerateInit</code>	
<code>kHIShapeEnumerateRect</code>	
<code>kHIShapeEnumerateTerminate</code>	
<code>kHIShapeParseFromBottom</code>	
<code>kHIShapeParseFromBottomRight</code>	
<code>kHIShapeParseFromLeft</code>	
<code>kHIShapeParseFromRight</code>	
<code>kHIShapeParseFromTop</code>	
<code>kHIShapeParseFromTopLeft</code>	

## Icons.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>IconRefContainsCGPoint</code>	Returns a Boolean value indicating whether an icon contains a specified point.
<code>IconRefIntersectsCGRect</code>	Returns a Boolean value indicating whether an icon intersects a specified rectangle.
<code>IconRefToHIShape</code>	Converts an icon into an HIShape object.

## Pasteboard.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kPasteboardStandardLocationTrash	
kPasteboardStandardLocationUnknown	
PasteboardStandardLocation	

## Processes.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

launchAppRef	
processAppRef	

## ATS

---

### ATSFont.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ATSFontActivateFromFileReference	Activates one or more fonts from a file reference.
ATSFontGetAutoActivationSettingForApplication	Gets the activation setting for the specified application.
ATSFontGetContainer	Gets the font container reference for a font.
ATSFontGetContainerFromFileReference	Gets the font container reference associated with an activated file reference.
ATSFontGetFileReference	Obtains the file reference for a font.
ATSFontGetGlobalAutoActivationSetting	Gets the user's global auto-activation setting.
ATSFontIsEnabled	Returns true if the font is enabled.
ATSFontSetAutoActivationSettingForApplication	Sets the auto-activation setting for the specified application bundle.
ATSFontSetEnabled	Sets a font state to enabled or disabled.
ATSFontSetGlobalAutoActivationSetting	Sets the user's global auto-activation setting.

**Data Types & Constants**

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ATSFonAutoActivationSetting	Values for automatic activation settings.
fontFileRefFilter	
kATSFileReferenceFilterSelector	
kATSFonAutoActivationAsk	Asks the user before automatically activating fonts requested by the application.
kATSFonAutoActivationDefault	Resets the setting the the default state. For application settings this clears the setting. For the global setting, it reverts to the initial system setting, kATSFonAutoActivationEnabled.
kATSFonAutoActivationDisabled	Disables automatic activation of fonts.
kATSFonAutoActivationEnabled	Enables automatic activation of fonts.
kATSOptionFlagsActivateDisabled	
kATSOptionFlagsIncludeDisabledMask	
kATSOptionFlagsRecordPersistently	

**ATSTypes.h**

---

**Data Types & Constants**

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ATSFSSpec	
ATSPoint	
CGFloat	The basic type for all floating-point values.
kFMFontFileRefFilterSelector	
URefCon	

**SFNTTypes.h**

---

**Data Types & Constants**

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kFontMacCompatibleFullName	
kFontPostScriptCIDName	
kFontPreferredFamilyName	
kFontPreferredSubfamilyName	
kFontSampleTextName	

## PrintCore

---

### PMCore.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

PMCreateGenericPrinter	Creates a generic printer object.
PMPageFormatCreateDataRepresentation	Creates a data representation of a page format object.
PMPageFormatCreateWithDataRepresentation	Creates a page format object from a data representation.
PMPageFormatGetPrinterID	Obtains the identifier of the formatting printer for a page format object.
PMPaperCreateCustom	Creates a custom paper object.
PMPaperCreateLocalizedName	Obtains the localized name for a given paper.
PMPaperGetPPDPaperName	Obtains the PPD paper name for a given paper.
PMPaperGetPrinterID	Obtains the printer ID of the printer to which a given paper corresponds.
PMPaperIsCustom	Returns a Boolean value indicating whether a specified paper is a custom paper.
PMPrinterGetOutputResolution	Obtains the printer hardware output resolution for the specified print settings.
PMPrinterIsPostScriptPrinter	Determines whether a printer is a PostScript printer.

<code>PMPrinterSetDefault</code>	Sets the default printer for the current user.
<code>PMPrinterSetOutputResolution</code>	Sets the print settings to reflect the specified printer hardware output resolution.
<code>PMPrinterWritePostScriptToURL</code>	Converts an input file of the specified MIME type to printer-ready PostScript for a destination printer.
<code>PMPrintSettingsCopyAsDictionary</code>	Creates a dictionary that contains the settings in a print settings object.
<code>PMPrintSettingsCopyKeys</code>	Obtains the keys for items in a print settings object.
<code>PMPrintSettingsCreateDataRepresentation</code>	Creates a data representation of a print settings object.
<code>PMPrintSettingsCreateWithDataRepresentation</code>	Creates a print settings object from a data representation.
<code>PMPrintSettingsToOptionsWithPrinterAndPageFormat</code>	Converts print settings and page format data into a CUPS options string for a specified printer.
<code>PMServerLaunchPrinterBrowser</code>	Launches the printer browser to browse the printers available for a print server.

## PMDefinitions.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kPMDDataFormatXMLCompressed</code>	Specifies a compressed data format that is approximately 20 times smaller than <code>kPMDDataFormatXMLDefault</code> .
<code>kPMDDataFormatXMLDefault</code>	Specifies a data format that is compatible with all Mac OS X versions.
<code>kPMDDataFormatXMLMinimal</code>	Specifies an uncompressed data format that is approximately 3-5 times smaller than <code>kPMDDataFormatXMLDefault</code> .
<code>PMDataFormat</code>	Constants that specify the format of the data representation created with the functions <code>PMPageFormatCreateDataRepresentation</code> and <code>PMPrintSettingsCreateDataRepresentation</code> .

## PMTicket.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

PMTicketCopyKeys	
------------------	--

## ImageIO

---

### CGImageProperties.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kCGImageProperty8BIMLayerNames	The layer names for an Adobe Photoshop file.
kCGImagePropertyCIFFCameraSerialNumber	The camera serial number.
kCGImagePropertyCIFFContinuousDrive	The continuous drive mode.
kCGImagePropertyCIFFDescription	The camera description..
kCGImagePropertyCIFFFirmware	The firmware version.
kCGImagePropertyCIFFFFlashExposureComp	The flash exposure compensation.
kCGImagePropertyCIFFFocusMode	The focus mode.
kCGImagePropertyCIFFImageFileName	The image file name.
kCGImagePropertyCIFFImageName	The image name.
kCGImagePropertyCIFFImageSerialNumber	The image serial number.
kCGImagePropertyCIFFLensMaxMM	The maximum lens length.
kCGImagePropertyCIFFLensMinMM	The minimum lens length.
kCGImagePropertyCIFFLensModel	The lens model.
kCGImagePropertyCIFFMeasuredEV	The measured EV.
kCGImagePropertyCIFFMeteringMode	The metering mode.
kCGImagePropertyCIFFOwnerName	The owner name.
kCGImagePropertyCIFFRecordID	The record ID>

kCGImagePropertyCIFFReleaseMethod	The release method.
kCGImagePropertyCIFFReleaseTiming	The release timing.
kCGImagePropertyCIFFSelfTimingTime	The self timing time.
kCGImagePropertyCIFFShootingMode	The shooting mode.
kCGImagePropertyCIFFWhiteBalanceIndex	The white balance index.
kCGImagePropertyDNGBackwardVersion	The oldest version for which a file is compatible.
kCGImagePropertyDNGCameraSerialNumber	The camera serial number.
kCGImagePropertyDNGDictionary	A dictionary of key-value pairs for an image that uses the Digital Negative (DNG) archival format. See “DNG Dictionary Keys.”
kCGImagePropertyDNGLensInfo	Information about the lens used for the image.
kCGImagePropertyDNGLocalizedCameraModel	The localized camera model name.
kCGImagePropertyDNGUniqueCameraModel	A unique, nonlocalized name for the camera model.
kCGImagePropertyDNGVersion	An encoding of the four-tier version number.
kCGImagePropertyExifAuxDictionary	An auxiliary dictionary of key-value pairs for an image that uses Exchangeable Image File Format (EXIF).
kCGImagePropertyExifAuxFirmware	Firmware information.
kCGImagePropertyExifAuxFlashCompensation	Flash compensation.
kCGImagePropertyExifAuxImageNumber	The image number.
kCGImagePropertyExifAuxLensID	The lens ID.
kCGImagePropertyExifAuxLensInfo	Lens information.
kCGImagePropertyExifAuxLensModel	The lens model.
kCGImagePropertyExifAuxLensSerialNumber	The lens serial number.
kCGImagePropertyExifAuxOwnerName	The owner name.
kCGImagePropertyExifAuxSerialNumber	The serial number.
kCGImagePropertyMakerCanonAspectRatioInfo	The image aspect ratio.
kCGImagePropertyMakerCanonCameraSerialNumber	The camera serial number.
kCGImagePropertyMakerCanonContinuousDrive	The presence of a continuous drive.

kCGImagePropertyMakerCanonDictionary	A dictionary of key-value pairs for an image from a Canon camera. See “Canon Camera Dictionary Keys.”
kCGImagePropertyMakerCanonFirmware	The firmware version.
kCGImagePropertyMakerCanonFlashExposureComp	The flash exposure compensation.
kCGImagePropertyMakerCanonImageSerialNumber	The image serial number.
kCGImagePropertyMakerCanonLensModel	The lens model.
kCGImagePropertyMakerCanonOwnerName	The owner name.
kCGImagePropertyMakerFujiDictionary	A dictionary of key-value pairs for an image from a Fuji camera.
kCGImagePropertyMakerMinoltaDictionary	A dictionary of key-value pairs for an image from a Minolta camera.
kCGImagePropertyMakerNikonCameraSerialNumber	The camera serial number.
kCGImagePropertyMakerNikonColorMode	The color mode.
kCGImagePropertyMakerNikonDictionary	A dictionary of key-value pairs for an image from a Nikon camera. See “Nikon Camera Dictionary Keys.”
kCGImagePropertyMakerNikonDigitalZoom	The digital zoom setting.
kCGImagePropertyMakerNikonFlashExposureComp	The flash exposure compensation.
kCGImagePropertyMakerNikonFlashSetting	The flash setting.
kCGImagePropertyMakerNikonFocusDistance	The focus distance.
kCGImagePropertyMakerNikonFocusMode	The focus mode.
kCGImagePropertyMakerNikonImageAdjustment	Image adjustment setting.
kCGImagePropertyMakerNikonISOSelection	The ISO selection.
kCGImagePropertyMakerNikonISOSetting	The ISO setting.
kCGImagePropertyMakerNikonLensAdapter	The lens adapter.
kCGImagePropertyMakerNikonLensInfo	Lens information.
kCGImagePropertyMakerNikonLensType	The lens type.
kCGImagePropertyMakerNikonQuality	The quality setting.
kCGImagePropertyMakerNikonSharpenMode	The sharpening mode.
kCGImagePropertyMakerNikonShootingMode	The shooting mode.



<code>kCGImagePropertyMakerNikonShutterCount</code>	The shutter count.
<code>kCGImagePropertyMakerNikonWhiteBalanceMode</code>	The white balance mode.
<code>kCGImagePropertyMakerOlympusDictionary</code>	A dictionary of key-value pairs for an image from a Olympus camera.
<code>kCGImagePropertyMakerPentaxDictionary</code>	A dictionary of key-value pairs for an image from a Pentax camera.

## CoreGraphics

---

### CGColor.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGColorCreateGenericCMYK</code>	Creates a color in the Generic CMYK color space.
<code>CGColorCreateGenericGray</code>	Creates a color in the Generic gray color space.
<code>CGColorCreateGenericRGB</code>	Creates a color in the Generic RGB color space.
<code>CGColorGetConstantColor</code>	Returns a color object that represents a constant color.

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kCGColorBlack</code>	The black color in the Generic gray color space.
<code>kCGColorClear</code>	The clear color in the Generic gray color space.
<code>kCGColorWhite</code>	The white color in the Generic gray color space.

### CGColorSpace.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGColorSpaceCopyICCProfile</code>	Returns a copy of the ICC profile of the provided color space.
---	--

<code>CGColorSpaceGetBaseColorSpace</code>	Returns the base color space of a pattern or indexed color space.
<code>CGColorSpaceGetColorTable</code>	Copies the entries in the color table of an indexed color space.
<code>CGColorSpaceGetColorTableCount</code>	Returns the number of entries in the color table of an indexed color space.
<code>CGColorSpaceGetModel</code>	Returns the color space model of the provided color space.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGColorSpaceModel</code>	Models for color spaces.
<code>kCGColorSpaceAdobeRGB1998</code>	
<code>kCGColorSpaceGenericRGBLinear</code>	
<code>kCGColorSpaceModelCMYK</code>	A CMYK color space model.
<code>kCGColorSpaceModelDeviceN</code>	A DeviceN color space model.
<code>kCGColorSpaceModelIndexed</code>	An indexed color space model.
<code>kCGColorSpaceModelLab</code>	A Lab color space model.
<code>kCGColorSpaceModelMonochrome</code>	A monochrome color space model.
<code>kCGColorSpaceModelPattern</code>	A pattern color space model.
<code>kCGColorSpaceModelRGB</code>	An RGB color space model.
<code>kCGColorSpaceModelUnknown</code>	An unknown color space model.
<code>kCGColorSpaceSRGB</code>	

### CGContext.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGContextBeginTransparencyLayerWithRect</code>	Begins a transparency layer whose contents are bounded by the specified rectangle.
<code>CGContextDrawLinearGradient</code>	Paints a gradient fill that varies along the line defined by the provided starting and ending points.

<code>CGContextDrawRadialGradient</code>	Paints a gradient fill that varies along the area defined by the provided starting and ending circles.
<code>CGContextDrawTiledImage</code>	Repeatedly draws an image, scaled to the provided rectangle, to fill the current clip region.
<code>CGContextShowGlyphsAtPositions</code>	Draws glyphs at the provided position.

### Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kCGBlendModeClear</code>	$R = 0$
<code>kCGBlendModeCopy</code>	$R = S$
<code>kCGBlendModeDestinationAtop</code>	$R = S*(1 - D_a) + D*S_a$
<code>kCGBlendModeDestinationIn</code>	$R = D*S_a$
<code>kCGBlendModeDestinationOut</code>	$R = D*(1 - S_a)$
<code>kCGBlendModeDestinationOver</code>	$R = S*(1 - D_a) + D$
<code>kCGBlendModePlusDarker</code>	$R = \text{MAX}(0, (1 - D) + (1 - S))$
<code>kCGBlendModePlusLighter</code>	$R = \text{MIN}(1, S + D)$
<code>kCGBlendModeSourceAtop</code>	$R = S*D_a + D*(1 - S_a)$
<code>kCGBlendModeSourceIn</code>	$R = S*D_a$
<code>kCGBlendModeSourceOut</code>	$R = S*(1 - D_a)$
<code>kCGBlendModeXOR</code>	$R = S*(1 - D_a) + D*(1 - S_a)$ . This XOR mode is only nominally related to the classical bitmap XOR operation, which is not supported by Quartz 2D.

### CGDataProvider.h

#### Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGDataProviderCopyData</code>	Returns a copy of the provider's data.
<code>CGDataProviderCreateDirect</code>	Creates a Quartz direct-access data provider.
<code>CGDataProviderCreateSequential</code>	Creates a Quartz sequential-access data provider.

## Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGDataProviderDirectCallbacks</code>	Defines pointers to client-defined callback functions that manage the sending of data for a direct-access data provider.
<code>CGDataProviderGetBytesAtPositionCallback</code>	A callback function that copies data from the provider into a Quartz buffer.
<code>CGDataProviderSequentialCallbacks</code>	Defines a structure containing pointers to client-defined callback functions that manage the sending of data for a sequential-access data provider.
<code>CGDataProviderSkipForwardCallback</code>	A callback function that advances the current position in the data stream supplied by the provider.
<code>getBytesAtPosition</code>	
<code>skipForward</code>	

## CGDisplayConfiguration.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGDisplayCopyColorSpace</code>	Returns the color space for a display.
<code>CGDisplayRotation</code>	Returns the rotation angle of a display in degrees.

## Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kCGDisplayDesktopShapeChangedFlag</code>	The shape of the desktop (the union of display areas) has changed.
--	--

## CGEvent.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGEventCreateScrollWheelEvent</code>	Returns a new Quartz scrolling event.
<code>CGEventGetUnflippedLocation</code>	Returns the location of a Quartz mouse event.

## CGEventSource.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGEventSourceGetPixelsPerLine</code>	Gets the scale of pixels per line in a scrolling event source.
<code>CGEventSourceSetPixelsPerLine</code>	Sets the scale of pixels per line in a scrolling event source.

## CGEventTypes.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGScrollEventUnit</code>	Constants that specify the unit of measurement for a scrolling event.
<code>kCGEventTapOptionDefault</code>	Specifies that a new event tap is an active filter.
<code>kCGScrollEventUnitLine</code>	Specifies that the unit of measurement is lines.
<code>kCGScrollEventUnitPixel</code>	Specifies that the unit of measurement is pixels.
<code>kCGScrollWheelEventFixedPtDeltaAxis1</code>	Key to access a field that contains scrolling data. The scrolling data represents a line-based or pixel-based change in vertical position since the last scrolling event from a Mighty Mouse scroller or a single-wheel mouse scroller. The scrolling data uses a fixed-point 16.16 signed integer format. For example, if the field contains a value of 1.0, the integer 0x00010000 is returned by <code>CGEventGetIntegerValueField</code> . If this key is passed to <code>CGEventGetDoubleValueField</code> , the fixed-point value is converted to a double value.

kCGScrollWheelEventFixedPtDeltaAxis2	Key to access a field that contains scrolling data. The scrolling data represents a line-based or pixel-based change in horizontal position since the last scrolling event from a Mighty Mouse scroller. The scrolling data uses a fixed-point 16.16 signed integer format. For example, if the field contains a value of 1.0, the integer 0x00010000 is returned by CGEventGetIntegerValueField. If this key is passed to CGEventGetDoubleValueField, the fixed-point value is converted to a double value.
kCGScrollWheelEventFixedPtDeltaAxis3	This field is not used.
kCGScrollWheelEventIsContinuous	Key to access an integer field that indicates whether a scrolling event contains continuous, pixel-based scrolling data. The value is non-zero when the scrolling data is pixel-based and zero when the scrolling data is line-based.
kCGScrollWheelEventPointDeltaAxis1	Key to access an integer field that contains pixel-based scrolling data. The scrolling data represents the change in vertical position since the last scrolling event from a Mighty Mouse scroller or a single-wheel mouse scroller.
kCGScrollWheelEventPointDeltaAxis2	Key to access an integer field that contains pixel-based scrolling data. The scrolling data represents the change in horizontal position since the last scrolling event from a Mighty Mouse scroller.
kCGScrollWheelEventPointDeltaAxis3	This field is not used.

## CGFont.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGFontCopyFullName	Returns the full name associated with a font object.
CGFontCopyGlyphNameForGlyph	Returns the glyph name associated with a font object.
CGFontCopyTableForTag	Returns the font table that corresponds to the provided tag.
CGFontCopyTableTags	Returns an array of tags that correspond to the font tables for a font.
CGFontCreateWithDataProvider	Creates a font object from data supplied from a data provider.
CGFontCreateWithFontName	Creates a font object corresponding to the font specified by a PostScript or full name.
CGFontGetAscent	Returns the ascent of a font.

<code>CGFontGetCapHeight</code>	Returns the cap height of a font.
<code>CGFontGetDescent</code>	Returns the descent of a font.
<code>CGFontGetFontBBox</code>	Returns the bounding box of a font.
<code>CGFontGetGlyphAdvances</code>	Gets the bound box of each glyph in the provided array.
<code>CGFontGetGlyphBBoxes</code>	Get the bounding box of each glyph in an array.
<code>CGFontGetGlyphWithGlyphName</code>	Returns the glyph for the font name associated with the specified font object.
<code>CGFontGetItalicAngle</code>	Returns the italic angle of a font.
<code>CGFontGetLeading</code>	Returns the leading of a font.
<code>CGFontGetNumberOfGlyphs</code>	Returns the number of glyphs in a font.
<code>CGFontGetStemV</code>	Returns the thickness of the dominant vertical stems of glyphs in a font.
<code>CGFontGetUnitsPerEm</code>	Returns the number of glyph space units per em for the provided font.
<code>CGFontGetXHeight</code>	Returns the x-height of a font.

## CGGeometry.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGPointCreateDictionaryRepresentation</code>	Returns a dictionary representation of the provided point.
<code>CGPointMakeWithDictionaryRepresentation</code>	Fills in a <code>CGPoint</code> structure using the contents of the provided dictionary.
<code>CGRectCreateDictionaryRepresentation</code>	Returns a dictionary representation of the provided rectangle.
<code>CGRectMakeWithDictionaryRepresentation</code>	Fills in a <code>CGRect</code> structure using the contents of the provided dictionary.
<code>CGSizeCreateDictionaryRepresentation</code>	Returns a dictionary representation of the provided size.
<code>CGSizeMakeWithDictionaryRepresentation</code>	Fills in a <code>CGSize</code> structure using the contents of the provided dictionary.

## CGGradient.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGGradientCreateWithColorComponents</code>	Creates a <code>CGGradient</code> object from a color space and the provided color components and locations.
<code>CGGradientCreateWithColors</code>	Creates a <code>CGGradient</code> object from a color space and the provided color objects and locations.
<code>CGGradientGetTypeID</code>	Returns the Core Foundation type identifier for <code>CGGradient</code> objects.
<code>CGGradientRelease</code>	Decrements the retain count of a <code>CGGradient</code> object.
<code>CGGradientRetain</code>	Increments the retain count of a <code>CGGradient</code> object.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGGradientDrawingOptions</code>	Drawing locations for gradients.
<code>CGGradientRef</code>	An opaque type that represents a Quartz gradient.
<code>kCGGradientDrawsAfterEndLocation</code>	The fill should extend beyond the ending location. The color that extends beyond the ending point is the solid color defined by the <code>CGGradient</code> object to be at location 1.
<code>kCGGradientDrawsBeforeStartLocation</code>	The fill should extend beyond the starting location. The color that extends beyond the starting point is the solid color defined by the <code>CGGradient</code> object to be at location 0.

## CGPDFContext.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGPDFContextClose</code>	Closes a PDF document.
--------------------------------	------------------------



## Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kCGPDFContextEncryptionKeyLength	The encryption key length in bits; see Table 3.18 "Entries common to all encryption dictionaries", PDF Reference: Adobe PDF version 1.5 (4th ed.) for more information.
kCGPDFContextKeywords	The keywords for this document.
kCGPDFContextSubject	The subject of a document.

## CGWindow.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGWindowListCopyWindowInfo	Generates and returns information about the selected windows in the current user session.
CGWindowListCreate	Returns the list of window IDs associated with the specified windows in the current user session.
CGWindowListCreateDescriptionFromArray	Generates and returns information about windows with the specified window IDs.
CGWindowListCreateImage	Returns a composite image based on a dynamically generated list of windows.
CGWindowListCreateImageFromArray	Returns a composite image of the specified windows.

## Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGWindowBackingType	The data type used to specify the backing option for a given window.
CGWindowID	The data type used to store window identifiers.
CGWindowImageOption	The data type to use to specify the type of image to be generated for a window.
CGWindowListOption	The data type used to specify the options for gathering a list of windows.

<code>CGWindowSharingType</code>	The data type used to specify the sharing mode used by a window.
<code>kCGBackingStoreBuffered</code>	The window draws into a display buffer and then flushes that buffer to the screen.
<code>kCGBackingStoreNonretained</code>	The window draws directly to the screen without using any buffer.
<code>kCGBackingStoreRetained</code>	The window uses a buffer, but draws directly to the screen where possible and to the buffer for obscured portions.
<code>kCGWindowAlpha</code>	The value for this key is a <code>CFNumberRef</code> type (encoded as <code>kCFNumberFloatType</code> ) that contains the window's alpha fade level.
<code>kCGWindowBackingLocationVideoMemory</code>	The key that identifies whether the window's backing store is located in video memory.
<code>kCGWindowBounds</code>	The value for this key is a <code>CFDictionaryRef</code> type that must be decoded to a <code>CGRect</code> type using the <code>CGRectMakeWithDictionaryRepresentation</code> function.
<code>kCGWindowImageBoundsIgnoreFraming</code>	When the requested capture rectangle is <code>CGRectNull</code> , using this option captures the window area only and does not capture the area occupied by any window framing effects.
<code>kCGWindowImageDefault</code>	When the requested capture rectangle is <code>CGRectNull</code> , using this option captures the entire window plus the area required to display any framing effects, such as the window's shadow.
<code>kCGWindowImageOnlyShadows</code>	When capturing the window, only the shadow effects are captured.
<code>kCGWindowImageShouldBeOpaque</code>	When capturing the window, partially transparent areas are backed by a solid white color so that the resulting image is fully opaque.
<code>kCGWindowIsOnscreen</code>	The key that identifies whether the window is currently onscreen.
<code>kCGWindowLayer</code>	The value for this key is a <code>CFNumberRef</code> type (encoded as <code>kCFNumberIntType</code> ) that contains the window layer number.
<code>kCGWindowListExcludeDesktopElements</code>	Exclude any windows from the list that are elements of the desktop, including the background picture and desktop icons.
<code>kCGWindowListOptionAll</code>	List all windows, including both onscreen and offscreen windows.

<code>kCGWindowListOptionIncludingWindow</code>	Include the specified window (from the <code>relativeToWindow</code> parameter) in the returned list.
<code>kCGWindowListOptionOnScreenAboveWindow</code>	List all windows that are currently onscreen and in front of the window specified in the <code>relativeToWindow</code> parameter.
<code>kCGWindowListOptionOnScreenBelowWindow</code>	List all windows that are currently onscreen and in behind the window specified in the <code>relativeToWindow</code> parameter.
<code>kCGWindowListOptionOnScreenOnly</code>	List all windows that are currently onscreen.
<code>kCGWindowMemoryUsage</code>	The value for this key is a <code>CFNumberRef</code> type (encoded as <code>kCFNumberLongLongType</code> ) that contains an estimate of the amount of memory (measured in bytes) used by the window and its supporting data structures.
<code>kCGWindowName</code>	The key that identifies the name of the window, as configured in Quartz.
<code>kCGWindowNumber</code>	The value for this key is a <code>CFNumberRef</code> type (encoded as <code>kCGWindowIDCFNumberType</code> ) that contains the window ID.
<code>kCGWindowOwnerName</code>	The key that identifies the name of the application that owns the window.
<code>kCGWindowOwnerPID</code>	The value for this key is a <code>CFNumberRef</code> type (encoded as <code>kCFNumberIntType</code> ) that contains the process ID of the application that owns the window.
<code>kCGWindowSharingNone</code>	The window is not shared.
<code>kCGWindowSharingReadOnly</code>	The window is shared and its contents can be read by all processes but modified only by the process that created it.
<code>kCGWindowSharingReadWrite</code>	The window is shared and its contents can be read and modified by any process.
<code>kCGWindowSharingState</code>	The value for this key is a <code>CFNumberRef</code> type (encoded as <code>kCFNumberIntType</code> ) that contains one of the constants defined in “Window Sharing Constants.”
<code>kCGWindowStoreType</code>	The value for this key is a <code>CFNumberRef</code> type (encoded as <code>kCFNumberIntType</code> ) that contains one of the constants defined in “Backing Store Types.”
<code>kCGWindowWorkspace</code>	The key that identifies the workspace to which the window belongs.

## ColorSync

---

### CMAApplication.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CMSetPreferredCMM	
CWGetCMMSignature	

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

cmTextureRGBtoRGBX16	
cmTextureRGBtoRGBXFloat32	

### CMMComponent.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CMM_CheckBitmap	
CMM_CheckColors	
CMM_ConcatColorWorld	
CMM_CreateLinkProfile	
CMM_GetProperty	
CMM_MatchBitmap	
CMM_MatchColors	
CMM_MatchFloatBitmap	
CMM_ValidateProfile	
CWColorWorldGetProperty	
CWColorWorldSetProperty	

## 10.4 Symbol Changes

This article lists the symbols added to `ApplicationServices.framework` in Mac OS X v10.4.

### C Symbols

All of the header files with new symbols are listed alphabetically, with their new symbols described.

#### LaunchServices

##### LSInfo.h

##### Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>LSCopyAllHandlersForURLScheme</code>	Returns an array of application bundle identifiers for applications capable of handling the specified URL scheme.
<code>LSCopyAllRoleHandlersForContentType</code>	Returns an array of application bundle identifiers for applications capable of handling a specified content type with the specified roles.
<code>LSCopyDefaultHandlerForURLScheme</code>	Returns the application bundle identifier of the user's preferred default handler for the specified URL scheme.
<code>LSCopyDefaultRoleHandlerForContentType</code>	Returns the application bundle identifier of the user's preferred default handler for the specified content type with the specified role.
<code>LSCopyItemAttribute</code>	Obtains the value of an item's attribute.
<code>LSCopyItemAttributes</code>	Obtains multiple item attribute values as a dictionary.
<code>LSGetHandlerOptionsForContentType</code>	Gets the handler options for the specified content type.
<code>LSSetDefaultHandlerForURLScheme</code>	Sets the user's preferred default handler for the specified URL scheme.
<code>LSSetDefaultRoleHandlerForContentType</code>	Sets the user's preferred default handler for the specified content type in the specified roles.

<code>LSSetHandlerOptionsForContentType</code>	Sets the handler option for the specified content type.
--	---

### Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kLSExecutableIncorrectFormat</code>	
<code>kLSHandlerOptionsDefault</code>	When set, causes Launch Services to use a content item's creator (when available) to select a handler. This is the default setting.
<code>kLSHandlerOptionsIgnoreCreator</code>	When set, causes Launch Services to ignore the content item's creator when selecting a role handler for the specified content type.
<code>kLSItemContentType</code>	The item's content type identifier, which is a uniform type identifier string. The value type of this attribute is <code>CFStringRef</code> .
<code>kLSItemDisplayKind</code>	The localized kind string describing the item's type. The value type of this attribute is <code>CFStringRef</code> .
<code>kLSItemDisplayName</code>	The item's name as displayed to the user. The display name reflects localization and extension hiding that may be in effect. The value type of this attribute is <code>CFStringRef</code> .
<code>kLSItemExtension</code>	The item's filename extension. The value type of this attribute is <code>CFStringRef</code> .
<code>kLSItemExtensionIsHidden</code>	A value of <code>kCFBooleanTrue</code> if the item's extension is set to be hidden; otherwise, <code>kCFBooleanFalse</code> . The value type of this attribute is <code>CFBooleanRef</code> .
<code>kLSItemFileCreator</code>	The item's file creator (OSType). The value type of this attribute is <code>CFStringRef</code> .
<code>kLSItemFileType</code>	The item's file type (OSType). The value type of this attribute is <code>CFStringRef</code> .
<code>kLSItemIsInvisible</code>	A value of <code>kCFBooleanTrue</code> if the item is normally hidden from users; otherwise, <code>kCFBooleanFalse</code> . The value type of this attribute is <code>CFBooleanRef</code> .
<code>kLSItemRoleHandlerDisplayName</code>	The display name of the application that is set to handle this item, subject to the role mask. The value type of this attribute is <code>CFStringRef</code> .
<code>kLSRolesShell</code>	Requests the role Shell (the application can execute the item).
<code>LShandlerOptions</code>	Specify the options for controlling how content handlers are selected.

## LSOpen.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

LSOpenApplication	Launches the specified application.
LSOpenItemsWithRole	Opens items specified as an array of values of type FSRef with a specified role.
LSOpenURLsWithRole	Opens one or more URLs with the specified roles.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

application	
asyncLaunchRefCon	
environment	
initialEvent	
kLSLaunchAndDisplayErrors	Requests that launch and open failures be displayed in the UI.
kLSLaunchHasUntrustedContents	Requests that the items to be launched should be marked as untrusted.
LSApplicationParameters	Specifies the application, launch flags, and additional parameters that control how an application is launched.

## UTCoreTypes.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kUTTypeAliasFile	
kUTTypeAliasRecord	
kUTTypeAppleICNS	
kUTTypeAppleProtectedMPEG4Audio	

kUTTypeApplication	
kUTTypeApplicationBundle	
kUTTypeApplicationFile	
kUTTypeArchive	
kUTTypeAudio	
kUTTypeAudiovisualContent	
kUTTypeBMP	
kUTTypeBundle	
kUTTypeCHheader	
kUTTypeCompositeContent	
kUTTypeContact	
kUTTypeContent	
kUTTypeCPlusPlusHeader	
kUTTypeCPlusPlusSource	
kUTTypeCSource	
kUTTypeData	
kUTTypeDirectory	
kUTTypeDiskImage	
kUTTypeFileURL	
kUTTypeFlatRTFD	
kUTTypeFolder	
kUTTypeFramework	
kUTTypeGIF	
kUTTypeHTML	
kUTTypeICO	
kUTTypeImage	
kUTTypeInkText	
kUTTypeItem	



kUTTypeJavaSource	
kUTTypeJPEG	
kUTTypeJPEG2000	
kUTTypeMessage	
kUTTypeMountPoint	
kUTTypeMovie	
kUTTypeMP3	
kUTTypeMPEG	
kUTTypeMPEG4	
kUTTypeMPEG4Audio	
kUTTypeObjectiveCPlusPlusSource	
kUTTypeObjectiveCSource	
kUTTypePackage	
kUTTypePDF	
kUTTypePICT	
kUTTypePlainText	
kUTTypePNG	
kUTTypeQuickTimeImage	
kUTTypeQuickTimeMovie	
kUTTypeResolvable	
kUTTypeRTF	
kUTTypeRTFD	
kUTTypeSourceCode	
kUTTypeSymLink	
kUTTypeText	
kUTTypeTIFF	
kUTTypeTXNTextAndMultimediaData	
kUTTypeURL	

kUTTypeUTF16ExternalPlainText	
kUTTypeUTF16PlainText	
kUTTypeUTF8PlainText	
kUTTypeVCard	
kUTTypeVideo	
kUTTypeVolume	
kUTTypeWebArchive	
kUTTypeXML	

## ATS

---

### SFNTLayoutTypes.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kCJKItalicRomanOffSelector	
kCJKItalicRomanOnSelector	
kRubyKanaOffSelector	
kRubyKanaOnSelector	

## QD

---

### Fonts.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

FMFontGetCGFontRefFromFontFamilyInstance	Obtains the Quartz font associated with a typeface from a QuickDraw font family.
--	--

## Quickdraw.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

IsValidRgnHandle	
------------------	--

## HIServices

---

### AXActionConstants.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kAXShowMenuAction	
-------------------	--

### AXAttributeConstants.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kAXAllowedValuesAttribute	An array of the allowed values for an accessibility object.
kAXAttributedStringForRangeParameterizedAttribute	The CFAttributedStringType representation of the text associated with this accessibility object that is specified by the given range.
kAXClearButtonAttribute	
kAXColumnHeaderUIElementsAttribute	An array of accessibility objects representing the column headers of this table or browser view.
kAXColumnNameAttribute	The title of the column element represented by this accessibility object.
kAXDescription	
kAXDescriptionAttribute	

kAXDisclosureLevelAttribute	
kAXIndexAttribute	The index of the row or column represented by this accessibility object.
kAXInsertionPointLineNumberAttribute	The line number of the insertion point in the text associated with this accessibility object.
kAXIsApplicationRunningAttribute	Indicates if the application represented by the Dock icon this accessibility object represents is currently running.
kAXLabelUIElementsAttribute	An array of accessibility objects representing the labels displayed near the control represented by this accessibility object.
kAXLabelValueAttribute	The value of the label represented by this accessibility object. This attribute is required for all accessibility objects that represent labels.
kAXLinkedUIElementsAttribute	An array of accessibility objects with which this accessibility object is related.
kAXMarkerTypeAttribute	
kAXMarkerTypeDescriptionAttribute	
kAXMarkerUIElementsAttribute	
kAXMatteContentUIElementAttribute	The accessibility object clipped by the matte.
kAXMatteHoleAttribute	The accessibility object that represents the area available to the user through the matte hole.
kAXMenuItemPrimaryUIElementAttribute	The accessibility object representing the primary menu item in a group of dynamic menu items.
kAXSearchButtonAttribute	
kAXServesAsTitleForUIElementsAttribute	An array of accessibility objects for which this accessibility object serves as the title.
kAXSharedCharacterRangeAttribute	The portion of shared text this accessibility object currently displays.
kAXSharedTextUIElementsAttribute	An array of accessibility objects with which the text of this accessibility object is shared.

kAXShownMenuUIElementAttribute	An array of accessibility objects that represent the contextual or Dock menus provided by this accessibility object.
kAXTopLevelUIElementAttribute	The window, sheet, or drawer element that contains this accessibility object.
kAXUnitDescriptionAttribute	
kAXUnitsAttribute	

## AXNotificationConstants.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kAXCreatedNotification	
kAXHelpTagCreatedNotification	
kAXMovedNotification	
kAXResizedNotification	
kAXRowCountChangedNotification	
kAXSelectedChildrenChangedNotification	
kAXSelectedColumnsChangedNotification	
kAXSelectedRowsChangedNotification	
kAXSelectedTextChangedNotification	

## AXRoleConstants.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kAXApplicationDockItemSubrole	An icon in the Dock that represents an application.
kAXDisclosureTriangleRole	A disclosure triangle control.
kAXDockExtraDockItemSubrole	An icon in the Dock that represents a Dock Extra.
kAXDockItemRole	An icon that represents an item in the Dock.

kAXDocumentDockItemSubrole	An icon in the Dock that represents a document.
kAXFolderDockItemSubrole	An icon in the Dock that represents a folder.
kAXHelpTagRole	A help tag.
kAXMatteRole	The outer view that represents the entire contents, including the view through the matte hole, the contents hidden by the matte frame, and the resizing and repositioning controls.
kAXMinimizedWindowDockItemSubrole	An icon in the Dock that represents a minimized window.
kAXProcessSwitcherListSubrole	The display of running applications (processes) that appears when a user presses Command-Tab.
kAXRulerMarkerRole	
kAXRulerRole	
kAXSearchFieldSubrole	A search field.
kAXSortButtonSubrole	A column heading button in a list or column view.
kAXTrashDockItemSubrole	The icon in the Dock that represents the Trash.
kAXURLDockItemSubrole	An icon in the Dock that represents a URL.

## AXTextAttributedString.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

AXUnderlineStyle	
kAXAttachmentTextAttribute	
kAXBackgroundColorTextAttribute	
kAXFontFamilyKey	
kAXFontNameKey	
kAXFontSizeKey	
kAXFontTextAttribute	
kAXForegroundColorTextAttribute	
kAXForegroundColorTextAttribute	
kAXLinkTextAttribute	

kAXMisspelledTextAttribute	
kAXNaturalLanguageTextAttribute	
kAXReplacementStringTextAttribute	
kAXShadowTextAttribute	
kAXStrikethroughColorTextAttribute	
kAXStrikethroughTextAttribute	
kAXSuperscriptTextAttribute	
kAXUnderlineColorTextAttribute	
kAXUnderlineStyleDouble	
kAXUnderlineStyleNone	
kAXUnderlineStyleSingle	
kAXUnderlineStyleThick	
kAXUnderlineTextAttribute	
kAXVisibleNameKey	

## AXUIElement.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

AXIsProcessTrusted	
AXMakeProcessTrusted	
AXUIElementCopyMultipleAttributeValues	
AXUIElementSetMessagingTimeout	

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

AXCopyMultipleAttributeOptions	
kAXCopyMultipleAttributeOptionStopOnError	

## AXValue.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kAXValueAXErrorType	
---------------------	--

## AXValueConstants.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kAXAscendingSortDirectionValue	
--------------------------------	--

kAXDescendingSortDirectionValue	
---------------------------------	--

kAXUnknownOrientationValue	
----------------------------	--

kAXUnknownSortDirectionValue	
------------------------------	--

## Pasteboard.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

PasteboardCopyName	
--------------------	--

Gets the name of a pasteboard.	
--------------------------------	--

## UniversalAccess.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

UAZoomChangeFocus	
-------------------	--

UAZoomEnabled	
---------------	--



## Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kUAZoomFocusTypeInsertionPoint	
kUAZoomFocusTypeOther	
UAZoomChangeFocusType	

## PrintCore

---

### PMCore.h

---

## Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

PMFlattenPageFormatToCFData	Flattens a page format object into a Core Foundation data object for storage in a user document.
PMFlattenPageFormatToURL	Flattens a page format object into a file for storage in a user document.
PMFlattenPrintSettingsToCFData	Flattens a print settings object into a Core Foundation data object for storage in a user document.
PMFlattenPrintSettingsToURL	Flattens a print settings object into a URL for storage in a user document.
PMGetDuplex	Obtains the selected duplex mode.
PMPrinterCopyDescriptionURL	Obtains the URL of the description file for a given printer.
PMPrinterCopyDeviceURI	Obtains the device URI of a given printer.
PMPrinterCreateFromPrinterID	Creates a printer object from a print queue identifier.
PMPrintSettingsGetJobName	Obtains the name of a print job.
PMPrintSettingsGetValue	Obtains the value of a setting in a print settings object.
PMPrintSettingsSetJobName	Specifies the name of a print job.
PMPrintSettingsSetValue	Stores the value of a setting in a print settings object.
PMSessionBeginCGDocumentNoDialog	Begins a print job that draws into a Quartz graphics context and suppresses the printing status dialog.

<code>PMSessionGetCGGraphicsContext</code>	Obtains the Quartz graphics context for the current page in a printing session.
<code>PMSetDuplex</code>	Sets the duplex mode.
<code>PMUnflattenPageFormatWithCFData</code>	Rebuilds a page format object from a Core Foundation data object that contains flattened page format data.
<code>PMUnflattenPageFormatWithURL</code>	Rebuilds a page format object from a file system URL that contains flattened page format data.
<code>PMUnflattenPrintSettingsWithCFData</code>	Rebuilds a print settings object from a Core Foundation data object that contains flattened print settings data.
<code>PMUnflattenPrintSettingsWithURL</code>	Rebuilds a print settings object from a file that contains flattened print settings data.

## PMDefinitions.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kPMDestinationProcessPDF</code>	Specifies output to a PDF workflow option.
<code>PMDuplexMode</code>	Constants that specify duplex mode settings.

## PMIOModule.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kPMDNSSDConnection</code>	
---------------------------------	--

## PMTicket.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>PMTicketToXMLURL</code>	
-------------------------------	--

**Data Types & Constants**

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kPMApplIcationColorMatching	
kPMApplIcationColorMatchingStr	
kPMColorMatchingModeKey	
kPMColorMatchingModeStr	
kPMColorSyncMatching	
kPMColorSyncMatchingStr	
kPMCustomPaperKey	
kPMCustomPaperStr	
kPMFaxDateLabelKey	
kPMFaxDateLabelStr	
kPMFaxFromLabelKey	
kPMFaxFromLabelStr	
kPMFaxSheetsLabelKey	
kPMFaxSheetsLabelStr	
kPMFaxSubjectLabelKey	
kPMFaxSubjectLabelStr	
kPMFaxToLabelKey	
kPMFaxToLabelStr	
kMPPageToPaperMappingAllowScalingUpKey	
kMPPageToPaperMappingAllowScalingUpStr	
kMPPageToPaperMappingNone	
kMPPageToPaperMappingScaleToFit	
kMPPageToPaperMappingTypeKey	
kMPPageToPaperMappingTypeStr	
kMPPageToPaperMediaNameKey	
kMPPageToPaperMediaNameStr	

kPMSupportsVendorMatchingModeKey	
kPMSupportsVendorMatchingModeStr	
kPMVendorColorMatching	
kPMVendorColorMatchingStr	
PMPageToPaperMappingType	

## CoreGraphics

---

### CGAffineTransform.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGAffineTransformEqualToTransform	Checks whether two affine transforms are equal.
CGAffineTransformIsIdentity	Checks whether an affine transform is the identity transform.
CGRectApplyAffineTransform	Applies an affine transform to a rectangle.

### CGBase.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CG_OBSOLETE	
define	

### CGBitmapContext.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGBitmapContextCreateImage	Creates and returns a Quartz image from the pixel data in a bitmap graphics context.
----------------------------	--

<code>CGBitmapContextGetBitmapInfo</code>	Obtains the bitmap information associated with a bitmap graphics context.
---	---

## CGColorSpace.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kCGColorSpaceGenericCMYK</code>	
<code>kCGColorSpaceGenericGray</code>	
<code>kCGColorSpaceGenericRGB</code>	

## CGContext.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGContextAddEllipseInRect</code>	Adds an ellipse that fits inside the specified rectangle.
<code>CGContextClipToMask</code>	Maps a mask into the specified rectangle and intersects it with the current clipping area of the graphics context.
<code>CGContextConvertPointToDeviceSpace</code>	Returns a point that is transformed from user space coordinates to device space coordinates.
<code>CGContextConvertPointToUserSpace</code>	Returns a point that is transformed from device space coordinates to user space coordinates.
<code>CGContextConvertRectToDeviceSpace</code>	Returns a rectangle that is transformed from user space coordinate to device space coordinates.
<code>CGContextConvertRectToUserSpace</code>	Returns a rectangle that is transformed from device space coordinate to user space coordinates.
<code>CGContextConvertSizeToDeviceSpace</code>	Returns a size that is transformed from user space coordinates to device space coordinates.
<code>CGContextConvertSizeToUserSpace</code>	Returns a size that is transformed from device space coordinates to user space coordinates.

<code>CGContextFillEllipseInRect</code>	Paints the area of the ellipse that fits inside the provided rectangle, using the fill color in the current graphics state.
<code>CGContextGetUserSpaceToDeviceSpaceTransform</code>	Returns an affine transform that maps user space coordinates to device space coordinates.
<code>CGContextPathContainsPoint</code>	Checks to see whether the specified point is contained in the current path.
<code>CGContextReplacePathWithStrokedPath</code>	Replaces the path in the graphics context with the stroked version of the path.
<code>CGContextSetAllowsAntialiasing</code>	Sets whether or not to allow anti-aliasing for a graphics context.
<code>CGContextSetBlendMode</code>	Sets how Quartz composites sample values for a graphics context.
<code>CGContextStrokeEllipseInRect</code>	Strokes an ellipse that fits inside the specified rectangle.
<code>CGContextStrokeLineSegments</code>	Strokes a sequence of line segments.

### Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGBlendMode</code>	Compositing operations for images.
<code>kCGBlendModeColor</code>	Uses the luminance values of the background with the hue and saturation values of the source image. This mode preserves the gray levels in the image. You can use this mode to color monochrome images or to tint color images.
<code>kCGBlendModeColorBurn</code>	Darkens the background image samples to reflect the source image samples. Source image sample values that specify white do not produce a change.
<code>kCGBlendModeColorDodge</code>	Brightens the background image samples to reflect the source image samples. Source image sample values that specify black do not produce a change.
<code>kCGBlendModeDarken</code>	Creates the composite image samples by choosing the darker samples (either from the source image or the background).
<code>kCGBlendModeDifference</code>	Subtracts either the source image sample color from the background image sample color, or the reverse, depending on which sample has the greater brightness value.
<code>kCGBlendModeExclusion</code>	Produces an effect similar to that produced by <code>kCGBlendModeDifference</code> , but with lower contrast. Source image sample values that are black don't produce a change; white inverts the background color values.

<code>kCGBlendModeHardLight</code>	Either multiplies or screens colors, depending on the source image sample color.
<code>kCGBlendModeHue</code>	Uses the luminance and saturation values of the background with the hue of the source image.
<code>kCGBlendModeLighten</code>	Creates the composite image samples by choosing the lighter samples (either from the source image or the background).
<code>kCGBlendModeLuminosity</code>	Uses the hue and saturation of the background with the luminance of the source image. This mode creates an effect that is inverse to the effect created by <code>kCGBlendModeColor</code> .
<code>kCGBlendModeMultiply</code>	Multiplies the source image samples with the background image samples. This results in colors that are at least as dark as either of the two contributing sample colors.
<code>kCGBlendModeNormal</code>	Paints the source image samples over the background image samples.
<code>kCGBlendModeOverlay</code>	Either multiplies or screens the source image samples with the background image samples, depending on the background color.
<code>kCGBlendModeSaturation</code>	Uses the luminance and hue values of the background with the saturation of the source image. Areas of the background that have no saturation (that is, pure gray areas) don't produce a change.
<code>kCGBlendModeScreen</code>	Multiplies the inverse of the source image samples with the inverse of the background image samples. This results in colors that are at least as light as either of the two contributing sample colors.
<code>kCGBlendModeSoftLight</code>	Either darkens or lightens colors, depending on the source image sample color.

## CGDataConsumer.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGDataConsumerCreateWithCFData</code>	Creates a data consumer that writes to a <code>CFData</code> object.
---	--

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGDataConsumerPutBytesCallback</code>	Copies data from a Quartz-supplied buffer into a data consumer.
---	---

<code>CGDataConsumerReleaseInfoCallback</code>	Releases any private data or resources associated with the data consumer.
--	---

## CGDataProvider.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGDataProviderCreateWithCFData</code>	Creates a Quartz data provider that reads from a CFData object.
---	---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGDataProviderReleaseDataCallback</code>	A callback function that releases data you supply to the function <code>CGDataProviderCreateWithData</code> .
--	---

## CGDisplayConfiguration.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGConfigureDisplayStereoOperation</code>	Enables or disables stereo operation for a display, as part of a display configuration.
<code>CGDisplayIsStereo</code>	Returns a Boolean value indicating whether a display is running in a stereo graphics mode.
<code>CGDisplaySetStereoOperation</code>	Immediately enables or disables stereo operation for a display.

## CGError.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kCGErrorClassicApplicationsMustBeLaunchedByClassic</code>	
<code>kCGErrorForkFailed</code>	



## CGEvent.h

## Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGEventCreate	Returns a new Quartz event.
CGEventCreateCopy	Returns a copy of an existing Quartz event.
CGEventCreateData	Returns a flattened data representation of a Quartz event.
CGEventCreateFromData	Returns a Quartz event created from a flattened data representation of the event.
CGEventCreateKeyboardEvent	Returns a new Quartz keyboard event.
CGEventCreateMouseEvent	Returns a new Quartz mouse event.
CGEventCreateSourceFromEvent	Returns a Quartz event source created from an existing Quartz event.
CGEventGetDoubleValueField	Returns the floating-point value of a field in a Quartz event.
CGEventGetFlags	Returns the event flags of a Quartz event.
CGEventGetIntegerValueField	Returns the integer value of a field in a Quartz event.
CGEventGetLocation	Returns the location of a Quartz mouse event.
CGEventGetSource	Returns a Quartz event source created from an existing Quartz event.
CGEventGetTimestamp	Returns the timestamp of a Quartz event.
CGEventGetType	Returns the event type of a Quartz event (left mouse down, for example).
CGEventGetTypeID	Returns the type identifier for the opaque type CGEventRef.
CGEventKeyboardGetUnicodeString	Returns the Unicode string associated with a Quartz keyboard event.
CGEventKeyboardSetUnicodeString	Sets the Unicode string associated with a Quartz keyboard event.
CGEventPost	Posts a Quartz event into the event stream at a specified location.
CGEventPostToPSN	Posts a Quartz event into the event stream for a specific application.
CGEventSetDoubleValueField	Sets the floating-point value of a field in a Quartz event.

<code>CGEventSetFlags</code>	Sets the event flags of a Quartz event.
<code>CGEventSetIntegerValueField</code>	Sets the integer value of a field in a Quartz event.
<code>CGEventSetLocation</code>	Sets the location of a Quartz mouse event.
<code>CGEventSetSource</code>	Sets the event source of a Quartz event.
<code>CGEventSetTimestamp</code>	Sets the timestamp of a Quartz event.
<code>CGEventSetType</code>	Sets the event type of a Quartz event (left mouse down, for example).
<code>CGEventTapCreate</code>	Creates an event tap.
<code>CGEventTapCreateForPSN</code>	Creates an event tap for a specified process.
<code>CGEventTapEnable</code>	Enables or disables an event tap.
<code>CGEventTapIsEnabled</code>	Returns a Boolean value indicating whether an event tap is enabled.
<code>CGEventTapPostEvent</code>	Posts a Quartz event from an event tap into the event stream.
<code>CGGetEventTapList</code>	Gets a list of currently installed event taps.

## CGEventSource.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGEventSourceButtonState</code>	Returns a Boolean value indicating the current button state of a Quartz event source.
<code>CGEventSourceCounterForEventType</code>	Returns a count of events of a given type seen since the window server started.
<code>CGEventSourceCreate</code>	Returns a Quartz event source created with a specified source state.
<code>CGEventSourceFlagsState</code>	Returns the current flags of a Quartz event source.
<code>CGEventSourceGetKeyboardType</code>	Returns the keyboard type to be used with a Quartz event source.
<code>CGEventSourceGetLocalEventsFilter-DuringSuppressionState</code>	Returns the mask that indicates which classes of local hardware events are enabled during event suppression.

<code>CGEventSourceGetLocalEventsSuppressionInterval</code>	Returns the interval that local hardware events may be suppressed following the posting of a Quartz event.
<code>CGEventSourceGetSourceStateID</code>	Returns the source state associated with a Quartz event source.
<code>CGEventSourceGetTypeID</code>	Returns the type identifier for the opaque type <code>CGEventSourceRef</code> .
<code>CGEventSourceGetUserData</code>	Returns the 64-bit user-specified data for a Quartz event source.
<code>CGEventSourceKeyState</code>	Returns a Boolean value indicating the current keyboard state of a Quartz event source.
<code>CGEventSourceSecondsSinceLastEventType</code>	Returns the elapsed time since the last event for a Quartz event source.
<code>CGEventSourceSetKeyboardType</code>	Sets the keyboard type to be used with a Quartz event source.
<code>CGEventSourceSetLocalEventsFilter-DuringSuppressionState</code>	Sets the mask that indicates which classes of local hardware events are enabled during event suppression.
<code>CGEventSourceSetLocalEventsSuppressionInterval</code>	Sets the interval that local hardware events may be suppressed following the posting of a Quartz event.
<code>CGEventSourceSetUserData</code>	Sets the 64-bit user-specified data for a Quartz event source.

## CGEventTypes.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>avgUsecLatency</code>	
<code>CGEventField</code>	Constants used as keys to access specialized fields in low-level events.
<code>CGEventFlags</code>	Constants that indicate the modifier key state at the time an event is created, as well as other event-related states.

<code>CGEventMask</code>	Defines a mask that identifies the set of Quartz events to be observed in an event tap.
<code>CGEventMaskBit</code>	
<code>CGEventMouseSubtype</code>	Constants used with the <code>kCGMouseEventSubtype</code> event field.
<code>CGEventRef</code>	Defines an opaque type that represents a low-level hardware event.
<code>CGEventSourceKeyboardType</code>	Defines a code that represents the type of keyboard used with a specified event source.
<code>CGEventSourceRef</code>	Defines an opaque type that represents the source of a Quartz event.
<code>CGEventSourceStateID</code>	Constants that specify the possible source states of an event source.
<code>CGEventTapCallback</code>	A client-supplied callback function that's invoked whenever an associated event tap receives a Quartz event.
<code>CGEventTapLocation</code>	Constants that specify possible tapping points for events.
<code>CGEventTapOptions</code>	Constants that specify whether a new event tap is an active filter or a passive listener.
<code>CGEventTapPlacement</code>	Constants that specify where a new event tap is inserted into the list of active event taps.
<code>CGEventTapProxy</code>	Defines an opaque type that represents state within the client application that's associated with an event tap.
<code>CGEventTimestamp</code>	Defines the elapsed time in nanoseconds since startup that a Quartz event occurred.
<code>CGEventType</code>	Constants that specify the different types of input events.
<code>CGMouseButton</code>	Constants that specify buttons on a one, two, or three-button mouse.
<code>eventsOfInterest</code>	
<code>eventTapID</code>	

<code>kCGAnnotatedSessionEventTap</code>	Specifies that an event tap is placed at the point where session events have been annotated to flow to an application.
<code>kCGAnyInputEventType</code>	A constant that specifies any input event type.
<code>kCGEventFlagMaskAlphaShift</code>	Indicates that the Caps Lock key is down for a keyboard, mouse, or flag-changed event.
<code>kCGEventFlagMaskAlternate</code>	Indicates that the Alt or Option key is down for a keyboard, mouse, or flag-changed event.
<code>kCGEventFlagMaskCommand</code>	Indicates that the Command key is down for a keyboard, mouse, or flag-changed event.
<code>kCGEventFlagMaskControl</code>	Indicates that the Control key is down for a keyboard, mouse, or flag-changed event.
<code>kCGEventFlagMaskHelp</code>	Indicates that the Help modifier key is down for a keyboard, mouse, or flag-changed event. This key is not present on most keyboards, and is different than the Help key found in the same row as Home and Page Up.
<code>kCGEventFlagMaskNonCoalesced</code>	Indicates that mouse and pen movement events are not being coalesced.
<code>kCGEventFlagMaskNumericPad</code>	Identifies key events from the numeric keypad area on extended keyboards.
<code>kCGEventFlagMaskSecondaryFn</code>	Indicates that the Fn (Function) key is down for a keyboard, mouse, or flag-changed event. This key is found primarily on laptop keyboards.
<code>kCGEventFlagMaskShift</code>	Indicates that the Shift key is down for a keyboard, mouse, or flag-changed event.
<code>kCGEventFlagsChanged</code>	Specifies a key changed event for a modifier or status key.
<code>kCGEventKeyDown</code>	Specifies a key down event.
<code>kCGEventKeyUp</code>	Specifies a key up event.
<code>kCGEventLeftMouseDown</code>	Specifies a mouse down event with the left button.

<code>kCGEventLeftMouseDown</code>	Specifies a mouse drag event with the left button down.
<code>kCGEventLeftMouseUp</code>	Specifies a mouse up event with the left button.
<code>kCGEventMaskForAllEvents</code>	An event mask that specifies all event types.
<code>kCGEventMouseMove</code>	Specifies a mouse moved event.
<code>kCGEventMouseSubtypeDefault</code>	Specifies that the event is an ordinary mouse event, and does not contain additional tablet device information.
<code>kCGEventMouseSubtypeTabletPoint</code>	Specifies that the mouse event originated from a tablet device, and that the various <code>kCGTabletEvent</code> field selectors may be used to obtain tablet-specific data from the mouse event.
<code>kCGEventMouseSubtypeTabletProximity</code>	Specifies that the mouse event originated from a tablet device with the pen in proximity but not necessarily touching the tablet, and that the various <code>kCGTabletProximity</code> field selectors may be used to obtain tablet-specific data from the mouse event.
<code>kCGEventNull</code>	Specifies a null event.
<code>kCGEventOtherMouseDown</code>	Specifies a mouse down event with one of buttons 2-31.
<code>kCGEventOtherMouseDownDragged</code>	Specifies a mouse drag event with one of buttons 2-31 down.
<code>kCGEventOtherMouseUp</code>	Specifies a mouse up event with one of buttons 2-31.
<code>kCGEventRightMouseDown</code>	Specifies a mouse down event with the right button.
<code>kCGEventRightMouseDownDragged</code>	Specifies a mouse drag event with the right button down.
<code>kCGEventRightMouseUp</code>	Specifies a mouse up event with the right button.
<code>kCGEventScrollWheel</code>	Specifies a scroll wheel moved event.
<code>kCGEventSourceGroupID</code>	Key to access a field that contains the event source Unix effective GID.

kCGEventSourceStateCombinedSessionState	Specifies that an event source should use the event state table that reflects the combined state of all event sources posting to the current user login session.
kCGEventSourceStateHIDSystemState	Specifies that an event source should use the event state table that reflects the combined state of all hardware event sources posting from the HID system.
kCGEventSourceStateID	Key to access a field that contains the event source state ID used to create this event.
kCGEventSourceStatePrivate	Specifies that an event source should use a private event state table.
kCGEventSourceUnixProcessID	Key to access a field that contains the event source Unix process ID.
kCGEventSourceUserData	Key to access a field that contains the event source user-supplied data, up to 64 bits.
kCGEventSourceUserID	Key to access a field that contains the event source Unix effective UID.
kCGEventTabletPointer	Specifies a tablet pointer event.
kCGEventTabletProximity	Specifies a tablet proximity event.
kCGEventTapDisabledByTimeout	Specifies an event indicating the event tap is disabled because of timeout.
kCGEventTapDisabledByUserInput	Specifies an event indicating the event tap is disabled because of user input.
kCGEventTapOptionListenOnly	Specifies that a new event tap is a passive listener.
kCGEventTargetProcessSerialNumber	Key to access a field that contains the event target process serial number. The value is a 64-bit long word.
kCGEventTargetUnixProcessID	Key to access a field that contains the event target Unix process ID.
kCGHeadInsertEventTap	Specifies that a new event tap should be inserted before any pre-existing event taps at the same location.
kCGHIDEventTap	Specifies that an event tap is placed at the point where HID system events enter the window server.

<code>kCGKeyboardEventAutorepeat</code>	Key to access an integer field, non-zero when this is an autorepeat of a key-down, and zero otherwise.
<code>kCGKeyboardEventKeyboardType</code>	Key to access an integer field that contains the keyboard type identifier.
<code>kCGKeyboardEventKeycode</code>	Key to access an integer field that contains the virtual keycode of the key-down or key-up event.
<code>kCGMouseButtonCenter</code>	Specifies the center mouse button on a three-button mouse.
<code>kCGMouseButtonLeft</code>	Specifies the only mouse button on a one-button mouse, or the left mouse button on a two-button or three-button mouse.
<code>kCGMouseButtonRight</code>	Specifies the right mouse button on a two-button or three-button mouse.
<code>kCGMouseEventButtonNumber</code>	Key to access an integer field that contains the mouse button number. For information about the possible values, see “Mouse Buttons.”
<code>kCGMouseEventClickState</code>	Key to access an integer field that contains the mouse button click state. A click state of 1 represents a single click. A click state of 2 represents a double-click. A click state of 3 represents a triple-click.
<code>kCGMouseEventDeltaX</code>	Key to access an integer field that contains the horizontal mouse delta since the last mouse movement event.
<code>kCGMouseEventDeltaY</code>	Key to access an integer field that contains the vertical mouse delta since the last mouse movement event.
<code>kCGMouseEventInstantMouser</code>	Key to access an integer field. The value is non-zero if the event should be ignored by the Inkwell subsystem.
<code>kCGMouseEventNumber</code>	Key to access an integer field that contains the mouse button event number. Matching mouse-down and mouse-up events will have the same event number.



kCGMouseEventPressure	Key to access a double field that contains the mouse button pressure. The pressure value may range from 0 to 1, with 0 representing the mouse being up. This value is commonly set by tablet pens mimicking a mouse.
kCGMouseEventSubtype	Key to access an integer field that encodes the mouse event subtype as a kCFNumberIntType.
kCGNotifyEventTapAdded	
kCGNotifyEventTapRemoved	
kCGScrollWheelEventDeltaAxis1	Key to access an integer field that contains scrolling data. This field typically contains the change in vertical position since the last scrolling event from a Mighty Mouse scroller or a single-wheel mouse scroller.
kCGScrollWheelEventDeltaAxis2	Key to access an integer field that contains scrolling data. This field typically contains the change in horizontal position since the last scrolling event from a Mighty Mouse scroller.
kCGScrollWheelEventDeltaAxis3	This field is not used.
kCGScrollWheelEventInstantMouser	Key to access an integer field that indicates whether the event should be ignored by the Inkwell subsystem. If the value is non-zero, the event should be ignored.
kCGSessionEventTap	Specifies that an event tap is placed at the point where HID system and remote control events enter a login session.
kCGTabletEventDeviceID	Key to access an integer field that contains the system-assigned unique device ID.
kCGTabletEventPointButtons	Key to access an integer field that contains the tablet button state. Bit 0 is the first button, and a set bit represents a closed or pressed button. Up to 16 buttons are supported.
kCGTabletEventPointPressure	Key to access a double field that contains the tablet pen pressure. A value of 0.0 represents no pressure, and 1.0 represents maximum pressure.

<code>kCGTabletEventPointX</code>	Key to access an integer field that contains the absolute X coordinate in tablet space at full tablet resolution.
<code>kCGTabletEventPointY</code>	Key to access an integer field that contains the absolute Y coordinate in tablet space at full tablet resolution.
<code>kCGTabletEventPointZ</code>	Key to access an integer field that contains the absolute Z coordinate in tablet space at full tablet resolution.
<code>kCGTabletEventRotation</code>	Key to access a double field that contains the tablet pen rotation.
<code>kCGTabletEventTangentialPressure</code>	Key to access a double field that contains the tangential pressure on the device. A value of 0.0 represents no pressure, and 1.0 represents maximum pressure.
<code>kCGTabletEventTiltX</code>	Key to access a double field that contains the horizontal tablet pen tilt. A value of 0.0 represents no tilt, and 1.0 represents maximum tilt.
<code>kCGTabletEventTiltY</code>	Key to access a double field that contains the vertical tablet pen tilt. A value of 0.0 represents no tilt, and 1.0 represents maximum tilt.
<code>kCGTabletEventVendor1</code>	Key to access an integer field that contains a vendor-specified value.
<code>kCGTabletEventVendor2</code>	Key to access an integer field that contains a vendor-specified value.
<code>kCGTabletEventVendor3</code>	Key to access an integer field that contains a vendor-specified value.
<code>kCGTabletProximityEventCapabilityMask</code>	Key to access an integer field that contains the device capabilities mask.
<code>kCGTabletProximityEventDeviceID</code>	Key to access an integer field that contains the system-assigned device ID.
<code>kCGTabletProximityEventEnterProximity</code>	Key to access an integer field that indicates whether the pen is in proximity to the tablet. The value is non-zero if the pen is in proximity to the tablet and zero when leaving the tablet.
<code>kCGTabletProximityEventPointerID</code>	Key to access an integer field that contains the vendor-defined ID of the pointing device.

kCGTabletProximityEventPointerType	Key to access an integer field that contains the pointer type.
kCGTabletProximityEventSystemTabletID	Key to access an integer field that contains the system-assigned unique tablet ID.
kCGTabletProximityEventTabletID	Key to access an integer field that contains the vendor-defined tablet ID, typically the USB product ID.
kCGTabletProximityEventVendorID	Key to access an integer field that contains the vendor-defined ID, typically the USB vendor ID.
kCGTabletProximityEventVendorPointerSerialNumber	Key to access an integer field that contains the vendor-defined pointer serial number.
kCGTabletProximityEventVendorPointerType	Key to access an integer field that contains the vendor-assigned pointer type.
kCGTabletProximityEventVendorUniqueID	Key to access an integer field that contains the vendor-defined unique ID.
kCGTailAppendEventTap	Specifies that a new event tap should be inserted after any pre-existing event taps at the same location.
maxUsecLatency	
minUsecLatency	
processBeingTapped	
tappingProcess	
tapPoint	

## CGFont.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGFontCanCreatePostScriptSubset	Determines whether Quartz can create a subset of the font in PostScript format.
CGFontCopyPostScriptName	Obtains the PostScript name of a font.
CGFontCopyVariationAxes	Returns an array of the variation axis dictionaries for a font.

CGFontCopyVariations	Returns the variation specification dictionary for a font.
CGFontCreateCopyWithVariations	Creates a copy of a font using a variation specification dictionary.
CGFontCreatePostScriptEncoding	Creates a PostScript encoding of a font.
CGFontCreatePostScriptSubset	Creates a subset of the font in the specified PostScript format.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGFontPostScriptFormat	Possible formats for a PostScript font subset.
kCGFontPostScriptFormatType1	This is documented in Adobe Type 1 Font Format, which is available from <a href="http://partners.adobe.com/">http://partners.adobe.com/</a> .
kCGFontPostScriptFormatType3	This is documented in PostScript Language Reference, 3rd edition, which is available from <a href="http://partners.adobe.com/">http://partners.adobe.com/</a> .
kCGFontPostScriptFormatType42	This is documented in Adobe Technical Note 5012, The Type 42 Font Format Specification, which is available from <a href="http://partners.adobe.com/">http://partners.adobe.com/</a> .
kCGFontVariationAxisDefaultValue	The key used to obtain the default variation axis value from a variation axis dictionary. The value obtained with this key is a CFNumberRef that specifies the default value of the variation axis.
kCGFontVariationAxisMaxValue	The key used to obtain the maximum variation axis value from a variation axis dictionary. The value obtained with this key is a CFNumberRef that specifies the maximum value of the variation axis.
kCGFontVariationAxisMinValue	The key used to obtain the minimum variation axis value from a variation axis dictionary. The value obtained with this key is a CFNumberRef that specifies the minimum value of the variation axis.
kCGFontVariationAxisName	The key used to obtain the variation axis name from a variation axis dictionary. The value obtained with this key is a CFStringRef that specifies the name of the variation axis.

### CGGeometry.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGRectIsInfinite</code>	Returns whether a rectangle is infinite.
-------------------------------	--

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGRectInfinite</code>	A rectangle that has infinite extent.
-----------------------------	---------------------------------------

## CGImage.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGImageCreateCopy</code>	Creates a copy of a bitmap image.
<code>CGImageCreateWithImageInRect</code>	Creates a bitmap image using the data contained within a subregion of an existing bitmap image.
<code>CGImageCreateWithMask</code>	Creates a bitmap image from an existing image and an image mask.
<code>CGImageCreateWithMaskingColors</code>	Creates a bitmap image by masking an existing bitmap image with the provided color values.
<code>CGImageGetBitmapInfo</code>	Returns the bitmap information for a bitmap image.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGBitmapInfo</code>	Component information for a bitmap image.
<code>kCGBitmapAlphaInfoMask</code>	The alpha information mask. Use this to extract alpha information that specifies whether a bitmap contains an alpha channel and how the alpha channel is generated.
<code>kCGBitmapByteOrder16Big</code>	16-bit, big endian format.
<code>kCGBitmapByteOrder16Host</code>	16-bit, host endian format.
<code>kCGBitmapByteOrder16Little</code>	16-bit, little endian format.
<code>kCGBitmapByteOrder32Big</code>	32-bit, big endian format.
<code>kCGBitmapByteOrder32Host</code>	32-bit, host endian format.

<code>kCGBitmapByteOrder32Little</code>	32-bit, little endian format.
<code>kCGBitmapByteOrderDefault</code>	The default byte order.
<code>kCGBitmapByteOrderMask</code>	The byte ordering of pixel formats.
<code>kCGBitmapFloatComponents</code>	The components of a bitmap are floating-point values.

## CGLayer.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGContextDrawLayerAtPoint</code>	Draws the contents of a CGLayer object at the specified point.
<code>CGContextDrawLayerInRect</code>	Draws the contents of a CGLayer object into the specified rectangle.
<code>CGLayerCreateWithContext</code>	Creates a CGLayer object that is associated with a graphics context.
<code>CGLayerGetContext</code>	Returns the graphics context associated with a CGLayer object.
<code>CGLayerGetSize</code>	Returns the width and height of a CGLayer object.
<code>CGLayerGetTypeID</code>	Returns the unique type identifier used for CGLayer objects.
<code>CGLayerRelease</code>	Decrements the retain count of a CGLayer object.
<code>CGLayerRetain</code>	Increments the retain count of a CGLayer object.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGLayerRef</code>	An opaque type used for offscreen drawing.
-------------------------	--

## CGPath.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGPathAddEllipseInRect</code>	Adds to a path an ellipse that fits inside a rectangle.
<code>CGPathContainsPoint</code>	Checks whether a point is contained in a graphics path.

## CGPDFContentStream.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFContentStreamCreateWithPage	Creates a content stream object from a PDF page object.
CGPDFContentStreamCreateWithStream	Creates a PDF content stream object from an existing PDF content stream object.
CGPDFContentStreamGetResource	Gets the specified resource from a PDF content stream object.
CGPDFContentStreamGetStreams	Gets the array of PDF content streams contained in a PDF content stream object.
CGPDFContentStreamRelease	Decrements the retain count of a PDF content stream object.
CGPDFContentStreamRetain	Increments the retain count of a PDF content stream object.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFContentStreamRef	An opaque type that provides access to the data that describes the appearance of a PDF page.
-----------------------	--

## CGPDFContext.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFContextAddDestinationAtPoint	Sets a destination to jump to when a point in the current page of a PDF graphics context is clicked.
CGPDFContextBeginPage	Begins a new page in a PDF graphics context.
CGPDFContextEndPage	Ends the current page in the PDF graphics context.
CGPDFContextSetDestinationForRect	Sets a destination to jump to when a rectangle in the current PDF page is clicked.
CGPDFContextSetURLForRect	Sets the URL associated with a rectangle in a PDF graphics context.

## Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kCGPDFContextAllowsCopying</code>	Whether the document allows copying when unlocked with the user password. The value of this key must be a <code>CFBoolean</code> object. The default value of this key is <code>kCFBooleanTrue</code> .
<code>kCGPDFContextAllowsPrinting</code>	Whether the document allows printing when unlocked with the user password. The value of this key must be a <code>CFBoolean</code> value. The default value of this key is <code>kCFBooleanTrue</code> .
<code>kCGPDFContextArtBox</code>	The art box for the document or for a given page. This key is optional. If present, the value of this key must be a <code>CFData</code> object that contains a <code>CGRect</code> (stored by value, not by reference).
<code>kCGPDFContextAuthor</code>	The corresponding value is a string that represents the name of the person who created the document. This key is optional.
<code>kCGPDFContextBleedBox</code>	The bleed box for the document or for a given page. This key is optional. If present, the value of this key must be a <code>CFData</code> object that contains a <code>CGRect</code> (stored by value, not by reference).
<code>kCGPDFContextCreator</code>	The corresponding value is a string that represents the name of the application used to produce the document. This key is optional.
<code>kCGPDFContextCropBox</code>	The crop box for the document or for a given page. This key is optional. If present, the value of this key must be a <code>CFData</code> object that contains a <code>CGRect</code> (stored by value, not by reference).
<code>kCGPDFContextMediaBox</code>	The media box for the document or for a given page. This key is optional. If present, the value of this key must be a <code>CFData</code> object that contains a <code>CGRect</code> (stored by value, not by reference).
<code>kCGPDFContextOutputIntent</code>	The output intent PDF/X.
<code>kCGPDFContextOutputIntents</code>	Output intent dictionaries.
<code>kCGPDFContextOwnerPassword</code>	The owner password of the PDF document.
<code>kCGPDFContextTitle</code>	The corresponding value is a string that represents the title of the document. This key is optional.
<code>kCGPDFContextTrimBox</code>	The trim box for the document or for a given page. This key is optional. If present, the value of this key must be a <code>CFData</code> object that contains a <code>CGRect</code> (stored by value, not by reference).



kCGPDFContextUserPassword	The user password of the PDF document.
kCGPDFXDestinationOutputProfile	An ICC profile stream defining the transformation from the PDF document's source colors to output device colorants.
kCGPDFXInfo	A human-readable text string containing additional information or comments about the intended target device or production condition.
kCGPDFXOutputCondition	A text string identifying the intended output device or production condition in a human- readable form. This key is optional. If present, the value of this key must be a CFString object.
kCGPDFXOutputConditionIdentifier	A string identifying the intended output device or production condition in a human- or machine-readable form.
kCGPDFXOutputIntentSubtype	The output intent subtype. This key is required. The value of this key must be a CFString object equal to "GTS_PDFX"; otherwise, the dictionary is ignored.
kCGPDFXRegistryName	A string identifying the registry in which the condition designated by kCGPDFXOutputConditionIdentifier is defined.

## CGPDFDocument.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFDocumentGetID	Gets the file identifier for a PDF document.
CGPDFDocumentGetInfo	Gets the information dictionary for a PDF document.

## CGPDFOperatorTable.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFOperatorTableCreate	Creates an empty PDF operator table.
CGPDFOperatorTableRelease	Decrements the retain count of a CGPDFOperatorTable object.
CGPDFOperatorTableRetain	Increments the retain count of a CGPDFOperatorTable object.

CGPDFOperatorTableSetCallback	Sets a callback function for a PDF operator.
-------------------------------	--

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFOperatorCallback	Performs custom processing for PDF operators.
CGPDFOperatorTableRef	An opaque type that stores callback functions for PDF operators.

## CGPDFScanner.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFScannerCreate	Creates a CGPDFScanner object.
CGPDFScannerGetContentStream	Returns the content stream associated with a CGPDFScanner object.
CGPDFScannerPopArray	Retrieves an array object from the scanner stack.
CGPDFScannerPopBoolean	Retrieves a Boolean object from the scanner stack.
CGPDFScannerPopDictionary	Retrieves a PDF dictionary object from the scanner stack.
CGPDFScannerPopInteger	Retrieves an integer object from the scanner stack.
CGPDFScannerPopName	Retrieves a character string from the scanner stack.
CGPDFScannerPopNumber	Retrieves a real value object from the scanner stack.
CGPDFScannerPopObject	Retrieves an object from the scanner stack.
CGPDFScannerPopStream	Retrieves a PDF stream object from the scanner stack.
CGPDFScannerPopString	Retrieves a string object from the scanner stack.
CGPDFScannerRelease	Decrements the retain count of a scanner object.
CGPDFScannerRetain	Increments the retain count of a scanner object.
CGPDFScannerScan	Parses the content stream of a CGPDFScanner object.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFScannerRef	An opaque type used to parse a PDF content stream.
-----------------	--

## CGPDFStream.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFDataFormatJPEG2000	The data stream is encoded in JPEG-2000 format.
-------------------------	---

## CGPDFString.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFStringCopyDate	Converts a string to a date.
---------------------	------------------------------

## CGRemoteOperation.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGEventSupressionState	
kCGEventSupressionStateRemoteMouseDrag	
kCGEventSupressionStateSupressionInterval	
kCGNumberOfEventSupressionStates	
kCGScreenUpdateOperationReducedDirtyRectangleCount	When presented as part of the requested operations to the function <code>CGWaitForScreenUpdateRects</code> , specifies that the function should try to minimize the number of rectangles returned to represent the changed areas of the display.

## CGSession.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kCGNotifyGUIConsoleSessionChanged	
kCGNotifyUISessionUserChanged	

## AE

---

### AEDataModel.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

typeUTF16ExternalRepresentation	Unicode text in 16-bit external representation with byte-order-mark (BOM).
---------------------------------	--

## AERegistry.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

keyAEQuitWithoutUI	
keyAESearchText	Identifies an optional parameter to the open documents Apple event, described in “Event ID Constants.” The parameter contains the search text from the Spotlight search that identified the documents to be opened.
kTSMHiliteBlockFillText	Specifies block fill highlight style.
kTSMHiliteCaretPosition	Specifies caret position.
kTSMHiliteConvertedText	Specifies range of converted text.
kTSMHiliteNoHilite	Specifies range of non-highlighted text.
kTSMHiliteOutlineText	Specifies outline highlight style.
kTSMHiliteRawText	Specifies range of raw text.

<code>kTSMHiliteSelectedConvertedText</code>	Specifies range of selected converted text.
<code>kTSMHiliteSelectedRawText</code>	Specifies range of selected raw text.
<code>kTSMHiliteSelectedText</code>	Specifies selected highlight style.

## AppleEvents.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kAEAutsaveNow</code>	
<code>kAEOpenContents</code>	Event that provides an application with dragged content, such as text or an image.

## ImageIO

---

### CGImageDestination.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGImageDestinationAddImage</code>	Adds an image to an image destination.
<code>CGImageDestinationAddImageFromSource</code>	Adds an image from an image source to an image destination.
<code>CGImageDestinationCopyTypeIdentifiers</code>	Returns an array of the uniform type identifiers (UTIs) that are supported for image destinations.
<code>CGImageDestinationCreateWithData</code>	Creates an image destination that writes to a Core Foundation mutable data object.
<code>CGImageDestinationCreateWithDataConsumer</code>	Creates an image destination that writes to the specified data consumer.
<code>CGImageDestinationCreateWithURL</code>	Creates an image destination that writes to a location specified by a URL.
<code>CGImageDestinationFinalize</code>	Writes image data and properties to the data, URL, or data consumer associated with the image destination.

<code>CGImageDestinationGetTypeID</code>	Returns the unique type identifier of an image destination opaque type.
<code>CGImageDestinationSetProperties</code>	Applies one or more properties to all images in an image destination.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGImageDestinationRef</code>	An opaque type that represents an image destination.
<code>kCGImageDestinationBackgroundColor</code>	The desired background color to composite against when writing an image that has an alpha component to a destination format that does not support alpha.
<code>kCGImageDestinationLossyCompressionQuality</code>	The desired compression quality to use when writing to an image destination.

### CGImageProperties.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kCGImageProperty8BIMDictionary</code>	A dictionary of key-value pairs for an Adobe Photoshop image. See “8BIM Dictionary Keys.”
<code>kCGImagePropertyCIFFDictionary</code>	A dictionary of key-value pairs for an image that uses Camera Image File Format (CIFF). See “CIFF Dictionary Keys.”
<code>kCGImagePropertyColorModel</code>	The color model of the image such as, "RGB", "CMYK", "Gray", or "Lab". The value of this key is <code>CFStringRef</code> .
<code>kCGImagePropertyColorModelCMYK</code>	A CMYK color model.
<code>kCGImagePropertyColorModelGray</code>	A Gray color model.
<code>kCGImagePropertyColorModelLab</code>	A Lab color model.
<code>kCGImagePropertyColorModelRGB</code>	An RGB color model.

kCGImagePropertyDepth	The number of bits in each color sample of each pixel. If present, this key is a CFNumber value.
kCGImagePropertyDPIHeight	The resolution, in dots per inch, in the x dimension. If present, this key is a CFNumber value.
kCGImagePropertyDPIWidth	The resolution, in dots per inch, in the y dimension. If present, this key is a CFNumber value.
kCGImagePropertyExifApertureValue	The aperture value.
kCGImagePropertyExifBrightnessValue	The brightness value.
kCGImagePropertyExifCFAPattern	The color filter array (CFA) pattern, which is the geometric pattern of the image sensor for a 1-chip color sensor area.
kCGImagePropertyExifColorSpace	The color space.
kCGImagePropertyExifComponentsConfiguration	The components configuration. For compressed data, specifies that the channels of each component are arranged in increasing numeric order (from first component to the fourth).
kCGImagePropertyExifCompressedBitsPerPixel	The compressed bits per pixel.
kCGImagePropertyExifContrast	The contrast applied to the image.
kCGImagePropertyExifCustomRendered	Special rendering performed on the image data.
kCGImagePropertyExifDateTimeDigitized	The digitized date and time.
kCGImagePropertyExifDateTimeOriginal	The original date and time.
kCGImagePropertyExifDeviceSettingDescription	For a particular camera mode, indicates the conditions for taking the picture.
kCGImagePropertyExifDictionary	A dictionary of key-value pairs for an image that uses Exchangeable Image File Format (EXIF). See “EXIF Dictionary Keys.”
kCGImagePropertyExifDigitalZoomRatio	The digital zoom ratio.
kCGImagePropertyExifExposureBiasValue	The exposure bias value.
kCGImagePropertyExifExposureIndex	The selected exposure index.
kCGImagePropertyExifExposureMode	The exposure mode setting.

kCGImagePropertyExifExposureProgram	The exposure program.
kCGImagePropertyExifExposureTime	The exposure time.
kCGImagePropertyExifFileSource	The image source.
kCGImagePropertyExifFlash	The flash status when the image was shot.
kCGImagePropertyExifFlashEnergy	The strobe energy when the image was captures, in beam candle power seconds.
kCGImagePropertyExifFlashPixVersion	The FlashPix version supported by an FPXR file. FlashPix is a format for multi-resolution, tiled images, that facilitates fast onscreen viewing.
kCGImagePropertyExifFNumber	The F number.
kCGImagePropertyExifFocalLength	The focal length.
kCGImagePropertyExifFocalLenIn35mmFilm	The equivalent focal length in 35 mm film.
kCGImagePropertyExifFocalPlaneResolutionUnit	The unit of measurement for the focal plane x and y tags.
kCGImagePropertyExifFocalPlaneXResolution	The number of image-width pixels (x) per focal plane resolution unit.
kCGImagePropertyExifFocalPlaneYResolution	The number of image-height pixels (y)per focal plane resolution unit.
kCGImagePropertyExifGainControl	The gain adjustment applied to the image.
kCGImagePropertyExifGamma	The gamma setting.
kCGImagePropertyExifImageUniqueID	The unique ID of the image.
kCGImagePropertyExifISOSpeedRatings	ISO speed ratings.
kCGImagePropertyExifLightSource	The light source.
kCGImagePropertyExifMakerNote	A maker note.
kCGImagePropertyExifMaxApertureValue	The maximum aperture value.
kCGImagePropertyExifMeteringMode	The metering mode.
kCGImagePropertyExifOECF	The opto-electrical conversion function (OECF), which defines the relationship between the optical input of the camera and the image values.



kCGImagePropertyExifPixelXDimension	The pixel x dimension.
kCGImagePropertyExifPixelYDimension	The pixel y dimension.
kCGImagePropertyExifRelatedSoundFile	A related sound file.
kCGImagePropertyExifSaturation	The saturation applied to the image.
kCGImagePropertyExifSceneCaptureType	The scene capture type (standard, landscape, portrait, night).
kCGImagePropertyExifSceneType	The scene type.
kCGImagePropertyExifSensingMethod	The sensor type of the camera or input device.
kCGImagePropertyExifSharpness	The sharpness applied to the image.
kCGImagePropertyExifShutterSpeedValue	The shutter speed value.
kCGImagePropertyExifSpatialFrequencyResponse	The spatial frequency table and spatial frequency response values in the direction of image width, image height, and diagonal directions. See ISO 12233..
kCGImagePropertyExifSpectralSensitivity	The spectral sensitivity of each channel.
kCGImagePropertyExifSubjectArea	The subject area.
kCGImagePropertyExifSubjectDistance	The distance to the subject, in meters.
kCGImagePropertyExifSubjectDistRange	The subject distance range.
kCGImagePropertyExifSubjectLocation	The location of the scene's primary subject.
kCGImagePropertyExifSubsecTime	The fraction of seconds for the date and time tag.
kCGImagePropertyExifSubsecTimeDigitized	The fraction of seconds for the digitized time tag.
kCGImagePropertyExifSubsecTimeOriginal	The fraction of seconds for the original date and time tag.
kCGImagePropertyExifUserComment	A user comment.
kCGImagePropertyExifVersion	The version.
kCGImagePropertyExifWhiteBalance	The white balance mode.
kCGImagePropertyFileSize	The size of the image file in bytes, if known. If present, this key is a CFNumber value.
kCGImagePropertyGIFDelayTime	The delay time.

kCGImagePropertyGIFDictionary	A dictionary of key-value pairs for an image that uses Graphics Interchange Format (GIF). See “GIF Dictionary Keys.”
kCGImagePropertyGIFHasGlobalColorMap	Whether or not the GIF has a global color map.
kCGImagePropertyGIFImageColorMap	The image color map.
kCGImagePropertyGIFLoopCount	The loop count.
kCGImagePropertyGPSAltitude	The altitude.
kCGImagePropertyGPSAltitudeRef	The reference altitude.
kCGImagePropertyGPSAreaInformation	The name of the GPS area.
kCGImagePropertyGPSDateStamp	The data and time information relative to Coordinated Universal Time (UTC).
kCGImagePropertyGPSDestBearing	The bearing to the destination point.
kCGImagePropertyGPSDestBearingRef	The reference for giving the bearing to the destination point.
kCGImagePropertyGPSDestDistance	The distance to the destination point.
kCGImagePropertyGPSDestDistanceRef	The units for expressing the distance to the destination point.
kCGImagePropertyGPSDestLatitude	The latitude of the destination point.
kCGImagePropertyGPSDestLatitudeRef	Whether the latitude of the destination point is northern or southern.
kCGImagePropertyGPSDestLongitude	The longitude of the destination point.
kCGImagePropertyGPSDestLongitudeRef	Whether the longitude of the destination point is east or west.
kCGImagePropertyGPSDictionary	A dictionary of key-value pairs for an image that has Global Positioning System (GPS) information. See “GPS Dictionary Keys.”
kCGImagePropertyGPSDifferential	Whether differential correction is applied to the GPS receiver.
kCGImagePropertyGPSDOP	The data degree of precision (DOP).
kCGImagePropertyGPSImgDirection	The direction of the image.
kCGImagePropertyGPSImgDirectionRef	The reference for the direction of the image.

kCGImagePropertyGPSLatitude	The latitude.
kCGImagePropertyGPSLatitudeRef	Whether the latitude is northern or southern.
kCGImagePropertyGPSLongitude	The longitude.
kCGImagePropertyGPSLongitudeRef	Whether the longitude is east or west.
kCGImagePropertyGPSMapDatum	The geodetic survey data used by the GPS receiver.
kCGImagePropertyGPSMeasureMode	The measurement mode.
kCGImagePropertyGPSProcessingMethod	The name of the method used for finding a location.
kCGImagePropertyGPSSatellites	The satellites used for GPS measurements.
kCGImagePropertyGPSSpeed	The GPS receiver speed of movement.
kCGImagePropertyGPSSpeedRef	The unit for expressing the GPS receiver speed of movement.
kCGImagePropertyGPSStatus	The status of the GPS receiver.
kCGImagePropertyGPSTimeStamp	The time as UTC (Coordinated Universal Time).
kCGImagePropertyGPSTrack	The direction of GPS receiver movement.
kCGImagePropertyGPSTrackRef	The reference for the direction of GPS receiver movement.
kCGImagePropertyGPSVersion	The version.
kCGImagePropertyHasAlpha	Whether or not the image has an alpha channel. The value of this key is <code>kCFBooleanTrue</code> if the image contains an alpha channel.
kCGImagePropertyIPTCActionAdvised	The advised action.
kCGImagePropertyIPTCByline	The byline.
kCGImagePropertyIPTCBylineTitle	The byline title.
kCGImagePropertyIPTCCaptionAbstract	The caption abstract.
kCGImagePropertyIPTCCategory	The category.
kCGImagePropertyIPTCCity	The city.
kCGImagePropertyIPTCContact	Contact information.

kCGImagePropertyIPTCContentLocationCode	The content location code.
kCGImagePropertyIPTCContentLocationName	The content location name.
kCGImagePropertyIPTCCopyrightNotice	The copyright notice.
kCGImagePropertyIPTCCountryPrimaryLocationCode	The country primary location code.
kCGImagePropertyIPTCCountryPrimaryLocationName	The country primary location name.
kCGImagePropertyIPTCCredit	Credit information.
kCGImagePropertyIPTCDateCreated	The date created.
kCGImagePropertyIPTCDictionary	A dictionary of key-value pairs for an image that uses International Press Telecommunications Council (IPTC) metadata. See “IPTC Dictionary Keys.”
kCGImagePropertyIPTCDigitalCreationDate	The digital creation date.
kCGImagePropertyIPTCDigitalCreationTime	The digital creation time.
kCGImagePropertyIPTCEditorialUpdate	An editorial update.
kCGImagePropertyIPTCEditStatus	The edit status.
kCGImagePropertyIPTCExpirationDate	The expiration date.
kCGImagePropertyIPTCExpirationTime	The expiration time.
kCGImagePropertyIPTCFixtureIdentifier	A fixture identifier.
kCGImagePropertyIPTCHeadline	The headline.
kCGImagePropertyIPTCImageOrientation	The image orientation.
kCGImagePropertyIPTCImageType	The image type.
kCGImagePropertyIPTCKeywords	Keywords.
kCGImagePropertyIPTCLanguageIdentifier	The language identifier.
kCGImagePropertyIPTCObjectAttributeReference	The object attribute.
kCGImagePropertyIPTCObjectCycle	The object cycle.
kCGImagePropertyIPTCObjectName	The object name.
kCGImagePropertyIPTCObjectTypeReference	The object type.
kCGImagePropertyIPTCOriginalTransmissionReference	The original transmission reference.
kCGImagePropertyIPTCOriginatingProgram	The originating program.
kCGImagePropertyIPTCProgramVersion	The program version.

kCGImagePropertyIPTCProvinceState	The province or state.
kCGImagePropertyIPTCReferenceDate	The reference date.
kCGImagePropertyIPTCReferenceNumber	The reference number.
kCGImagePropertyIPTCReferenceService	The reference service.
kCGImagePropertyIPTCReleaseDate	The release date.
kCGImagePropertyIPTCReleaseTime	The release time.
kCGImagePropertyIPTCSource	The source.
kCGImagePropertyIPTCSpecialInstructions	Special instructions.
kCGImagePropertyIPTCStarRating	The star rating.
kCGImagePropertyIPTCSubjectReference	The subject.
kCGImagePropertyIPTCSubLocation	The sublocation.
kCGImagePropertyIPTCSupplementalCategory	A supplemental category.
kCGImagePropertyIPTCTimeCreated	The time created.
kCGImagePropertyIPTCUrgency	The urgency level.
kCGImagePropertyIPTCWriterEditor	The writer or editor.
kCGImagePropertyIsFloat	Whether or not the image contains floating-point pixel samples. The value of this key is kCFBooleanTrue if the image contains them.
kCGImagePropertyIsIndexed	Whether or not the image contains indexed pixel samples (sometimes called paletted samples). The value of this key is kCFBooleanTrue if the image contains them.
kCGImagePropertyJFIFDensityUnit	The density unit.
kCGImagePropertyJFIFDictionary	A dictionary of key-value pairs for an image that uses JPEG File Interchange Format (JFIF). See “JFIF Dictionary Keys.”
kCGImagePropertyJFIFIsProgressive	Whether or not the image is progressive.
kCGImagePropertyJFIFVersion	The version.
kCGImagePropertyJFIFXDensity	The x density.
kCGImagePropertyJFIFYDensity	The y density.

kCGImagePropertyOrientation	The intended display orientation of the image.
kCGImagePropertyPixelHeight	The number of pixels in the y dimension. If present, this key is a CFNumber value.
kCGImagePropertyPixelWidth	The number of pixels in the x dimension. If present, this key is a CFNumber value.
kCGImagePropertyPNGChromaticities	The chromaticities.
kCGImagePropertyPNGDictionary	A dictionary of key-value pairs for an image that uses Portable Network Graphics (PNG) format. See “PNG Dictionary Keys.”
kCGImagePropertyPNGGamma	The gamma value.
kCGImagePropertyPNGInterlaceType	The interlace type.
kCGImagePropertyPNGsRGBIntent	The sRGB intent.
kCGImagePropertyPNGXPixelsPerMeter	The number of x pixels per meter.
kCGImagePropertyPNGYPixelsPerMeter	The number of y pixels per meter.
kCGImagePropertyProfileName	The name of the optional ICC profile embedded in the image, if known. If present, the value of this key is a CFStringRef.
kCGImagePropertyRawDictionary	A dictionary of key-value pairs for an image that contains minimally processed, or raw, data.
kCGImagePropertyTIFFArtist	The artist.
kCGImagePropertyTIFFCompression	The compression scheme used on the image data.
kCGImagePropertyTIFFCopyright	Copyright information.
kCGImagePropertyTIFFDateTime	The date and time.
kCGImagePropertyTIFFDictionary	A dictionary of key-value pairs for an image that uses Tagged Image File Format (TIFF). See “TIFF Dictionary Keys.”
kCGImagePropertyTIFFDocumentName	The document name.
kCGImagePropertyTIFFHostComputer	The computer or operation system used when the image was created.

<code>kCGImagePropertyTIFFImageDescription</code>	The image description.
<code>kCGImagePropertyTIFFMake</code>	The camera or input device make.
<code>kCGImagePropertyTIFFModel</code>	A camera or input device model.
<code>kCGImagePropertyTIFFOrientation</code>	The image orientation.
<code>kCGImagePropertyTIFFPhotometricInterpretation</code>	The color space of the image data.
<code>kCGImagePropertyTIFFPrimaryChromaticities</code>	The chromaticities of the primaries of the image.
<code>kCGImagePropertyTIFFResolutionUnit</code>	The units of resolution.
<code>kCGImagePropertyTIFFSoftware</code>	The name and version of the software used for image creation.
<code>kCGImagePropertyTIFFTransferFunction</code>	The transfer function, in tabular format, used to map pixel components from a nonlinear form into a linear form.
<code>kCGImagePropertyTIFFWhitePoint</code>	The white point.
<code>kCGImagePropertyTIFFXResolution</code>	The number of pixels per resolution unit in the image width direction.
<code>kCGImagePropertyTIFFYResolution</code>	The number of pixels per resolution unit in the image height direction.

## CGImageSource.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGImageSourceCopyProperties</code>	Returns the properties of the image source.
<code>CGImageSourceCopyPropertiesAtIndex</code>	Returns the properties of the image at a specified location in an image source.
<code>CGImageSourceCopyTypeIdentifiers</code>	Returns an array of uniform type identifiers (UTIs) that are supported for image sources.
<code>CGImageSourceCreateImageAtIndex</code>	Creates a <code>CGImage</code> object for the image data associated with the specified index in an image source.
<code>CGImageSourceCreateIncremental</code>	Create an incremental image source.
<code>CGImageSourceCreateThumbnailAtIndex</code>	Creates a thumbnail image of the image located at a specified location in an image source.

<code>CGImageSourceCreateWithData</code>	Creates an image source that reads from a Core Foundation data object.
<code>CGImageSourceCreateWithDataProvider</code>	Creates an image source that reads data from the specified data provider.
<code>CGImageSourceCreateWithURL</code>	Creates an image source that reads from a location specified by a URL.
<code>CGImageSourceGetCount</code>	Returns the number of images (not including thumbnails) in the image source.
<code>CGImageSourceGetStatus</code>	Return the status of an image source.
<code>CGImageSourceGetStatusAtIndex</code>	Returns the current status of an image that is at a specified location in an image source.
<code>CGImageSourceGetType</code>	Returns the uniform type identifier of the source container.
<code>CGImageSourceGetTypeID</code>	Returns the unique type identifier of an image source opaque type.
<code>CGImageSourceUpdateData</code>	Updates an incremental image source with new data.
<code>CGImageSourceUpdateDataProvider</code>	Updates an incremental image source with a new data provider.

### Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGImageSourceRef</code>	An opaque type that represents an image source.
<code>CGImageSourceStatus</code>	Status states for images and image sources.
<code>kCGImageSourceCreateThumbnailFromImageAlways</code>	Whether a thumbnail should be created from the full image even if a thumbnail is present in the image source file.
<code>kCGImageSourceCreateThumbnailFromImageIfAbsent</code>	Whether a thumbnail should be automatically created for an image if a thumbnail isn't present in the image source file.
<code>kCGImageSourceCreateThumbnailWithTransform</code>	Whether the thumbnail should be rotated and scaled according to the orientation and pixel aspect ratio of the full image.
<code>kCGImageSourceShouldAllowFloat</code>	Whether the image should be returned as a <code>CGImage</code> object that uses floating-point values, if supported by the file format.



<code>kCGImageSourceShouldCache</code>	Whether the image should be cached in a decoded form.
<code>kCGImageSourceThumbnailMaxPixelSize</code>	The maximum width and height in pixels of a thumbnail.
<code>kCGImageSourceTypeIdentifierHint</code>	The best guess of the uniform type identifier (UTI) for the format of the image source file.
<code>kCGImageStatusComplete</code>	The operation is complete.
<code>kCGImageStatusIncomplete</code>	The operation is not complete
<code>kCGImageStatusInvalidData</code>	The data is not valid.
<code>kCGImageStatusReadingHeader</code>	In the process of reading the header.
<code>kCGImageStatusUnexpectedEOF</code>	The end of the file was encountered unexpectedly.
<code>kCGImageStatusUnknownType</code>	The image is an unknown type.

## ColorSync

---

### CMApplication.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CMProfileCopyICCData</code>	
-----------------------------------	--

### CMDeviceIntegration.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kCMDisplayDeviceProfilesNotification</code>	
---	--

## CMFloatBitmap.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CMConvertRGBFloatBitmap	
CMConvertXYZFloatBitmap	
CMFloatBitmapMakeChunky	
CMMatchFloatBitmap	

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

buffers	
CMFloatBitmap	
CMFloatBitmapFlags	
colStride	
kCMFloatBitmapFlagsAlpha	
kCMFloatBitmapFlagsAlphaPremul	
kCMFloatBitmapFlagsNone	
kCMFloatBitmapFlagsRangeClipped	
kCMIlluminantD50	
kCMIlluminantD65	
rowStride	

## CMICCPProfile.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

cmBlackPointCompensation	
--------------------------	--

cmBlackPointCompensationMask	
------------------------------	--

## CMMComponent.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CMMCheckBitmap	
CMMCheckColors	
CMMClose	
CMMConcatInit	
CMMGetPS2ColorRendering	
CMMGetPS2ColorRenderingIntent	
CMMGetPS2ColorRenderingVMSize	
CMMGetPS2ColorSpace	
CMMMatchBitmap	
CMMMatchColors	
CMMMatchFloatBitmap	
CMMNewLinkProfile	
CMMOpen	
CMMSetProperties	
CMMValidateProfile	
NCMMConcatInit	
NCMMInit	
NCMMNewLinkProfile	



## 10.3 Symbol Changes

---

This article lists the symbols added to `ApplicationServices.framework` in Mac OS X v10.3.

### C Symbols

All of the header files with new symbols are listed alphabetically, with their new symbols described.

#### LaunchServices

---

##### LSInfo.h

---

##### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>LSCopyApplicationURLsForURL</code>	Locates all known applications suitable for opening an item designated by URL.
<code>LSRegisterFSRef</code>	Registers an application, designated by file-system reference, in the Launch Services database.
<code>LSRegisterURL</code>	Registers an application, designated by URL, in the Launch Services database.

##### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kLSAppInTrashErr</code>	The application cannot be run because it is inside a Trash folder.
<code>kLSIncompatibleSystemVersionErr</code>	The application to be launched cannot run on the current Mac OS version.
<code>kLSMultipleSessionsNotSupportedErr</code>	The application to be launched cannot run simultaneously in two different user sessions.

kLSNoClassicEnvironmentErr	The Classic emulation environment was required but is not available.
kLSNoExecutableErr	The executable file is missing or has an unusable format.
kLSNoLaunchPermissionErr	The user does not have permission to launch the application (on a managed network).
kLSNoRegistrationInfoErr	Not currently used.

## UTType.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

UTCreateStringForOSType	
UTGetOSTypeFromString	
UTTypeConformsTo	
UTTypeCopyDeclaration	
UTTypeCopyDeclaringBundleURL	
UTTypeCopyDescription	
UTTypeCopyPreferredTagWithClass	
UTTypeCreateAllIdentifiersForTag	
UTTypeCreatePreferredIdentifierForTag	
UTTypeEqual	

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kUTExportedTypeDeclarationsKey	
kUTImportedTypeDeclarationsKey	
kUTTagClassFilenameExtension	
kUTTagClassMIMETYPE	
kUTTagClassNSPboardType	

kUTTagClassOSType	
kUTTypeConformsToKey	
kUTTypeDescriptionKey	
kUTTypeIconFileKey	
kUTTypeIdentifierKey	
kUTTypeReferenceURLKey	
kUTTypeTagSpecificationKey	
kUTTypeVersionKey	

## HI Services

---

### AXError.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kAXErrorParameterizedAttributeUnsupported	
---	--

### AXUIElement.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

AXUIElementCopyParameterizedAttributeNames	
AXUIElementCopyParameterizedAttributeValue	

### AXValue.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

AXValueGetTypeID	
------------------	--

## Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

AXValueRef	
------------	--

## Icons.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

GetIconRefFromIconFamilyPtr	Provides an IconRef object from a specified icon family.
GetIconRefFromTypeInfo	Provides an IconRef object with the specified type information.
IsDataAvailableInIconRef	Indicates whether an IconRef has the specified data.

## Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kComputerIcon	
kIconServicesNoBadgeFlag	
kIconServicesUpdateIfNeededFlag	

## Pasteboard.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

PasteboardClear	Clears the contents of the specified pasteboard.
PasteboardCopyItemFlavorData	Obtains data from a pasteboard for the desired flavor.
PasteboardCopyItemFlavors	Obtains an array of flavors for a specified item in a pasteboard.
PasteboardCopyPasteLocation	Determines the location in which to paste promised data.
PasteboardCreate	Creates a reference to the specified global pasteboard.



PasteboardGetItemCount	Obtains the number of data items in the specified pasteboard.
PasteboardGetItemFlavorFlags	Obtains the flags for a given flavor.
PasteboardGetItemIdentifier	Obtains the item identifier for an item in a pasteboard.
PasteboardGetTypeID	
PasteboardPutItemFlavor	Adds flavor data or a promise to the specified pasteboard.
PasteboardResolvePromises	Resolves all promises to a given pasteboard.
PasteboardSetPasteLocation	Sets the paste location before requesting flavor data.
PasteboardSetPromiseKeeper	Registers the promise keeper callback function for a pasteboard.
PasteboardSynchronize	Synchronizes the local pasteboard reference to reflect the contents of the global pasteboard.

### Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

badPasteboardFlavorErr	The item flavor does not exist.
badPasteboardIndexErr	The specified pasteboard item index does not exist.
badPasteboardItemErr	The item reference does not exist.
badPasteboardSyncErr	The pasteboard has been modified and must be synchronized before use.
context	
duplicatePasteboardFlavorErr	The item flavor already exists.
kPasteboardClientIsOwner	The application recently cleared the pasteboard and is its current owner.
kPasteboardFlavorNoFlags	No flag information exists for this flavor.
kPasteboardFlavorNotSaved	The receiver should not save the data provided for this flavor.
kPasteboardFlavorPromised	The data associated with this flavor is not yet on the pasteboard.
kPasteboardFlavorRequestOnly	When the sender sets this flag, this flavor is hidden from calls to PasteboardCopyItemFlavors.
kPasteboardFlavorSenderOnly	Only the process that added this flavor can see it.

<code>kPasteboardFlavorSenderTranslated</code>	The sender translated this data in some fashion before adding it to the pasteboard.
<code>kPasteboardFlavorSystemTranslated</code>	This data flavor is available through the Translation Manager.
<code>kPasteboardModified</code>	The pasteboard was modified since the last time the application accessed it, and the local pasteboard has been synchronized to reflect any changes.
<code>noPasteboardPromiseKeeperErr</code>	The application attempted to add promised data without previously registering a promise keeper callback.
<code>notPasteboardOwnerErr</code>	The application did not clear the pasteboard before attempting to add flavor data.
<code>PasteboardFlavorFlags</code>	Indicate useful information associated with pasteboard item flavors.
<code>PasteboardItemID</code>	Defines a pasteboard item identifier.
<code>PasteboardRef</code>	Defines an opaque type that represents a pasteboard.
<code>PasteboardSyncFlags</code>	Indicate the pasteboard status after a call to <code>PasteboardSynchronize</code> .

## Processes.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>KillProcess</code>	Terminates a process with the specified ID.
<code>TransformProcessType</code>	Changes the type of the specified process.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kProcessTransformToForegroundApplication</code>	Use to convert a background-only application to a foreground application.
<code>ProcessApplicationTransformState</code>	Specify transformation types to be applied when calling <code>TransformProcessType</code> .

## TranslationServices.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

TranslationCopyDestinationType	
TranslationCopySourceType	
TranslationCreate	
TranslationCreateWithSourceArray	
TranslationGetTranslationFlags	
TranslationGetTypeID	
TranslationPerformForData	
TranslationPerformForFile	
TranslationPerformForURL	

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

badTranslationRefErr	
kTranslationDataTranslation	
kTranslationFileTranslation	
TranslationFlags	
TranslationRef	

## QD

---

## ATSUnicodeFlattening.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ATSFlatDataTextLayoutHeader	
flattenedTextLayouts	
numFlattenedTextLayouts	

## ATSUnicodeTypes.h

### Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ATSStyleLineCountType	Specifies how many lines to draw for a given style type.
kATSUDecimalTab	Specifies that the decimal point of a value should be centered on the tab stop.
kATSUFromFollowingLayout	Used for bidirectional cursor movement between paragraphs in the functions <code>ATSURightwardCursorPostion</code> and <code>ATSULeftwardCursorPosition</code> .
kATSUFromPreviousLayout	Used for bidirectional cursor movement between paragraphs in the functions <code>ATSURightwardCursorPosition</code> and <code>ATSULeftwardCursorPosition</code> .
kATSULineDecimalTabCharacterTag	Specifies the current setting for the decimal separator, and affects the behavior of decimal tabs for a text layout (not an individual line).
kATSULineHighlightCGColorTag	Specifies the current setting of the highlight color and opacity. The associated value is of type <code>CGColorRef</code> . This can be set as a line or layout control. The <code>CGColor</code> object ( <code>CGColorRef</code> ) is retained by the text layout object in which it is set.
kATSUStyleDoubleLineCount	Specifies to use a double line.
kATSUStyleDropShadowBlurOptionTag	Specifies the amount of blur for a drop shadow. The associated value is of type <code>float</code> . The default value is 0.0. May be set as a style attribute.
kATSUStyleDropShadowColorOptionTag	Specifies the color and opacity of a drop shadow. The associated value is of type <code>CGColorRef</code> . The default value is <code>NULL</code> . You need to set the <code>CGColorRef</code> to a value other than <code>NULL</code> if you want to see the drop shadow. May be set as a style attribute.

<code>kATSUStyleDropShadowOffsetOptionTag</code>	Specifies the amount of offset from the text to be used when drawing a drop shadow. The associated value is of type <code>CGSize</code> . The default value is (3.0, -3.0). May be set as a style attribute.
<code>kATSUStyleDropShadowTag</code>	Specifies the text should be drawn with a drop shadow.
<code>kATSUStyleSingleLineCount</code>	Specifies to use a single line.
<code>kATSUStyleStrikeThroughColorOptionTag</code>	Specifies the color of the strokes to draw for a strikethrough style.
<code>kATSUStyleStrikeThroughCountOptionTag</code>	Specifies the number of strokes to be drawn for a strikethrough. The associated value is of type <code>ATSUStyleLineCountType</code> . The default value is <code>kATSUStyleSingleLineCount</code> . May be set as a style attribute.
<code>kATSUStyleStrikeThroughTag</code>	Specifies strikethrough style. The associated value is of type <code>Boolean</code> . The default value is false. May be set as a style attribute.
<code>kATSUStyleUnderlineColorOptionTag</code>	Specifies the color of the strokes to draw for an underlined run of text.
<code>kATSUStyleUnderlineCountOptionTag</code>	Specifies the number of strokes to be drawn for an underline. The associated value is of type <code>ATSUStyleLineCountType</code> . The default value is <code>kATSUStyleSingleLineCount</code> . May be set as a style attribute.

## QDOffscreen.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kNativeEndianPixMap</code>	By default, the function <code>NewGWorld</code> allocates pixel buffers with big-endian byte ordering regardless of the system architecture.
<code>nativeEndianPixMapBit</code>	

## Quickdraw.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

QDGetCGDirectDisplayID	Returns the Quartz display ID that corresponds to a QuickDraw graphics device.
QDGetCursorData	
QDGetCursorNameForSystemCursor	
QDGetCursorScale	
QDGetPictureBounds	
QDUnregisterNamedPixMapCursor	

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kQDXAliasCursor	
kQDXArrowCursor	
kQDXCopyCursor	
kQDXIBeamCursor	
kQDXIBeamXORCursor	
kQDXMoveCursor	
kQDXNumberOfSystemCursors	
QDXSystemCursorID	

## ATS

---

### ATSTFont.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kATSInvalidGlyphAccess	
------------------------	--

## ATSLayoutTypes.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kATSLineBreakToNearestCharacter	Specifies that line breaking should occur at the nearest character, not word. This could cause a word to be split over multiple lines.
---------------------------------	--

## PrintCore

---

### PMCore.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

PMCopyAvailablePPDs	Obtains the list of PostScript printer description (PPD) files in a PPD domain.
PMCopyLocalizedPPD	Obtains a localized PostScript printer description (PPD) file.
PMCopyPPDData	Obtains the uncompressed PPD data for a PostScript printer description (PPD) file.
PMCreatePageFormatWithPMPaper	Creates a page format object with a specified paper.
PMGetPageFormatPaper	Obtains the paper associated with a page format object.
PMPaperCreate	Creates a paper object.
PMPaperGetHeight	Obtains the height of the sheet of paper represented by a paper object.
PMPaperGetID	Obtains the identifier of a paper object.
PMPaperGetMargins	Obtains the margins describing the unprintable area of the sheet represented by a paper object.
PMPaperGetName	Obtains the name for a given paper.
PMPaperGetWidth	Obtains the width of the sheet of paper represented by a paper object.
PMPresetCopyName	Obtains the localized name for a preset.
PMPresetCreatePrintSettings	Creates a print settings object with settings that correspond to a preset.

PMPresetGetAttributes	Obtains the attributes of a preset.
PMPrinterCopyHostName	Obtains the name of the server hosting the print queue for a given printer.
PMPrinterCopyPresets	Obtains a list of print settings presets for a printer.
PMPrinterGetCommInfo	Obtains information about the communication channel for a printer.
PMPrinterGetMimeTypes	Obtains a list of MIME content types supported by a printer using the specified print settings.
PMPrinterGetPaperList	Obtains the list of papers available for a printer.
PMPrinterIsRemote	Indicates whether a printer is hosted by a remote print server.
PMPrinterPrintWithFile	Submits a print job to a specified printer using a file that contains print data.
PMPrinterPrintWithProvider	Submits a print job to a specified printer using a Quartz data provider to obtain the print data.
PMPrintSettingsToOptions	Converts print settings into a CUPS options string.
PMSessionSetCurrentPMPrinter	Changes the current printer for a printing session.
PMWorkflowCopyItems	Obtains an array of the available PDF workflow items.
PMWorkflowSubmitPDFWithOptions	Submits a PDF file for workflow processing using the specified CUPS options string.
PMWorkflowSubmitPDFWithSettings	Submits a PDF file for workflow processing using the specified print settings.

## PMDefinitions.h

### Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kAllPPDDomains	Specifies all available domains.
kCUPSPPDDomain	Specifies the CUPS domain.
kLocalPPDDomain	Specifies the local domain.
kNetworkPPDDomain	Specifies the network domain.
kPMCMYKColorSpaceModel	
kPMDevNColorSpaceModel	



kPMGrayColorSpaceModel	
kPMInvalidPreset	Your application passed an invalid preset object.
kPMRGBColorSpaceModel	
kPMUnknownColorSpaceModel	
kSystemPPDDomain	Specifies the system domain.
kUserPPDDomain	Specifies the user domain.
PMColorSpaceModel	
PMPaper	An opaque type that stores information about the paper used in a print job.
PMPaperMargins	A data structure that specifies the unprintable area of a paper object.
PMPPDDomain	Constants that specify the domains for PostScript printer description (PPD) files.
PMPPreset	An opaque type that stores information about a named preset available for a print job.

## PMErrors.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kPMInvalidCalibrationTarget	The dictionary specifying a printer calibration target is invalid.
kPMJobGetTicketBadFormatError	
kPMJobGetTicketReadError	

## PMTicket.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

PMXMLURLToTicket	
------------------	--

## Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kPMCoverPageAfter	
kPMCoverPageBefore	
kPMCoverPageNone	

## PPDLib.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ppdCreateInstallableOptionsData	
---------------------------------	--

## CoreGraphics

---

## CGColor.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGColorCreate	Creates a Quartz color using a list of intensity values (including alpha) and an associated color space.
CGColorCreateCopy	Creates a copy of an existing Quartz color.
CGColorCreateCopyWithAlpha	Creates a copy of an existing Quartz color, substituting a new alpha value.
CGColorCreateWithPattern	Creates a Quartz color using a list of intensity values (including alpha), a pattern color space, and a pattern.
CGColorEqualToColor	Indicates whether two colors are equal.
CGColorGetAlpha	Returns the value of the alpha component associated with a Quartz color.
CGColorGetColorSpace	Returns the color space associated with a Quartz color.
CGColorGetComponents	Returns the values of the color components (including alpha) associated with a Quartz color.

<code>CGColorGetNumberOfComponents</code>	Returns the number of color components (including alpha) associated with a Quartz color.
<code>CGColorGetPattern</code>	Returns the pattern associated with a Quartz color in a pattern color space.
<code>CGColorGetTypeID</code>	Returns the Core Foundation type identifier for a Quartz color data type.
<code>CGColorRelease</code>	Decrements the retain count of a Quartz color.
<code>CGColorRetain</code>	Increments the retain count of a Quartz color.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGColorRef</code>	An opaque type that represents a color used in Quartz 2D drawing.
-------------------------	---

### CGContext.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGContextBeginTransparencyLayer</code>	Begins a transparency layer.
<code>CGContextDrawPDFPage</code>	Draws a page in the current user space of a PDF context.
<code>CGContextEndTransparencyLayer</code>	Ends a transparency layer.
<code>CGContextGetClipBoundingBox</code>	Returns the bounding box of a clipping path.
<code>CGContextSetFillColorWithColor</code>	Sets the current fill color in a graphics context, using a Quartz color.
<code>CGContextSetShadow</code>	Enables shadowing in a graphics context.
<code>CGContextSetShadowWithColor</code>	Enables shadowing with color a graphics context.
<code>CGContextSetStrokeColorWithColor</code>	Sets the current stroke color in a context, using a Quartz color.
<code>CGContextShowGlyphsWithAdvances</code>	Draws an array of glyphs with varying offsets.

## CGDataProvider.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGDataProviderGetBytePointerCallback	A callback function that returns a generic pointer to the provider data.
CGDataProviderGetBytesAtOffsetCallback	A callback function that copies data from the provider into a Quartz buffer.
CGDataProviderGetBytesCallback	A callback function that copies from a provider data stream into a Quartz-supplied buffer.
CGDataProviderReleaseBytePointerCallback	A callback function that releases the pointer Quartz obtained by calling CGDataProviderGetBytePointerCallback.
CGDataProviderReleaseInfoCallback	A callback function that releases any private data or resources associated with the data provider.
CGDataProviderRewindCallback	A callback function that moves the current position in the data stream back to the beginning.
CGDataProviderSkipBytesCallback	A callback function that advances the current position in the data stream supplied by the provider.

## CGDirectDisplay.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGCaptureAllDisplaysWithOptions	Captures all attached displays, using the specified options.
CGDisplayCaptureWithOptions	Captures a display for exclusive use by an application, using the specified options.
CGDisplayGammaTableCapacity	Returns the capacity, or number of entries, in the gamma table for a display.
CGDisplayGetDrawingContext	Returns a graphics context suitable for drawing to a captured display.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGCaptureOptions</code>	Specify configuration parameters when capturing displays.
<code>kCGCaptureNoFill</code>	Disables fill with black.
<code>kCGCaptureNoOptions</code>	Specifies that the system should use the default fill behavior, which is fill with black.

## CGDisplayConfiguration.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGDisplayRegisterReconfigurationCallback</code>	Registers a callback function to be invoked whenever a local display is reconfigured.
<code>CGDisplayRemoveReconfigurationCallback</code>	Removes the registration of a callback function that's invoked whenever a local display is reconfigured.
<code>CGDisplayScreenSize</code>	Returns the width and height of a display in millimeters.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGDisplayChangeSummaryFlags</code>	Specify the configuration parameters passed to a display reconfiguration callback function.
<code>CGDisplayReconfigurationCallback</code>	A client-supplied callback function that's invoked whenever the configuration of a local display is changed.
<code>kCGDisplayAddFlag</code>	The display has been added to the active display list.
<code>kCGDisplayBeginConfigurationFlag</code>	The display configuration is about to change.
<code>kCGDisplayDisabledFlag</code>	The display has been disabled.
<code>kCGDisplayEnabledFlag</code>	The display has been enabled.
<code>kCGDisplayMirrorFlag</code>	The display is now mirroring another display.
<code>kCGDisplayMovedFlag</code>	The location of the upper-left corner of the display in global display space has changed.
<code>kCGDisplayRemoveFlag</code>	The display has been removed from the active display list.
<code>kCGDisplaySetMainFlag</code>	The display is now the main display.

<code>kCGDisplaySetModeFlag</code>	The display mode has changed.
<code>kCGDisplayUnMirrorFlag</code>	The display is no longer mirroring another display.

## CGError.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kCGErrorApplicationAlreadyRunning</code>	
<code>kCGErrorApplicationCanOnlyBeRunInOneSessionAtATime</code>	
<code>kCGErrorApplicationIncorrectExecutableFormatFound</code>	
<code>kCGErrorApplicationIsLaunching</code>	

## CGGLContext.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGGLContextCreate</code>	Creates a Quartz graphics context from an OpenGL context.
<code>CGGLContextUpdateViewportSize</code>	Updates the size of the viewport associated with an OpenGL context.

## CGImage.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGImageCreateCopyWithColorSpace</code>	Create a copy of a bitmap image, replacing its colorspace.
--	--

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kCGImageAlphaOnly</code>	There is no color data, only an alpha channel.
--------------------------------	--

## CGPDFArray.h

## Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFArrayGetArray	Returns whether an object at a given index in a PDF array is another PDF array and, if so, retrieves that array.
CGPDFArrayGetBoolean	Returns whether an object at a given index in a PDF array is a PDF Boolean and, if so, retrieves that Boolean.
CGPDFArrayGetCount	Returns the number of items in a PDF array.
CGPDFArrayGetDictionary	Returns whether an object at a given index in a PDF array is a PDF dictionary and, if so, retrieves that dictionary.
CGPDFArrayGetInteger	Returns whether an object at a given index in a PDF array is a PDF integer and, if so, retrieves that object.
CGPDFArrayGetName	Returns whether an object at a given index in a PDF array is a PDF name reference (represented as a constant C string) and, if so, retrieves that name.
CGPDFArrayGetNull	Returns whether an object at a given index in a Quartz PDF array is a PDF null.
CGPDFArrayGetNumber	Returns whether an object at a given index in a PDF array is a PDF number and, if so, retrieves that object.
CGPDFArrayGetObject	Returns whether an object at a given index in a PDF array is a PDF object and, if so, retrieves that object.
CGPDFArrayGetStream	Returns whether an object at a given index in a PDF array is a PDF stream and, if so, retrieves that stream.
CGPDFArrayGetString	Returns whether an object at a given index in a PDF array is a PDF string and, if so, retrieves that string.

## Data Types &amp; Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFArrayRef	An opaque type that encapsulates a PDF array.
---------------	---

## CGPDFDictionary.h

## Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFDictionaryApplyFunction	Applies a function to each entry in a dictionary.
CGPDFDictionaryGetArray	Returns whether there is a PDF array associated with a specified key in a PDF dictionary and, if so, retrieves that array.
CGPDFDictionaryGetBoolean	Returns whether there is a PDF Boolean value associated with a specified key in a PDF dictionary and, if so, retrieves the Boolean value.
CGPDFDictionaryGetCount	Returns the number of entries in a PDF dictionary.
CGPDFDictionaryGetDictionary	Returns whether there is another PDF dictionary associated with a specified key in a PDF dictionary and, if so, retrieves that dictionary.
CGPDFDictionaryGetInteger	Returns whether there is a PDF integer associated with a specified key in a PDF dictionary and, if so, retrieves that integer.
CGPDFDictionaryGetName	Returns whether an object with a specified key in a PDF dictionary is a PDF name reference (represented as a constant C string) and, if so, retrieves that name.
CGPDFDictionaryGetNumber	Returns whether there is a PDF number associated with a specified key in a PDF dictionary and, if so, retrieves that number.
CGPDFDictionaryGetObject	Returns whether there is a PDF object associated with a specified key in a PDF dictionary and, if so, retrieves that object.
CGPDFDictionaryGetStream	Returns whether there is a PDF stream associated with a specified key in a PDF dictionary and, if so, retrieves that stream.
CGPDFDictionaryGetString	Returns whether there is a PDF string associated with a specified key in a PDF dictionary and, if so, retrieves that string.

## Data Types &amp; Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFDictionaryApplierFunction	Performs custom processing on a key-value pair from a PDF dictionary, using optional contextual information.
CGPDFDictionaryRef	An opaque type that encapsulates a PDF dictionary.



## CGPDFDocument.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFDocumentGetCatalog	Returns the document catalog of a Quartz PDF document.
CGPDFDocumentGetPage	Returns a page from a Quartz PDF document.
CGPDFDocumentGetVersion	Returns the major and minor version numbers of a Quartz PDF document.

## CGPDFObject.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFObjectGetType	Returns the PDF type identifier of an object.
CGPDFObjectGetValue	Returns whether an object is of a given type and if it is, retrieves its value.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFBoolean	A PDF Boolean value.
CGPDFInteger	A PDF integer value.
CGPDFObjectRef	An opaque type that contains information about a PDF object.
CGPDFObjectType	Types of PDF object.
CGPDFReal	A PDF real value.
kCGPDFObjectTypeArray	Type for a PDF array.
kCGPDFObjectTypeBoolean	The type for a PDF Boolean.
kCGPDFObjectTypeDictionary	The type for a PDF dictionary.
kCGPDFObjectTypeInteger	The type for a PDF integer.
kCGPDFObjectTypeName	Type for a PDF name.

kCGPDFObjectTypeNull	The type for a PDF null.
kCGPDFObjectTypeReal	The type for a PDF real.
kCGPDFObjectTypeStream	The type for a PDF stream.
kCGPDFObjectTypeString	The type for a PDF string.

## CGPDFPage.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFPageGetBoxRect	Returns the rectangle that represents a type of box for a content region or page dimensions of a PDF page.
CGPDFPageGetDictionary	Returns the dictionary of a PDF page.
CGPDFPageGetDocument	Returns the document for a page.
CGPDFPageGetDrawingTransform	Returns the affine transform that maps a box to a given rectangle on a PDF page.
CGPDFPageGetPageNumber	Returns the page number of the specified PDF page.
CGPDFPageGetRotationAngle	Returns the rotation angle of a PDF page.
CGPDFPageGetTypeID	Returns the CType ID for PDF page objects.
CGPDFPageRelease	Decrements the retain count of a PDF page.
CGPDFPageRetain	Increments the retain count of a PDF page.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFBox	Box types for a PDF page.
CGPDFPageRef	An opaque type that represents a page in a PDF document.
kCGPDFArtBox	The page art box—a rectangle, expressed in default user space units, defining the extent of the page's meaningful content (including potential white space) as intended by the page's creator.
kCGPDFBleedBox	The page bleed box—a rectangle, expressed in default user space units, that defines the region to which the contents of the page should be clipped when output in a production environment

kCGPDFCropBox	The page crop box—a rectangle, expressed in default user space units, that defines the visible region of default user space. When the page is displayed or printed, its contents are to be clipped to this rectangle.
kCGPDFMediaBox	The page media box—a rectangle, expressed in default user space units, that defines the boundaries of the physical medium on which the page is intended to be displayed or printed
kCGPDFTrimBox	The page trim box—a rectangle, expressed in default user space units, that defines the intended dimensions of the finished page after trimming.

## CGPDFStream.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFStreamCopyData	Returns the data associated with a PDF stream.
CGPDFStreamGetDictionary	Returns the dictionary associated with a PDF stream.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFDataFormat	The encoding format of PDF data.
CGPDFDataFormatJPEGEncoded	The data stream is encoded in JPEG format.
CGPDFDataFormatRaw	The data stream is not encoded.
CGPDFStreamRef	An opaque type that represents a PDF stream.

## CGPDFString.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFStringCopyTextString	Returns a CFString object that represents a PDF string as a text string.
CGPDFStringGetBytePtr	Returns a pointer to the bytes of a PDF string.
CGPDFStringGetLength	Returns the number of bytes in a PDF string.

## Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFStringRef	An opaque data type that represents a string in a PDF document.
----------------	---

## CGPSConverter.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPSConverterAbort	Tells a PostScript converter to abort a conversion at the next available opportunity.
CGPSConverterConvert	Uses a PostScript converter to convert PostScript data to PDF data.
CGPSConverterCreate	Creates a new PostScript converter.
CGPSConverterGetTypeID	Returns the Core Foundation type identifier for PostScript converters.
CGPSConverterIsConverting	Checks whether the converter is currently converting data.

## Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

beginDocument	
beginPage	
CGPSConverterBeginDocumentCallback	Performs custom tasks at the beginning of a PostScript conversion process.
CGPSConverterBeginPageCallback	Performs custom tasks at the beginning of each page in a PostScript conversion process.
CGPSConverterCallbacks	A structure for holding the callbacks provided when you create a PostScript converter object.
CGPSConverterEndDocumentCallback	Performs custom tasks at the end of a PostScript conversion process.
CGPSConverterEndPageCallback	Performs custom tasks at the end of each page of a PostScript conversion process.

CGPSConverterMessageCallback	Passes messages generated during a PostScript conversion process.
CGPSConverterProgressCallback	Reports progress periodically during a PostScript conversion process.
CGPSConverterRef	An opaque data type used to convert PostScript data to PDF data.
CGPSConverterReleaseInfoCallback	Performs custom tasks when a PostScript converter is released.
endDocument	
endPage	
noteMessage	
noteProgress	

## CGRemoteOperation.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGCursorIsDrawnInFramebuffer	Returns a Boolean value indicating whether the mouse cursor is drawn in frame buffer memory.
CGCursorIsVisible	Returns a Boolean value indicating whether the mouse cursor is visible.
CGScreenRegisterMoveCallback	Registers a callback function to be invoked when an area of the display is moved.
CGScreenUnregisterMoveCallback	Removes a previously registered callback function invoked when an area of the display is moved.
CGSetLocalEventsFilterDuringSuppressionState	Filters local hardware events from the keyboard and mouse during the short interval after a synthetic event is posted.
CGWaitForScreenUpdateRects	Waits for screen update operations.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGEventSuppressionState</code>	Specify the event suppression states that can occur after posting an event.
<code>CGScreenUpdateMoveCallback</code>	A client-supplied callback function that's invoked when an area of the display is moved.
<code>CGScreenUpdateMoveDelta</code>	Represents the distance a region on the screen moves in pixel units.
<code>CGScreenUpdateOperation</code>	Specify types of screen update operations.
<code>dX</code>	
<code>dY</code>	
<code>kCGEventSuppressionStateRemoteMouseDrag</code>	Specifies that certain local hardware events may be suppressed during a mouse drag operation (mouse movement with the left or only mouse button down).
<code>kCGEventSuppressionStateSuppressionInterval</code>	Specifies that certain local hardware events may be suppressed for a short interval after posting an event.
<code>kCGNumberOfEventSuppressionStates</code>	
<code>kCGScreenUpdateOperationMove</code>	Specifies a screen move operation.
<code>kCGScreenUpdateOperationRefresh</code>	Specifies a screen refresh operation.

## CGSession.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGSessionCopyCurrentDictionary</code>	Returns information about the caller's window server session.
---	---

## CGWindowLevel.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kCGAssistiveTechHighWindowLevelKey</code>	
---	--

## AE

---

### AEDataModel.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

keyReplyRequestedAttr	A Boolean value indicating whether the Apple event expects to be replied to.
typeApplicationBundleID	Indicates a descriptor containing UTF-8 characters that specify the bundle ID of an application. Bundle IDs should be constructed similarly to "com.company.directorylocation.ApplicationName".

### AERegistry.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

typeGIF	
typeJPEG	

### AppleEvents.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

AECreatRemoteProcessResolver	Creates an object for resolving a list of remote processes.
AEDisposeRemoteProcessResolver	Disposes of an AERemoteProcessResolverRef.
AERemoteProcessResolverGetProcesses	Returns an array of objects containing information about processes running on a remote machine.
AERemoteProcessResolverScheduleWithRunLoop	Schedules a resolver for execution on a given run loop in a given mode.

**Data Types & Constants**

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>AERemoteProcessResolverContext</code>	Supplied as a parameter when performing asynchronous resolution of remote processes.
<code>AERemoteProcessResolverRef</code>	An opaque reference to an object that encapsulates the mechanism for obtaining a list of processes running on a remote machine.
<code>kAERemoteProcessNameKey</code>	Use this key to obtain the visible name of the remote process, in the localization supplied by the server, as a <code>CFStringRef</code> .
<code>kAERemoteProcessProcessIDKey</code>	Use this key to obtain the process ID of the remote process, if available; if so, returned as a <code>CFNumberRef</code> .
<code>kAERemoteProcessURLKey</code>	Use this key to obtain the full URL to the remote process, as a <code>CFURLRef</code> .
<code>kAERemoteProcessUserIDKey</code>	Use this key to obtain the user ID of the remote process, if available; if so, returned as a <code>CFNumberRef</code> .

## SpeechSynthesis

---

### SpeechSynthesis.h

---

**Data Types & Constants**

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>soOutputToFileWithCFURL</code>	Pass a <code>CFURLRef</code> to write to this file, <code>NULL</code> to generate sound.
--------------------------------------	--

## ColorSync

---

### CMApplication.h

---

**Functions**

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CMCopyProfileDescriptionString</code>	Returns the name of a profile as a <code>CFString</code> .
---	--



<code>CMMakeProfile</code>	Makes a display or abstract profile by modifying an existing one.
<code>CWFillLookupTexture</code>	Fills a 3-D lookup texture from a color world.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>a</code>	
<code>b</code>	
<code>cmTextureRGBtoRGBX8</code>	



## 10.2 Symbol Changes

---

This article lists the symbols added to `ApplicationServices.framework` in Mac OS X v10.2.

### C Symbols

All of the header files with new symbols are listed alphabetically, with their new symbols described.

#### LaunchServices

---

##### LaunchServices.h

---

###### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>LSCopyApplicationForMIMETYPE</code>	Locates the preferred application for opening items with a specified MIME type.
<code>LSCopyKindStringForMIMETYPE</code>	Obtains the kind string for a specified MIME type.
<code>LSCopyKindStringForTypeInfo</code>	Obtains a kind string for items with a specified file type, creator signature, filename extension, or any combination of these characteristics.

#### AE

---

##### AEDataModel.h

---

###### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>AECreatDescFromExternalPtr</code>	Creates a new descriptor that uses a memory buffer supplied by the caller.
---	--

AEGetDescDataRange	Retrieves a specified series of bytes from the specified descriptor.
DisposeAEDisposeExternalUPP	Disposes of a universal procedure pointer to a function that disposes of data supplied to the AECreatDescFromExternalPtr function.
InvokeAEDisposeExternalUPP	Calls a dispose external universal procedure pointer.
NewAEDisposeExternalUPP	Creates a new universal procedure pointer to a function that disposes of data stored in a buffer.

### Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kAEHTTPProxyHostAttr	A value of type typeChar or typeUTF8Text.
kAEHTTPProxyPortAttr	A value of type typeSInt32.
kAESocks4Protocol	
kAESocks5Protocol	
kAESocksHostAttr	
kAESocksPasswordAttr	
kAESocksPortAttr	
kAESocksProxyAttr	
kAESocksUserAttr	
kAEUseHTTPProxyAttr	A value of type typeBoolean. Specifies whether to manually specify the proxy host and port. Defaults to true.
kAEUseSocksAttr	
keyRPCMethodParamOrder	

### AEMach.h

### Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

keyReplyPortAttr	
------------------	--

## AERegistry.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kAELogOut	
kAEReallyLogOut	
kAEShowRestartDialog	
kAEShowShutdownDialog	
keyAETSMGlyphInfoArray	
typeGlyphInfoArray	
typeUTF8Text	8-bit Unicode (UTF-8 encoding).

## FindByContent

---

### FindByContent.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

DisposeFBCHitTestUPP	Disposes of universal procedure pointer (UPP) to a search-hit testing callback.
FBCBlindExampleSearchWithCallback	Performs a similarity search that uses example files instead of a query string, and installs callbacks for progress reporting and search hit testing.
FBCHandleSummary	Disposes of a summary reference object.
FBCHandleSummaryOfCFString	Creates a summary reference object for CFString.
FBCHandleSummarySentenceCount	Obtains the number of sentences in a summary reference object.
FBCHandleSummarySentences	Obtains a summary that contains a specified number of sentences.
FBCHandleSessionCallback	Sets a callback function that allows your application to cancel a search operation.

<code>FBSetSessionHitTest</code>	Sets a callback function that allows your application to perform search-hit testing.
<code>FBSummarizeCFString</code>	Summarizes text that is specified as a <code>CFString</code> .
<code>InvokeFBCHitTestUPP</code>	Invokes a universal procedure pointer (UPP) to a search-hit testing callback.
<code>NewFBCHitTestUPP</code>	Creates a new universal procedure pointer (UPP) to a search-hit testing callback.

### Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>FBSummaryRef</code>	A pointer to an opaque data type (referred to as a summary reference object) that contains summary information, from which summary text can be obtained.
<code>kFBCarabicHighWord</code>	
<code>kFBCcenteuroHighWord</code>	
<code>kFBCcroatianHighWord</code>	
<code>kFBCcyrillicHighWord</code>	
<code>kFBCdanishHighWord</code>	
<code>kFBCdefaultLanguagesHighWord</code>	
<code>kFBCdevanagariHighWord</code>	
<code>kFBCdutchHighWord</code>	
<code>kFBCenglishHighWord</code>	
<code>kFBCfrenchHighWord</code>	
<code>kFBCgermanHighWord</code>	
<code>kFBCgreekHighWord</code>	
<code>kFBCgujuratiHighWord</code>	
<code>kFBCgurmukhiHighWord</code>	
<code>kFBChebrewHighWord</code>	
<code>kFBCicelandicHighWord</code>	

kFBCitalianHighWord	
kFBCjapaneseHighWord	
kFBCkoreanHighWord	
kFBCnotAllFoldersSearchable	Not all folders are searchable
kFBCportugueseHighWord	
kFBCromanHighWord	
kFBCromanianHighWord	
kFBCspanishHighWord	
kFBCsummarizationFailed	Summarization operation failed
kFBCswedishHighWord	
kFBCturkishHighWord	

## HI Services

---

### AXError.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

AXError	
kAXErrorActionUnsupported	
kAXErrorAPIDisabled	
kAXErrorAttributeUnsupported	
kAXErrorCannotComplete	
kAXErrorFailure	
kAXErrorIllegalArgument	
kAXErrorInvalidUIElement	
kAXErrorInvalidUIElementObserver	
kAXErrorNotificationAlreadyRegistered	
kAXErrorNotificationNotRegistered	

kAXErrorNotificationUnsupported	
kAXErrorNotImplemented	
kAXErrorNoValue	
kAXErrorSuccess	

## AXUIElement.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

AXAPIEnabled	
AXObserverAddNotification	
AXObserverCreate	
AXObserverGetRunLoopSource	
AXObserverGetTypeID	
AXObserverRemoveNotification	
AXUIElementCopyActionDescription	
AXUIElementCopyActionNames	
AXUIElementCopyAttributeNames	
AXUIElementCopyAttributeValue	
AXUIElementCopyAttributeValues	
AXUIElementCopyElementAtPosition	
AXUIElementCreateApplication	
AXUIElementCreateSystemWide	
AXUIElementGetAttributeValueCount	
AXUIElementGetPid	
AXUIElementGetTypeID	
AXUIElementIsAttributeSettable	
AXUIElementPerformAction	
AXUIElementPostKeyboardEvent	



AXUIElementSetAttributeValue	
------------------------------	--

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

AXObserverCallback	
--------------------	--

AXObserverRef	
---------------	--

AXUIElementRef	
----------------	--

## AXValue.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

AXValueCreate	
---------------	--

AXValueGetType	
----------------	--

AXValueGetValue	
-----------------	--

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

AXValueType	
-------------	--

kAXValueCRangeType	
--------------------	--

kAXValueCGPointType	
---------------------	--

kAXValueCGRectType	
--------------------	--

kAXValueCGSizeType	
--------------------	--

kAXValueIllegalType	
---------------------	--

## Icons.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kUnknownFSObjectIcon	
----------------------	--

## Processes.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ProcessInformationCopyDictionary	Obtains a superset of GetProcessInformation in modern data types.
SetFrontProcessWithOptions	Brings a process to the front of the process list, and activates it.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kProcessDictionaryIncludeAllInformationMask	
kSetFrontProcessFrontWindowOnly	Activate the process, but bring only the frontmost non-floating window forward.

## QD

---

## ATSUnicode.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ATSUBatchBreakLines	Calculates soft line breaks for the text associated with a text layout object.
ATSUGetTabArray	Retrieves the tab ruler associated with a text layout object.
ATSUGetUnjustifiedBounds	Obtains the typographic bounding rectangle for a line of text prior to final layout.
ATSUHighlightInactiveText	Highlights previously selected text using an alpha value of 0.5.
ATSUOffsetToCursorPosition	Obtains the caret position(s) corresponding to a memory offset, after a move of the specified length.

ATSUPositionToCursorOffset	Obtains the memory offset for the glyph edge nearest a mouse-down event, after a move of the specified length.
ATSUSetTabArray	Sets a tab ruler for a text layout object.

### Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ATSUGlyphSelector	Contains information that directs ATSUI to use a specific glyph instead of the one ATSUI normally derives.
ATSURGBAlphaColor	Contains color information that includes alpha channel information.
ATSUTab	Contains tab settings.
ATSUTabType	Specify text positioning for ATSUI tab stops.
GlyphCollection	Specify a character set.
kATSUAscentTag	Specifies the ascent value of a style's font. The associated value is of type <code>ATSUTextMeasurement</code> and has a default value of the ascent value of the style object's font with the current point size.
kATSUCenterTab	Specifies that the tabbed text should be centered on the tab stop.
kATSUDescentTag	Specifies the descent value of a style's font.
kATSUFontMatrixTag	Specifies a font transformation matrix.
kATSUGlyphSelectorTag	Specifies a glyph collection.
kATSULayoutOperationOverrideTag	Specifies to override a layout operation. The associated value is of type <code>ATSULayoutOperationOverrideSpecifier</code> and has a default value of <code>NULL</code> .
kATSULEadingTag	Specifies the leading value of a style's font. The associated value is of type <code>ATSUTextMeasurement</code> and has a default value of the leading value of the style object's font with the current point size.
kATSULeftTab	Specifies that the left side of the tabbed text should be flush against the tab stop.
kATSUNumberTabTypes	Specifies the number of valid tab types.
kATSURGBAlphaColorTag	Specifies RGB color with an alpha channel. The associated value is of type <code>ATSURGBAlphaColor</code> and has a default value of (0,0,0,1).

kATSURightTab	Specifies that the right side of the tabbed text should be flush against the tab stop.
kGlyphCollectionAdobeCNS1	Specifies Adobe CNS1 CID-keyed fonts.
kGlyphCollectionAdobeGB1	Specifies Adobe GB1 CID-keyed fonts.
kGlyphCollectionAdobeJapan1	Specifies Adobe Japan1 CID-keyed fonts.
kGlyphCollectionAdobeJapan2	Specifies Adobe Japan2 CID-keyed fonts.
kGlyphCollectionAdobeKorea1	Specifies Adobe Korea1 CID-keyed fonts.
kGlyphCollectionGID	Indicates that the glyph value represents the actual glyph ID of a specific font.
kGlyphCollectionUnspecified	Indicates that the glyph collection is not specified.
tabPosition	

## ATSUnicodeDirectAccess.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ATSUDirectAddStyleSettingRef	Looks up, and if necessary, adds a style setting to a line of text.
ATSUDirectGetLayoutDataArrayPtrFromLineRef	Obtains the glyph data specified by a direct-data selector and for a specific line of text.
ATSUDirectGetLayoutDataArrayPtrFromTextLayout	Obtains a copy of the glyph data specified by a direct-data selector and for a specific line of text in a text layout object.
ATSUDirectReleaseLayoutDataArrayPtr	Releases a pointer to a direct-data array.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ATSUDirectDataSelector	Specify the layout data to obtain when calling the functions ATSUDirectGetLayoutData- ArrayPtrFromLineRef or ATSUDirectGetLayoutDataArray- PtrFromTextLayout.
------------------------	---

ATSUStyleSettingRef	A reference to an opaque style setting object.
kATSUDirectDataAdvanceDeltaFixedArray	Specifies the parallel advance delta (delta X) array, which is an array of Fixed values.
kATSUDirectDataBaselineDeltaFixedArray	Specifies the parallel baseline delta (delta Y) array, which is an array of Fixed values.
kATSUDirectDataDeviceDeltaSInt16Array	Specifies the parallel device delta array, which is an array of SInt16 values used to adjust truncated fractional values for devices that do not accept fractional positioning.
kATSUDirectDataLayoutRecordATSLayoutRecordCurrent	Specifies the ATSLayoutRecord array, with the current version of the ATSLayoutRecord data structure.
kATSUDirectDataLayoutRecordATSLayoutRecordVersion1	Specifies the ATSLayoutRecord array, with the version 1 of the ATSLayoutRecord data structure.
kATSUDirectDataStyleIndexUInt16Array	Specifies the parallel style index array, which is an array of (UInt16) values.
kATSUDirectDataStyleSettingATSUStyleSettingRefArray	Specifies the style setting reference (ATSUStyleSettingRef) array.

## ATSUnicodeFlattening.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ATSUFlattenStyleRunsToStream	Flattens ATSUI style-run data so that it can be saved to disk or passed (through the pasteboard) to another application.
ATSUUnflattenStyleRunsFromStream	Unflattens previously-flattened ATSUI style run data so that it can be read from disk or accepted (through the pasteboard) from another application.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ATSFlatDataFontNameDataHeader	Contains font name information.
-------------------------------	---------------------------------

ATSFlatDataFontSpeciferType	Specifies the data type for flattened font name data.
ATSFlatDataFontSpecRawNameData	Contains data for a font name.
ATSFlatDataFontSpecRawNameDataHeader	Contains raw font name data.
ATSFlatDataLayoutControlsDataHeader	Contains the number of flattened layout controls and an array of layout control attribute data.
ATSFlatDataLineInfoData	Contains a line length and the number of line controls for a line of flattened text.
ATSFlatDataLineInfoHeader	Contains the number of lines and an array of line information data.
ATSFlatDataMainHeaderBlock	Contains the 'ustl' data structure version and size and provides offsets to the text layout, style run, and style list data blocks.
ATSFlatDataStyleListFeatureData	Contains flattened font feature data.
ATSFlatDataStyleListHeader	Contains the number of styles and the first item in the style list style data header.
ATSFlatDataStyleListStyleDataHeader	Contains size information and the number of attributes, features, and variations for the style list data block.
ATSFlatDataStyleListVariationData	Contains flattened font variation axis data.
ATSFlatDataStyleRunDataHeader	Contains the number of style runs and style run information for the style run data block.
ATSFlatDataTextLayoutDataHeader	Contains size, length, and offset information for a text layout data block.
ATSUFlattenedDataStreamFormat	Specify the format to use when flattening or unflattening data.
ATSUFlattenStyleRunOptions	Specify options to use when flattening ATSUI style run data.
ATSUStyleRunInfo	Contains information for a style run.
ATSUUnFlattenStyleRunOptions	
controlArray	
fontNameLanguage	
fontNameLength	
fontNamePlatform	

fontNameScript	
fontNameType	
kATSFlatDataUstlCurrentVersion	Specifies the current version.
kATSFlatDataUstlVersion0	Specifies version 0. This version is obsolete.
kATSFlatDataUstlVersion1	Specifies version 1. This version is obsolete.
kATSFlatDataUstlVersion2	Specifies version 2.
kATSFlattenedFontSpecifierRawNameData	Specifies to use the font name as the flattened font name.
kATSUDataStreamUnicodeStyledText	Specifies to use the 'ustl' data specification when flattening or unflattening data.
kATSUFlattenOptionNoOptionsMask	Specifies that no options are to be used.
kATSUUnFlattenOptionNoOptionsMask	
lineInfoArray	
lineLength	
nameDataArray	
nameSpecifierSize	
nameSpecifierType	
numberOfFlattenedNames	
numberOfLayoutControls	
numberOfLineControls	
numberOfLines	
numberOfSetAttributes	
numberOfSetFeatures	
numberOfSetVariations	
numberOfStyleRuns	
numberOfStyles	
offsetToLayoutControls	
offsetToLineInfo	
offsetToStyleList	

offsetToStyleRuns	
offsetToTextLayouts	
runLength	
sizeOfDataBlock	
sizeOfLayoutData	
sizeOfStyleInfo	
styledataArray	
styleObjectIndex	
styleRunArray	
textLayoutLength	
theFeatureSelector	
theFeatureType	
theVariationAxis	
theVariationValue	

## Displays.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kDMNotifyDisplayDidWake	This selector is only available in Mac OS X.
kDMNotifyDisplayWillSleep	This selector is only available in Mac OS X.

## Quickdraw.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

QDGlobalToLocalPoint	
QDGlobalToLocalRect	
QDGlobalToLocalRegion	



QDIsNamedPixMapCursorRegistered	
QDLocalToGlobalPoint	
QDLocalToGlobalRect	
QDLocalToGlobalRegion	
QDRegisterNamedPixMapCursor	
QDSetCursorScale	
QDSetNamedPixMapCursor	
QDSwapPortTextFlags	
QDSwapTextFlags	
QDUnregisterNamedPixMapCursor	
SwapPortPolySaveHandle	
SwapPortRegionSaveHandle	

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

hwMirroredDevice	
kQDDontChangeFlags	
kQDSupportedFlags	
kQDUseCGTextMetrics	
kQDUseCGTextRendering	
kQDUseDefaultTextRendering	
kQDUseTrueTypeScalerGlyphs	

### QuickdrawText.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

SwapQDTextFlags	
-----------------	--

## ATS

---

### ATSTFont.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ATSCreateFontQueryRunLoopSource	Sets up your application to handle font queries.
ATSTFontNotificationSubscribe	Signs up your application to receive notification of changes to fonts and font directories.
ATSTFontNotificationUnsubscribe	Unsubscribes your application from receiving notifications of changes to fonts and font directories.
ATSTFontNotify	Notifies Apple Type Services of an action taken by your application.

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ATSTFontNotificationInfoRef	An opaque data type that represents a font notification information structure.
ATSTFontNotificationRef	An opaque data type that represents a font notification structure.
ATSTFontNotifyAction	Specify a notification action.
ATSTFontNotifyOption	Specify when ATS should notify your application of changes in the font database.
ATSTFontQueryMessageID	Specifies a message ID for a font request query.
ATSTFontQuerySourceContext	Contains font query information that is passed back to a font query callback.
kATSTFontNotifyActionDirectoriesChanged	Specifies that your application has made changes to one or more of the font directories.
kATSTFontNotifyActionFontsChanged	Specifies that your application has activated or deactivated fonts.
kATSTFontNotifyOptionDefault	Specifies to use the default behavior of the function ATSTFontNotificationSubscribe.

<code>kATSTFontNotifyOptionReceiveWhileSuspended</code>	Specifies to receive notifications even if the application is in the background.
<code>kATSTOptionFlagsDoNotNotify</code>	Specifies not to send a notification after a font is activated or deactivated globally.
<code>kATSTOptionFlagsProcessSubdirectories</code>	Specifies to process the font directories within a font directory. You can pass this as a parameter to the function <code>ATSTFontActivateFromFileSpecification</code> .
<code>kATSTQueryActivateFontMessage</code>	Specifies to activate a font message. The data associated with this message ID is a flattened <code>CFDictionaryRef</code> . The <code>CFDictionary</code> contains one or more of the keys described in “Font Request Query Keys.”

## ATSTLayoutTypes.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>DisposeATSUDirectLayoutOperationOverrideUPP</code>	Disposes of a universal procedure pointer (UPP) to a layout operation override callback.
<code>InvokeATSUDirectLayoutOperationOverrideUPP</code>	Calls your layout operation override callback.
<code>NewATSUDirectLayoutOperationOverrideUPP</code>	Creates a new universal procedure pointer (UPP) to a layout operation override callback.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>ATSTGlyphInfoFlags</code>	Specify properties for a glyph.
<code>ATSTLayoutRecord</code>	Contains basic layout information for a single glyph.
<code>ATSTLayoutOperationCallbackStatus</code>	Specify the status of a layout operation override callback.
<code>ATSTLayoutOperationOverrideSpecifier</code>	Contains an layout operation selector and a pointer to a layout operation override callback.
<code>ATSTLayoutOperationSelector</code>	Specify a layout operation.
<code>ATSTLineRef</code>	Represents a reference to a structure that specifies a line of text.

iLineRef	
iOperationCallbackParameterPtr	
kATSGlyphInfoAppleReserved	This flag is reserved by Apple. If you try to use it you may get an invalid value error.
kATSGlyphInfoByteSizeMask	Specifies the size of the character that spawned the glyph.
kATSGlyphInfoHasImposedWidth	Specifies that the glyph has an imposed width (that is, an advance width) specified by the style.
kATSGlyphInfoIsAttachment	Specifies that the glyph attaches to another glyph.
kATSGlyphInfoIsLTHanger	Specifies that the glyph can hang off the left or top edge of a line.
kATSGlyphInfoIsRBHanger	Specifies that the glyph can hang off the right or bottom edge of a line.
kATSGlyphInfoIsWhiteSpace	Specifies that the glyph is a whitespace glyph.
kATSGlyphInfoTerminatorGlyph	Specifies that the glyph is not truly a glyph, but an end-marker to allow the calculation of the previous glyph's advance.
kATSLineApplyAntiAliasing	Specifies that Apple Type Services should produce antialiased glyph images even if system preferences or Quartz settings indicate otherwise.
kATSLineDisableAllBaselineAdjustments	Specifies not to perform any baseline-adjustment operations on the line.
kATSLineDisableAllGlyphMorphing	Specifies not to perform any glyph-morphing operations on the line.
kATSLineDisableAllJustification	Specifies not to perform any justification operations on the line.
kATSLineDisableAllKerningAdjustments	Specifies not to perform any kerning-adjustment operations on the line.
kATSLineDisableAllLayoutOperations	Specifies to turn off all layout adjustments for this line.
kATSLineDisableAllTrackingAdjustments	Specifies not to perform any tracking-adjustment operations on the line.

<code>kATSLineDisableAutoAdjustDisplayPos</code>	Specifies not to automatically adjust individual character positions when rendering lines that have any integer glyph positioning, whether the integer glyph positioning is due to non-antialiased characters or though the use of the selector <code>kATSLineFractDisable</code> .
<code>kATSLineDisableNegativeJustification</code>	Specifies to allow glyph positions to extend beyond the line's assigned width if the line width is not sufficient to hold all its glyphs. This ensures that negative justification is not used.
<code>kATSLineNoAntiAliasing</code>	Specifies that Apple Type Services should turn-off antialiasing glyph imaging even if system preferences or Quartz settings indicate otherwise. This option negates the <code>kATSLineApplyAntiAliasing</code> bit if it is set.
<code>kATSLineUseDeviceMetrics</code>	Specifies to used rounded device metrics instead of fractional path metrics.
<code>kATSLineUseQDRendering</code>	Specifies to use QuickDraw to render a line of text instead of the default ATSUI rendering.
<code>kATSSStyleApplyAntiAliasing</code>	Specifies that Apple Type Services should produce antialiased glyph images even if system preferences or Quartz 2D settings indicate otherwise.
<code>kATSSStyleNoAntiAliasing</code>	Specifies that Apple Type Services should turn-off antialiasing glyph imaging even if system preferences or Quartz 2D settings indicate otherwise. This selector negates the <code>kATSSStyleApplyAntiAliasing</code> selector if is set.
<code>kATSSStyleNoHinting</code>	Specifies that Apple Type Services (ATS) should produce unhinted glyph outlines. The default behavior is for ATS to produce is hinted glyph outlines.
<code>kATSULayoutOperationAppleReserved</code>	This selector is reserved for future use.
<code>kATSULayoutOperationBaselineAdjustment</code>	Specifies the baseline-adjustment operation.
<code>kATSULayoutOperationCallbackStatusContinue</code>	Specifies that your callback function has not handled the operation which triggered the callback. This indicates to ATSUI that needs to perform its own processing for the layout operation.

<code>kATSULayoutOperationCallbackStatusHandled</code>	Specifies that your callback function has handled the operation which triggered the callback. This indicates to ATSUI that it does not need to perform any further processing for the layout operation.
<code>kATSULayoutOperationJustification</code>	Specifies the justification operation.
<code>kATSULayoutOperationKerningAdjustment</code>	Specifies the kerning-adjustment operation.
<code>kATSULayoutOperationMorph</code>	Specifies the character-morphing operation.
<code>kATSULayoutOperationNone</code>	Specifies that no layout operation is currently selected.
<code>kATSULayoutOperationPostLayoutAdjustment</code>	Specifies the period of time after ATSUI has completed its layout operations.
<code>kATSULayoutOperationTrackingAdjustment</code>	Specifies the tracking-adjustment operation.
<code>kATSUseGlyphAdvance</code>	Specifies that ATSUI use the natural glyph advance value in a line or entire text layout object.
<code>oCallbackStatus</code>	
<code>operationSelector</code>	
<code>originalOffset</code>	
<code>overrideUPP</code>	
<code>realPos</code>	

## ATSTypes.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kATSDeletedGlyphcode</code>	Indicates that a glyph is deleted. That is, the glyph is set to no longer appear in a text layout.
-----------------------------------	--

## SFNTTypes.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kFontNoLanguageCode	
kFontNoNameCode	
kFontNoPlatformCode	
kFontNoScriptCode	
sfntCMapExtendedSubHeader	
sizeof_sfntCMapExtendedSubHeader	

## PrintCore

---

### PMCore.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

PMCGImageCreateWithEPSDataProvider	Creates an image that references both the PostScript contents of EPS data and a preview (proxy) image for the data.
PMGetCollate	Obtains a Boolean value that indicates whether the job collate option is selected.
PMPrinterGetDeviceURI	Obtains a copy of a printer's device URI.
PMPrinterGetID	Returns the unique identifier of a printer.
PMPrinterGetLocation	Returns the location of a printer.
PMPrinterGetMakeAndModelName	Obtains the manufacturer and model name of the specified printer.
PMPrinterGetName	Returns the human-readable name of a printer.
PMPrinterGetState	Obtains the current state of the print queue for a printer.
PMPrinterIsDefault	Returns a Boolean value indicating whether a printer is the default printer for the current user.
PMPrinterIsFavorite	Returns a Boolean value indicating whether a printer is in the user's list of favorite printers.
PMPrinterIsPostScriptCapable	Returns a Boolean value indicating whether a printer is PostScript capable.
PMServerCreatePrinterList	Creates a list of printers available to a print server.

<code>PMSessionBeginDocumentNoDialog</code>	Begins a print job that, by default, draws into a QuickDraw graphics port, and suppresses the printing status dialog.
<code>PMSessionBeginPageNoDialog</code>	Starts a new page for printing in the specified printing session and suppresses the printing status dialog.
<code>PMSessionEndDocumentNoDialog</code>	Ends a print job started by calling the function <code>PMSessionBeginCGDocumentNoDialog</code> or <code>PMSessionBeginDocumentNoDialog</code> .
<code>PMSessionEndPageNoDialog</code>	Indicates the end of drawing the current page for the specified printing session.
<code>PMSetCollate</code>	Specifies whether the job collate option is selected.

## PMDefinitions.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kPMPrinterIdle</code>	Specifies the idle state.
<code>kPMPrinterProcessing</code>	Specifies the processing state.
<code>kPMPrinterStopped</code>	Specifies the stopped state.
<code>PMPrinterState</code>	Constants that specify the current state of a print queue.
<code>PMServer</code>	An opaque type that identifies a local or remote print server.

## PMErrors.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kPMFeatureNotInstalled</code>	
<code>kPMInvalidPaper</code>	Your application passed an invalid paper object.
<code>kPMReadGotZeroData</code>	A file or connection read operation returned no data.



## PMTemplate.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

PMTemplateWriteXML	
--------------------	--

## PMTicket.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

PMSessionGetTemplateFromSession	
---------------------------------	--

PMSessionGetTicketFromSession	
-------------------------------	--

PMTicketCreateDict	
--------------------	--

PMTicketHasEqualValues	
------------------------	--

## PPDLib.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

PPDAddFeatureEntriesFromPPDContext	
------------------------------------	--

ppdCloseContext	
-----------------	--

ppdGetCompiledPPDData	
-----------------------	--

ppdOpenContext	
----------------	--

## CoreGraphics

---

### CGBitmapContext.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGBitmapContextGetAlphaInfo	Returns the alpha information associated with the context, which indicates how a bitmap context handles the alpha component.
CGBitmapContextGetBitsPerComponent	Returns the bits per component of a bitmap context.
CGBitmapContextGetBitsPerPixel	Returns the bits per pixel of a bitmap context.
CGBitmapContextGetBytesPerRow	Returns the bytes per row of a bitmap context.
CGBitmapContextGetColorSpace	Returns the color space of a bitmap context.
CGBitmapContextGetData	Returns a pointer to the image data associated with a bitmap context.
CGBitmapContextGetHeight	Returns the height in pixels of a bitmap context.
CGBitmapContextGetWidth	Returns the width in pixels of a bitmap context.

### CGColorSpace.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGColorSpaceCreateWithName	Creates a specified type of Quartz color space.
CGColorSpaceGetTypeID	Returns the Core Foundation type identifier for Quartz color spaces.

### CGContext.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGContextAddPath	Adds a previously created Quartz path object to the current path in a graphics context.
CGContextDrawShading	Fills the clipping path of a context with the specified shading.
CGContextGetTypeID	Returns the type identifier for Quartz graphics contexts.
CGContextSetPatternPhase	Sets the pattern phase of a context.
CGContextSetShouldSmoothFonts	Enables or disables font smoothing in a graphics context.

## CGDataConsumer.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGDataConsumerGetTypeID	Returns the Core Foundation type identifier for Quartz data consumers.
-------------------------	--

## CGDataProvider.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGDataProviderGetTypeID	Returns the Core Foundation type identifier for Quartz data providers.
-------------------------	--

## CGDirectDisplay.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGDisplayBestModeForParameters - AndRefreshRateWithProperty	Returns information about the display mode closest to a specified depth, screen size, and refresh rate, with a required property.
CGGetOnlineDisplayList	Provides a list of displays that are online (active, mirrored, or sleeping).
CGMainDisplayID	Returns the display ID of the main display.
CGOpenGLDisplayMaskToDisplayID	Maps an OpenGL display mask to a display ID.

## CGDisplayConfiguration.h

## Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGBeginDisplayConfiguration	Begins a new set of display configuration changes.
CGCancelDisplayConfiguration	Cancels a set of display configuration changes.
CGCompleteDisplayConfiguration	Completes a set of display configuration changes.
CGConfigureDisplayMirrorOfDisplay	Changes the configuration of a mirroring set.
CGConfigureDisplayMode	Configures the display mode of a display.
CGConfigureDisplayOrigin	Configures the origin of a display in global display (desktop) coordinates.
CGDisplayIOServicePort	Returns the I/O Kit service port of the specified display.
CGDisplayIsActive	Returns a Boolean value indicating whether a display is active.
CGDisplayIsAlwaysInMirrorSet	Returns a Boolean value indicating whether a display is always in a mirroring set.
CGDisplayIsAsleep	Returns a Boolean value indicating whether a display is sleeping (and is therefore not drawable.)
CGDisplayIsBuiltin	Returns a Boolean value indicating whether a display is built-in, such as the internal display in portable systems.
CGDisplayIsInHWMirrorSet	Returns a Boolean value indicating whether a display is in a hardware mirroring set.
CGDisplayIsInMirrorSet	Returns a Boolean value indicating whether a display is in a mirroring set.
CGDisplayIsMain	Returns a Boolean value indicating whether a display is the main display.
CGDisplayIsOnline	Returns a Boolean value indicating whether a display is connected or online.
CGDisplayMirrorsDisplay	For a secondary display in a mirroring set, returns the primary display.
CGDisplayModelNumber	Returns the model number of a display monitor.
CGDisplayPrimaryDisplay	Returns the primary display in a hardware mirroring set.

<code>CGDisplaySerialNumber</code>	Returns the serial number of a display monitor.
<code>CGDisplayUnitNumber</code>	Returns the logical unit number of a display.
<code>CGDisplayUsesOpenGLAcceleration</code>	Returns a Boolean value indicating whether Quartz is using OpenGL-based window acceleration (Quartz Extreme) to render in a display.
<code>CGDisplayVendorNumber</code>	Returns the vendor number of the specified display's monitor.
<code>CGRestorePermanentDisplayConfiguration</code>	Restores the permanent display configuration settings for the current user.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGConfigureOption</code>	
<code>CGDisplayConfigRef</code>	Defines a reference to a display configuration transaction.
<code>kCGConfigureForAppOnly</code>	Specifies that changes persist for the lifetime of the current application. After the application terminates, the display configuration settings revert to the current login session.
<code>kCGConfigureForSession</code>	Specifies that changes persist for the lifetime of the current login session. After the current session terminates, the displays revert to the last saved permanent configuration.
<code>kCGConfigurePermanently</code>	Specifies that changes persist in future login sessions by the same user.

### CGDisplayFade.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGAcquireDisplayFadeReservation</code>	Reserves the fade hardware for a specified time interval.
<code>CGConfigureDisplayFadeEffect</code>	Modifies the settings of the built-in fade effect that occurs during a display configuration.
<code>CGDisplayFade</code>	Performs a single fade operation.
<code>CGDisplayFadeOperationInProgress</code>	Returns a Boolean value indicating whether a fade operation is currently in progress.

<code>CGReleaseDisplayFadeReservation</code>	Releases a display fade reservation, and unfades the display if needed.
--	---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGDisplayBlendFraction</code>	Represents the percentage of blend color used in a fade operation.
<code>CGDisplayFadeInterval</code>	Represents the duration in seconds of a fade operation or a fade hardware reservation.
<code>CGDisplayFadeReservationToken</code>	Defines a token issued by Quartz when reserving one or more displays for a fade operation during a specified interval.
<code>CGDisplayReservationInterval</code>	Represents the time interval for a fade reservation.

## CGError.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kCGErrorApplicationNotPermittedToExecute</code>	Macintosh Manager is active, and this application is not permitted to run.
<code>kCGErrorApplicationRequiresNewerSystem</code>	The application being launched requires a newer version of Mac OS X.

## CGFont.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGFontGetTypeID</code>	Returns the Core Foundation type identifier for Quartz fonts.
------------------------------	---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGFontIndex</code>	An index into a font table.
--------------------------	-----------------------------

<code>kCGFontIndexInvalid</code>	An invalid font index (a value which never represents a valid glyph).
<code>kCGFontIndexMax</code>	The maximum allowed value for <code>CGFontIndex</code> .
<code>kCGGlyphMax</code>	The same as <code>kCGFontIndexMax</code> .

## CGFunction.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGFunctionCreate</code>	Creates a Quartz function.
<code>CGFunctionGetTypeID</code>	Returns the type identifier for Quartz function objects.
<code>CGFunctionRelease</code>	Decrements the retain count of a function object.
<code>CGFunctionRetain</code>	Increments the retain count of a function object.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGFunctionCallbacks</code>	A structure that contains callbacks needed by a <code>CGFunction</code> object.
<code>CGFunctionEvaluateCallback</code>	Performs custom operations on the supplied input data to produce output data.
<code>CGFunctionRef</code>	An opaque type that represents a callback function.
<code>CGFunctionReleaseInfoCallback</code>	Performs custom clean-up tasks when Quartz deallocates a <code>CGFunction</code> object.
<code>evaluate</code>	

## CGImage.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGImageCreateWithPNGDataProvider</code>	Creates a Quartz bitmap image using PNG-encoded data supplied by a data provider.
---	---

<code>CGImageGetTypeID</code>	Returns the type identifier for Quartz bitmap images.
-------------------------------	---

## CGPath.h

### Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGPathAddArc</code>	Appends an arc to a mutable graphics path, possibly preceded by a straight line segment.
<code>CGPathAddArcToPoint</code>	Appends an arc to a mutable graphics path, possibly preceded by a straight line segment.
<code>CGPathAddCurveToPoint</code>	Appends a Bézier curve to a mutable graphics path.
<code>CGPathAddLines</code>	Appends an array of new line segments to a mutable graphics path.
<code>CGPathAddLineToPoint</code>	Appends a line segment to a mutable graphics path.
<code>CGPathAddPath</code>	Appends a path to a mutable graphics path.
<code>CGPathAddQuadCurveToPoint</code>	Appends a quadratic curve to a mutable graphics path.
<code>CGPathAddRect</code>	Appends a rectangle to a mutable graphics path.
<code>CGPathAddRects</code>	Appends an array of rectangles to a mutable graphics path.
<code>CGPathApply</code>	For each element in a graphics path, calls a custom applier function.
<code>CGPathCloseSubpath</code>	Closes and completes a subpath in a mutable graphics path.
<code>CGPathCreateCopy</code>	Creates an immutable copy of a graphics path.
<code>CGPathCreateMutable</code>	Creates a mutable graphics path.
<code>CGPathCreateMutableCopy</code>	Creates a mutable copy of an existing graphics path.
<code>CGPathEqualToPath</code>	Indicates whether two graphics paths are equivalent.
<code>CGPathGetBoundingBox</code>	Returns the bounding box of a graphics path.
<code>CGPathGetCurrentPoint</code>	Returns the current point in a graphics path.
<code>CGPathGetTypeID</code>	Returns the Core Foundation type identifier for Quartz graphics paths.
<code>CGPathIsEmpty</code>	Indicates whether or not a graphics path is empty.
<code>CGPathIsRect</code>	Indicates whether or not a graphics path represents a rectangle.



<code>CGPathMoveToPoint</code>	Starts a new subpath at a specified location in a mutable graphics path.
<code>CGPathRelease</code>	Decrements the retain count of a graphics path.
<code>CGPathRetain</code>	Increments the retain count of a graphics path.

### Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGMutablePathRef</code>	An opaque type that represents a mutable graphics path.
<code>CGPathApplierFunction</code>	Defines a callback function that can view an element in a graphics path.
<code>CGPathElement</code>	A data structure that provides information about a path element.
<code>CGPathElementType</code>	The type of element found in a path.
<code>CGPathRef</code>	An opaque type that represents an immutable graphics path.
<code>kCGPathElementAddCurveToPoint</code>	The path element that adds a cubic curve from the current point to the specified point. See the function <code>CGPathAddCurveToPoint</code> .
<code>kCGPathElementAddLineToPoint</code>	The path element that adds a line from the current point to the specified point. See the function <code>CGPathAddLineToPoint</code> .
<code>kCGPathElementAddQuadCurveToPoint</code>	The path element that adds a quadratic curve from the current point to the specified point. See the function <code>CGPathAddQuadCurveToPoint</code> .
<code>kCGPathElementCloseSubpath</code>	The path element that closes and completes a subpath. See the function <code>CGPathCloseSubpath</code> .
<code>kCGPathElementMoveToPoint</code>	The path element that starts a new subpath. See the function <code>CGPathMoveToPoint</code> .
<code>points</code>	

### CGPattern.h

#### Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGPatternGetTypeID</code>	Returns the type identifier for Quartz patterns.
---------------------------------	--

**Data Types & Constants**

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPatternDrawPatternCallback	Draws a pattern cell.
CGPatternReleaseInfoCallback	Release private data or resources associated with the pattern.

**CGPDFDocument.h**

---

**Functions**

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGPDFDocumentAllowsCopying	Returns whether the specified PDF document allows copying.
CGPDFDocumentAllowsPrinting	Returns whether a PDF document allows printing.
CGPDFDocumentGetTypeID	Returns the type identifier for Quartz PDF documents.
CGPDFDocumentIsEncrypted	Returns whether the specified PDF file is encrypted.
CGPDFDocumentIsUnlocked	Returns whether the specified PDF document is currently unlocked.
CGPDFDocumentUnlockWithPassword	Unlocks an encrypted PDF document, if a valid password is supplied.

**CGShading.h**

---

**Functions**

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGShadingCreateAxial	Creates a shading object to use for axial shading.
CGShadingCreateRadial	Creates a shading object to use for radial shading.
CGShadingGetTypeID	Returns the Core Foundation type identifier for Quartz shading objects.
CGShadingRelease	Decrements the retain count of a shading object.
CGShadingRetain	Increments the retain count of a shading object.

**Data Types & Constants**

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGShadingRef	An opaque type that represents a Quartz shading.
--------------	--

**CGWindowLevel.h**

---

**Data Types & Constants**

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kCGCursorWindowLevelKey	The level for the cursor.
-------------------------	---------------------------

**ColorSync**

---

**CMICCProfile.h**

---

**Data Types & Constants**

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

cmParametricType4	$Y = (aX+b)^{\gamma} + e$ [ $X \geq d$ ], $Y = cX+f$ [ $X < d$ ]
-------------------	--



# 10.1 Symbol Changes

This article lists the symbols added to `ApplicationServices.framework` in Mac OS X v10.1.

## C Symbols

All of the header files with new symbols are listed alphabetically, with their new symbols described.

### LaunchServices

#### LaunchServices.h

##### Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>LSCopyDisplayNameForRef</code>	Obtains the display name for an item designated by file-system reference.
<code>LSCopyDisplayNameForURL</code>	Obtains the display name for an item designated by URL.
<code>LSGetExtensionInfo</code>	Obtains the starting index of the extension within a filename.
<code>LSSetExtensionHiddenForRef</code>	Specifies whether the filename extension for an item designated by file-system reference should be hidden or shown.
<code>LSSetExtensionHiddenForURL</code>	Specifies whether the filename extension for an item designated by URL should be hidden or shown.

##### Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kLSCannotSetInfoErr</code>	The filename extension to be hidden cannot be hidden.
<code>kLSInvalidExtensionIndex</code>	The value obtained by the <code>LSGetExtensionInfo</code> function if the filename does not contain a valid extension.
<code>kLSItemInfoExtensionIsHidden</code>	Item has a hidden filename extension.

kLSMinCatInfoBitmap	A minimal catalog information bitmap; no longer used.
kLSRequestExtensionFlagsOnly	Requests only the kLSItemInfoExtensionIsHidden item-information flag.

## AE

---

### AEDataModel.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kAEDebugPOSTHeader	
kAEDebugReplyHeader	
kAEDebugXMLDebugAll	
kAEDebugXMLRequest	
kAEDebugXMLResponse	
kAERPCClass	
kAESharedScriptHandler	
kAESOAPScheme	
kAEXMLRPCScheme	
keyAdditionalHTTPHeaders	
keyAEPOSTHeaderData	
keyAEReplyHeaderData	
keyAEXMLReplyData	
keyAEXMLRequestData	
keyDisableAuthenticationAttr	
keyRPCMethodName	
keyRPCMethodParam	
keySOAPAction	
keySOAPMethodNameSpace	

keySOAPMethodNameSpaceURI	
keySOAPSchemaVersion	
keySOAPSMDNamespace	
keySOAPSMDNamespaceURI	
keySOAPSMDType	
keySOAPStructureMetaData	
keyUserNameAttr	
keyUserPasswordAttr	
keyXMLDebuggingAttr	
kSOAP1999Schema	
kSOAP2001Schema	
typeFileURL	A file URL. That is, the associated data consists of the bytes of a UTF-8 encoded URL with a scheme of "file". This type is appropriate for describing a file that may not yet exist—see Technical Note 2022 for more information.

## AERegistry.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

keyAETSMTextFMFont	
typeISO8601DateTime	

## QD

---

## ATSUnicode.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ATSUCreateFontFallbacks	Creates an opaque object that can be set to contain a font list and a font-search method.
ATSUDisposeFontFallbacks	Disposes of the memory associated with a font fallback object.
ATSUGetObjFontFallbacks	Obtains the font list and font-search method associated with a font fallback object.
ATSUSetObjFontFallbacks	Assigns a font list and a font-search method to a font fallback object.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ATSUFontFallbacks	An opaque structure that contains a font fallback list and font fallback cache information.
kATSULineFontFallbacksTag	Specifies line font fallbacks. The associated value is of type <code>ATSUFontFallbacks</code> . See “Font Fallback Methods” for information on the values that can be associated with this tag.
kATSUTruncateSpecificationMask	Reserved for the truncation specification (0 - 7).
kATSUTruncFeatNoSquishing	Specifies not to perform any negative justification in lieu of truncation.

## Displays.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kDisplayGestaltBrightnessAffectsGammaMask	
kDisplayGestaltCalibratorAttr	
kDisplayGestaltDisplayCommunicationAttr	
kDisplayGestaltForbidI2CMask	
kDisplayGestaltUseI2CPowerMask	
kDisplayGestaltViewAngleAffectsGammaMask	



## Fonts.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

FMGetATSFontFamilyRefFromFontFamily	Obtains the ATS font family reference associated with a font family object.
FMGetATSFontRefFromFont	Obtains the ATS font reference associated with a font object.
FMGetFontContainerFromFontFamilyInstance	Obtains the font container associated with a font family instance.
FMGetFontFamilyFromATSFontFamilyRef	Obtains the font family associated with an ATS font family reference.
FMGetFontFamilyResource	Obtains the font family resource for a font family.
FMGetFontFromATSFontRef	Obtains the font object associated with an ATS font reference.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kFMDefaultActivationContext	Specifies to use the default scope, which is local.
kFMDefaultIterationScope	Specifies to use the default.
kFMGlobalActivationContext	Specifies the scope is global; fonts are available to all applications.
kFMGlobalIterationScope	Specifies the scope is global, iterate over all applications.
kFMLocalActivationContext	Specifies the scope is local; fonts are available only to the application.
kFMLocalIterationScope	Specifies the scope is local, restrict the iteration to the application.

## FontSync.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>FNSProfileCreateWithFSRef</code>	Creates an empty FontSync profile using an FSRef.
<code>FNSProfileOpenWithFSRef</code>	Opens an existing font profile using an FSRef.

## QDPictToCGContext.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>QDPictCreateWithProvider</code>	Creates a QDPict picture, using QuickDraw picture data supplied with a Quartz data provider.
<code>QDPictCreateWithURL</code>	Creates a QDPict picture, using QuickDraw picture data specified with a Core Foundation URL.
<code>QDPictDrawToCGContext</code>	Draws a QuickDraw picture in a Quartz context.
<code>QDPictGetBounds</code>	Returns the intended location and size of a QDPict picture.
<code>QDPictGetResolution</code>	Returns the horizontal and vertical resolution of a QDPict picture.
<code>QDPictRelease</code>	Releases a QDPict picture.
<code>QDPictRetain</code>	Retains a QDPict picture.

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>QDPictRef</code>	Defines an opaque data type that represents a QuickDraw picture in the Quartz 2D graphics environment.
------------------------	--

## Quickdraw.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>IsPortClipRegionEmpty</code>	
<code>IsPortVisibleRegionEmpty</code>	
<code>QDAddRectToDirtyRegion</code>	
<code>QDAddRegionToDirtyRegion</code>	

QDBeginCGContext	Returns a Quartz 2D drawing environment associated with a graphics port.
QDDisposeRegionBits	
QDEndCGContext	Terminates a Quartz 2D drawing environment associated with a graphics port.
QDGetPatternOrigin	
QDRestoreRegionBits	
QDSaveRegionBits	
QDSetPatternOrigin	
QDSwapPort	
RgnToHandle	
SectRegionWithPortClipRegion	
SectRegionWithPortVisibleRegion	
SetPortTextFace	
SetPortTextFont	
SetPortTextMode	
SetPortTextSize	
SwapPortPicSaveHandle	

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

QDRegionBitsRef	
-----------------	--

### Video.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

csCommFlags	
-------------	--

<code>cscProbeConnection</code>	
<code>csMinReplyDelay</code>	
<code>csSupportedCommFlags</code>	
<code>kVideoReplyMicroSecDelayMask</code>	

## ATS

---

### ATSTFont.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>kATSTFontContextLocal</code>	Specifies to use a local context. Fonts with a local context are available to your application.
<code>kATSTOptionFlagsDefaultScope</code>	Specifies to use the default scope, which is equivalent to <code>kATSTOptionFlagsUnRestrictedScope</code> . You can pass this as a parameter to the functions <code>ATSTFontFamilyIteratorCreate</code> , <code>ATSTFontFamilyIteratorReset</code> , <code>ATSTFontIteratorCreate</code> and <code>ATSTFontIteratorReset</code> .
<code>kATSTOptionFlagsIterateByPrecedenceMask</code>	Specifies to iterate fonts in the order dictated by a precedence mask.
<code>kATSTOptionFlagsIterationScopeMask</code>	Specifies mask option bits 12-14 for iteration scopes.
<code>kATSTOptionFlagsRestrictedScope</code>	Specifies to use a restricted scope. You can pass this as a parameter to the functions <code>ATSTFontFamilyIteratorCreate</code> , <code>ATSTFontFamilyIteratorReset</code> , <code>ATSTFontIteratorCreate</code> and <code>ATSTFontIteratorReset</code> .
<code>kATSTOptionFlagsUnRestrictedScope</code>	Specifies to use an unrestricted scope. You can pass this as a parameter to the functions <code>ATSTFontFamilyIteratorCreate</code> , <code>ATSTFontFamilyIteratorReset</code> , <code>ATSTFontIteratorCreate</code> and <code>ATSTFontIteratorReset</code> .

## ATSLayoutTypes.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kATSLineIgnoreFontLeading	Specifies to ignore any leading value specified by a font.
---------------------------	--

## SFNTTypes.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kFontMicrosoftUCS4Script	
--------------------------	--

## PrintCore

---

### PMCore.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

PMSessionCopyDestinationFormat	Obtains the destination format for a print job.
PMSessionCopyDestinationLocation	Obtains a destination location for a print job.
PMSessionCopyOutputFormatList	Obtains an array of destination formats supported by the current print destination.
PMSessionCreatePageFormatList	Obtains a list of page format objects, each of which describes a paper size available on the specified printer.
PMSessionCreatePrinterList	Creates a list of printers available in the specified printing session.
PMSessionGetDestinationType	Obtains the output destination for a print job.
PMSessionSetCurrentPrinter	Changes the current printer for a printing session to a printer specified by name.
PMSessionSetDestination	Sets the destination location, format, and type for a print job.

## PMDefinitions.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kPMDestinationInvalid	Specifies the destination is invalid.
kPMDestinationPreview	Specifies output to print preview.

## PMTicket.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kPMPSTargetLanguageLevel1	Level 1.
kPMPSTargetLanguageLevel1and2	Level 1 compatible, may take advantage of Level 2 and 3 features.
kPMPSTargetLanguageLevel2	Level 2.
kPMPSTargetLanguageLevel2and3	Level 2 compatible, may take advantage of Level 3 features.
kPMPSTargetLanguageLevel3	Level 3.
kPMPSTargetLanguageLevelDefault	Same as kPMPSTargetLanguageLevelUnknown.
kPMPSTargetLanguageLevelUnknown	Language level of target is unknown.

## PPDLib.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

getUIConstraintListH	
PPDAddFeatureEntries	
ppdApplyConstraints	
ppdCheckConstraints	
ppdCheckDates	

ppdClearOptions	
ppdClose	
ppdCloseCompiledPPDFromTicket	
ppdCountUIHeaders	
ppdCountUIOptions	
ppdFindFile	
ppdGetAllOptions	
ppdGetFolder	
ppdGetGenericPPDName	
ppdGetGroupAlias	
ppdGetIndUIHeader	
ppdGetIndUIOption	
ppdGetInvocation	
ppdGetInvocationFile	
ppdGetInvocationLocator	
ppdGetInvocationString	
ppdGetInvocationStruct	
ppdGetMainAlias	
ppdGetMainIndex	
ppdGetMainString	
ppdGetMessageTranslateList	
ppdGetOptionAlias	
ppdGetOptionIndex	
ppdGetOptionString	
ppdGetParseFolder	
ppdGetUIHeader	
ppdGetUIKeyType	
ppdGetUIOption	

ppdMatchPrinter	
ppdOpen	
ppdOpenAndParsePPDAutoSetup	
ppdOpenCompiledPPDFFromTicket	
ppdParseFile	
ppdParseHandle	
ppdSetDefaultOptions	
ppdSetSelection	
setInstallableOptions	

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

chosen	
defaultOption	
forbidden	
groupKeyIndex	
InfoButtonProcPtr	
invalidMainKeyIndex	
invalidOptionKeyIndex	
invocation	
invocationLoc	
InvocationLocator	
invocationSize	
invocationType	
lastModDate	
line	
llx	
lly	



mainKeyIndex	
mainKeyTranslation	
matchModel	
Mindex	
numConstraints	
numOptions	
numStrings	
Oindex	
optionKeyIndex	
optionTranslation	
orderSection	
pickConstraint	
Pindex	
PPDContext	
ppdErrForbidden	
ppdErrNotFound	
PPDEventFilter	
PPDFileSpec	
PPDGetGenericPPDNameProc	
PPDMatchHandle	
PPDMatchPtr	
PPDPrinterDesc	
product	
queryInvocation	
recordOptionPairProc	
StrList	
StrListHdl	
StrListPtr	

Tindex	
UIConstraint	
UIConstraintList	
UIConstraintListH	
UIConstraintListP	
UIConstraintP	
urx	
ury	
WebSearchDlgFilter	
WebSearchProgressProc	
xUIHeader	
xUIHeaderP	
xUIOption	
xUIOptionP	

## CoreGraphics

---

### CGColorSpace.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGColorSpaceCreatePattern	Creates a pattern color space.
CGColorSpaceCreateWithPlatformColorSpace	Creates a platform-specific color space.

### CGContext.h

---

#### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGContextGetInterpolationQuality</code>	Returns the current level of interpolation quality for a graphics context.
<code>CGContextSetFillPattern</code>	Sets the fill pattern in the specified graphics context.
<code>CGContextSetInterpolationQuality</code>	Sets the level of interpolation quality for a graphics context.
<code>CGContextSetStrokePattern</code>	Sets the stroke pattern in the specified graphics context.

### Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGInterpolationQuality</code>	Levels of interpolation quality for rendering an image.
<code>kCGInterpolationDefault</code>	The default level of quality.
<code>kCGInterpolationHigh</code>	A high level of interpolation quality. This setting may slow down image rendering.
<code>kCGInterpolationLow</code>	A low level of interpolation quality. This setting may speed up image rendering.
<code>kCGInterpolationNone</code>	No interpolation.

## CGImage.h

### Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGImageCreateWithJPEGDataProvider</code>	Creates a bitmap image using JPEG-encoded data supplied by a data provider.
--	---

## CGPattern.h

### Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGPatternCreate</code>	Creates a pattern object.
<code>CGPatternRelease</code>	Decrements the retain count of a Quartz pattern.
<code>CGPatternRetain</code>	Increments the retain count of a Quartz pattern.

**Data Types & Constants**

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGPatternCallbacks</code>	A structure that holds a version and two callback functions for drawing a custom pattern.
<code>CGPatternRef</code>	An opaque type that represents a pattern.
<code>CGPatternTiling</code>	Different methods for rendering a tiled pattern.
<code>drawPattern</code>	
<code>kCGPatternTilingConstantSpacing</code>	Pattern cells are spaced consistently, as with <code>kCGPatternTilingConstantSpacingMinimalDistortion</code> . The pattern cell may be distorted additionally to permit a more efficient implementation.
<code>kCGPatternTilingConstantSpacingMinimalDistortion</code>	Pattern cells are spaced consistently. The pattern cell may be distorted by as much as 1 device pixel when the pattern is painted.
<code>kCGPatternTilingNoDistortion</code>	The pattern cell is not distorted when painted. The spacing between pattern cells may vary by as much as 1 device pixel.
<code>releaseInfo</code>	

**CGRemoteOperation.h****Functions**

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>CGEnableEventStateCombining</code>	Enables or disables the merging of actual key and mouse state with the application-specified state in a synthetic event.
<code>CGSetLocalEventsFilterDuringSuppressionState</code>	
<code>CGWindowServerCFMachPort</code>	Returns a Core Foundation mach port ( <code>CFMachPort</code> ) that corresponds to the Mac OS X window server.

**Data Types & Constants**

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CGEventFilterMask	Specify masks for classes of low-level events that can be filtered during event suppression states.
CGEventSupressionState	
kCGEventFilterMaskPermitLocalKeyboardEvents	
kCGEventFilterMaskPermitLocalMouseEvents	
kCGEventFilterMaskPermitSystemDefinedEvents	
kCGEventSupressionStateRemoteMouseDrag	
kCGEventSupressionStateSupressionInterval	
kCGNumberOfEventSupressionStates	

**CGWindowLevel.h**

---

**Data Types & Constants**

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kCGDesktopIconWindowLevelKey	The level for desktop icons.
kCGHelpWindowLevelKey	The level for help windows.
kCGOverlayWindowLevelKey	The level for overlay windows.
kCGUtilityWindowLevelKey	The level for utility windows.

**ColorSync**

---

**CMAApplication.h**

---

**Data Types & Constants**

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

cm4	
-----	--

## CMDeviceIntegration.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

cmDeviceProfileInfoVersion2	Available in Mac OS X v10.1 and later.
CMDeviceProfileScope	
cmIterateAllDeviceProfiles	Iterate all profiles, without replacement.
NCMDeviceProfileInfo	
profileScope	

## CMICCPProfile.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

CMProfileMD5AreEqual	
----------------------	--

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

blueGammaValue	
bluePhosphor	
cm10CLRData	Available in Mac OS X v10.1 and later.
cm11CLRData	Available in Mac OS X v10.1 and later.
cm12CLRData	Available in Mac OS X v10.1 and later.
cm13CLRData	Available in Mac OS X v10.1 and later.
cm14CLRData	Available in Mac OS X v10.1 and later.
cm15CLRData	Available in Mac OS X v10.1 and later.
CM4Header	
cm9CLRData	Available in Mac OS X v10.1 and later.

CMFixedXYColor	
cmICCPProfileVersion4	Available in Mac OS X v10.1 and later.
CMMultiFunctCLUTType	
CMMultiFunctLutA2BType	
CMMultiFunctLutB2AType	
CMMultiFunctLutType	
CMNativeDisplayInfo	Contains color information for a native display.
cmNativeDisplayInfoTag	Available in Mac OS X v10.1 and later.
CMNativeDisplayInfoType	Contains color information and a type descriptor for a native display.
CMParametricCurveType	
cmParametricType0	$Y = X^{\gamma}$
cmParametricType1	$Y = (aX+b)^{\gamma} [X \geq -b/a], Y = 0 [X < -b/a]$
cmParametricType2	$Y = (aX+b)^{\gamma} + c [X \geq -b/a], Y = c [X < -b/a]$
cmParametricType3	$Y = (aX+b)^{\gamma} [X \geq d], Y = cX [X < d]$
cmSigCrdInfoType	Available in Mac OS X v10.1 and later.
cmSigMultiFunctA2BType	Available in Mac OS X v10.1 and later.
cmSigMultiFunctB2AType	Available in Mac OS X v10.1 and later.
cmSigNativeDisplayInfoType	Available in Mac OS X v10.1 and later.
cmSigParametricCurveType	Available in Mac OS X v10.1 and later.
cmSigProfileSequenceDescType	Available in Mac OS X v10.1 and later.
cmTechnologyDigitalCamera	Available in Mac OS X v10.1 and later.
cmYCbCrData	Available in Mac OS X v10.1 and later.
functionType	
gammaChannels	
gammaData	
gammaEntryCount	
gammaEntrySize	
greenGammaValue	

greenPhosphor	
nativeDisplayInfo	
offsetAcurves	
offsetBcurves	
offsetCLUT	
offsetMatrix	
offsetMcurves	
redGammaValue	
redPhosphor	
whitePoint	



# Document Revision History

---

This table describes the changes to *Application Services Reference Update*.

Date	Notes
2007-07-18	Updated with the symbols added to the Application Services framework in Mac OS X v10.5.
2005-07-07	Updated to include additional symbols added to the Application Services framework in Mac OS X v10.4.
2005-04-29	Made minor editorial changes.
	New document that summarizes the symbols added to the Application Services framework in Mac OS X v10.4.

