
NSWorkspace Class Reference

General





Apple Inc.
© 2009 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, Cocoa, FireWire, iChat, Mac, Mac OS, Quartz, QuickDraw, and Spaces are trademarks of Apple Inc., registered in the United States and other countries.

Finder and Spotlight are trademarks of Apple Inc.

OpenGL is a registered trademark of Silicon Graphics, Inc.

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

NSWorkspace Class Reference 9

Overview	9
Tasks	10
Accessing the Shared NSWorkspace Instance	10
Accessing the NSWorkspace Notification Center	10
Opening Files	10
Manipulating Applications	10
Manipulating Files	11
Manipulating Uniform Type Identifier Information	11
Requesting Information	11
Image Animation	12
Requesting Additional Time Before Logout	12
Tracking Changes to the File System	12
Updating Registered Services and File Types	12
Tracking Changes to the Defaults Database	12
Tracking Status Changes for Applications and Devices	13
Providing Custom Icons	13
Unmounting a Device	13
Working with Bundles	13
Managing the Desktop Image	14
Performing Finder Spotlight Searches	14
Finder File Labels	14
Class Methods	14
sharedWorkspace	14
Instance Methods	15
absolutePathForAppBundleWithIdentifier:	15
activateFileViewerSelectingURLs:	15
activeApplication	15
desktopImageOptionsForScreen:	16
desktopImageURLForScreen:	16
duplicateURLs:completionHandler:	17
extendPowerOffBy:	18
fileLabelColors	18
fileLabels	19
filenameExtension:isValidForType:	19
fullPathForApplication:	19
getFileSystemInfoForPath:isRemovable:isWritable:isUnmountable:description:type:	20
getInfoForFile:application:type:	20
hideOtherApplications	21
iconForFile:	21
iconForFiles:	22

iconForFileType:	22
isFilePackageAtPath:	23
launchApplication:	23
launchApplication:showIcon:autoLaunch:	24
launchApplicationAtURL:options:configuration:error:	25
launchAppWithBundleIdentifier:options:additionalEventParamDescriptor: launchIdentifier:	25
launchedApplications	26
localizedDescriptionForType:	27
mountedLocalVolumePaths	27
mountedRemovableMedia	27
noteFileSystemChanged:	28
notificationCenter	29
openFile:	29
openFile:fromImage:at:inView:	30
openFile:withApplication:	30
openFile:withApplication:andDeactivate:	31
openURL:	32
openURLs:withAppBundleIdentifier:options:additionalEventParamDescriptor: launchIdentifiers:	32
performFileOperation:source:destination:files:tag:	33
preferredFilenameExtensionForType:	34
recycleURLs:completionHandler:	34
runningApplications	35
selectFile:inFileViewerRootedAtPath:	36
setDesktopImageURL:forScreen:options:error:	36
setIcon:forFile:options:	37
showSearchResultsForQueryString:	38
type:conformsToType:	38
typeOfFile:error:	39
unmountAndEjectDeviceAtPath:	39
unmountAndEjectDeviceAtURL:error:	40
URLForApplicationToOpenURL:	40
URLForApplicationWithBundleIdentifier:	41
Constants	41
File Types	42
Workspace Launch Configuration Options	42
File Operations	43
Desktop Image Dictionary Keys	45
NSWorkspaceLaunchOptions	45
Volume Mounting Notification User Info Keys	47
NSWorkspaceDidRenameVolumeNotification User Info Keys	47
NSWorkspaceApplicationKey User Info Key	48
Workspace icon creation options	48
Notifications	49
NSWorkspaceWillLaunchApplicationNotification	49
NSWorkspaceDidLaunchApplicationNotification	49

NSWorkspaceDidTerminateApplicationNotification 49
NSWorkspaceSessionDidBecomeActiveNotification 50
NSWorkspaceSessionDidResignActiveNotification 50
NSWorkspaceDidHideApplicationNotification 50
NSWorkspaceDidUnhideApplicationNotification 50
NSWorkspaceDidActivateApplicationNotification 51
NSWorkspaceDidDeactivateApplicationNotification 51
NSWorkspaceDidRenameVolumeNotification 51
NSWorkspaceDidMountNotification 52
NSWorkspaceWillUnmountNotification 52
NSWorkspaceDidUnmountNotification 52
NSWorkspaceDidPerformFileOperationNotification 52
NSWorkspaceDidChangeFileLabelsNotification 53
NSWorkspaceActiveSpaceDidChangeNotification 53
NSWorkspaceDidWakeNotification 53
NSWorkspaceWillPowerOffNotification 53
NSWorkspaceWillSleepNotification 54
NSWorkspaceScreensDidSleepNotification 54
NSWorkspaceScreensDidWakeNotification 54

Appendix A **Deprecated NSWorkspace Methods 55**

Deprecated in Mac OS X v10.6 55
 checkForRemovableMedia 55
 fileSystemChanged 55
 findApplications 56
 mountNewRemovableMedia 56
 noteFileSystemChanged 56
 noteUserDefaultsChanged 57
 openTempFile: 57
 slidImage:from:to: 58
 userDefaultsChanged 58

Document Revision History 61

Index 63

Tables

NSWorkspace Class Reference 9

Table 1	userInfo dictionary keys for activeApplication and launchedApplications and notifications for application launch and termination. 41
---------	--

NSWorkspace Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	Workspace Services Programming Topics
Declared in	NSRunningApplication.h NSWorkspace.h
Related sample code	AppList DesktopImage iChatStatusFromApplication Quartz Composer WWDC 2005 TextEdit SourceView

Overview

An `NSWorkspace` object responds to application requests to perform a variety of services:

- Opening, manipulating, and obtaining information about files and devices
- Tracking changes to the file system, devices, and the user database
- Getting and setting Finder information for files.
- Launching applications

There is one shared `NSWorkspace` object per application. You use the class method `sharedWorkspace` (page 14) to access it. For example, the following statement uses an `NSWorkspace` object to request that a file be opened in the TextEdit application:

```
[[NSWorkspace sharedWorkspace] openFile:@" /Myfiles/README "  
withApplication:@"TextEdit"];
```

Tasks

Accessing the Shared NSWorkspace Instance

- + [sharedWorkspace](#) (page 14)
Returns the shared `NSWorkspace` instance.

Accessing the NSWorkspace Notification Center

- [notificationCenter](#) (page 29)
Returns the notification center for workspace notifications.

Opening Files

- [openFile:](#) (page 29)
Opens the specified file specified using the default application associated with its type.
- [openFile:withApplication:](#) (page 30)
Opens a file using the specified application.
- [openFile:fromImage:at:inView:](#) (page 30)
Opens a file using the default application for its type and animates the action using a custom icon.
- [openFile:withApplication:andDeactivate:](#) (page 31)
Opens the specified file and optionally deactivates the sending application.
- [openURL:](#) (page 32)
Opens the location at the specified URL.
- [openTempFile:](#) (page 57) **Deprecated in Mac OS X v10.6**
Opens the specified temporary file using the default application for its type.

Manipulating Applications

- [launchApplication:](#) (page 23)
Launches the specified application.
- [launchApplication:showIcon:autoLaunch:](#) (page 24)
Launches the specified application using additional options.
- [launchApplicationAtURL:options:configuration:error:](#) (page 25)
Launches the app at the specified URL.
- [hideOtherApplications](#) (page 21)
Hides all applications other than the sender.

Manipulating Files

- [duplicateURLs:completionHandler:](#) (page 17)
Duplicates the specified URLs asynchronously in the same manner as the Finder.
- [recycleURLs:completionHandler:](#) (page 34)
Moves the specified URLs to the trash in the same manner as the Finder.
- [performFileOperation:source:destination:files:tag:](#) (page 33)
Performs a file operation on a set of files in a particular directory.
- [activateFileViewerSelectingURLs:](#) (page 15)
Activates the Finder, and opens one or more windows selecting the specified files.
- [selectFile:inFileViewerRootedAtPath:](#) (page 36)
Selects the file specified by *fullPath*.

Manipulating Uniform Type Identifier Information

- [typeOfFile:error:](#) (page 39)
Returns the uniform type identifier of the specified file, if it can be determined.
- [localizedDescriptionForType:](#) (page 27)
Returns the localized description for the specified Uniform Type Identifier.
- [preferredFilenameExtensionForType:](#) (page 34)
Returns the preferred filename extension for the specified Uniform Type Identifier.
- [filenameExtension:isValidForType:](#) (page 19)
Returns whether the specified filename extension is appropriate for the Uniform Type Identifier.
- [type:conformsToType:](#) (page 38)
Returns a Boolean indicating that the first Uniform Type Identifier conforms to the second Uniform Type Identifier.
- [URLForApplicationWithBundleIdentifier:](#) (page 41)
Returns the URL for the application with the specified identifier.

Requesting Information

- [iconForFile:](#) (page 21)
Returns an image containing the icon for the specified file.
- [iconForFileType:](#) (page 22)
Returns an image containing the icon for files of the specified type.
- [iconForFiles:](#) (page 22)
Returns an image containing the icon for the specified files.
- [getInfoForFile:application:type:](#) (page 20)
Retrieves information about the specified file.
- [URLForApplicationToOpenURL:](#) (page 40)
Returns the URL to the default application that would be used to open the given URL.
- [fullPathForApplication:](#) (page 19)
Returns the full path for the specified application.

- [getFileSystemInfoForPath:isRemovable:isWritable:isUnmountable:description:type:](#) (page 20)
Describes the file system at *fullPath*.
- [isFilePackageAtPath:](#) (page 23)
Determines whether the specified path is a file package.
- [activeApplication](#) (page 15)
Returns a dictionary with information about the current active application.
- [launchedApplications](#) (page 26)
Returns an array of dictionaries, one entry for each running application.

Image Animation

- [slideImage:from:to:](#) (page 58) **Deprecated in Mac OS X v10.6**
Animates a sliding image from one point to another.

Requesting Additional Time Before Logout

- [extendPowerOffBy:](#) (page 18)
Requests the system wait for the specified amount of time before turning off the power or logging out the user.

Tracking Changes to the File System

- [noteFileSystemChanged:](#) (page 28)
Informs the `NSWorkspace` object that the file system changed at the specified path.
- [fileSystemChanged](#) (page 55) **Deprecated in Mac OS X v10.6**
Returns a Boolean value indicating whether a change to the file system has been registered with a [noteFileSystemChanged](#) (page 56) message since the last [fileSystemChanged](#) (page 55) message.
- [noteFileSystemChanged](#) (page 56) **Deprecated in Mac OS X v10.6**
Informs the `NSWorkspace` object that the file system has changed.

Updating Registered Services and File Types

- [findApplications](#) (page 56) **Deprecated in Mac OS X v10.6**
Examines all applications and updates the records of registered services and file types.

Tracking Changes to the Defaults Database

- [noteUserDefaultsChanged](#) (page 57) **Deprecated in Mac OS X v10.6**
Informs the `NSWorkspace` object that the defaults database has changed.

- [userDefaultsChanged](#) (page 58) **Deprecated in Mac OS X v10.6**
Returns a Boolean value indicating whether a change to the defaults database has been registered with a [noteUserDefaultsChanged](#) (page 57) message since the last [userDefaultsChanged](#) (page 58) message.

Tracking Status Changes for Applications and Devices

- [mountedRemovableMedia](#) (page 27)
Returns the full pathnames of all currently mounted removable disks.
- [mountedLocalVolumePaths](#) (page 27)
Returns the mount points of all local volumes, not just the removable ones returned by [mountedRemovableMedia](#) (page 27).
- [runningApplications](#) (page 35)
Returns an array of `NSRunningApplication` representing the running applications.
- [checkForRemovableMedia](#) (page 55) **Deprecated in Mac OS X v10.6**
Polls the system's drives for any disks that have been inserted but not yet mounted.
- [mountNewRemovableMedia](#) (page 56) **Deprecated in Mac OS X v10.6**
Returns the full pathnames of any newly mounted disks.

Providing Custom Icons

- [setIcon:forFile:options:](#) (page 37)
Sets the icon for the file or directory at the specified path.

Unmounting a Device

- [unmountAndEjectDeviceAtPath:](#) (page 39)
Unmounts and ejects the device at the specified path.
- [unmountAndEjectDeviceAtURL:error:](#) (page 40)
Attempts to eject the volume mounted at the given path.

Working with Bundles

- [absolutePathForAppBundleWithIdentifier:](#) (page 15)
Returns the absolute file-system path of an application bundle.
- [launchAppWithBundleIdentifier:options:additionalEventParamDescriptor:launchIdentifier:](#) (page 25)
Launches the application corresponding to the specified *bundleIdentifier*.
- [openURLs:withAppBundleIdentifier:options:additionalEventParamDescriptor:launchIdentifiers:](#) (page 32)
Opens one or more files from an array of URLs.

Managing the Desktop Image

- [desktopImageURLForScreen:](#) (page 16)
Returns the URL for the desktop image for the given screen.
- [setDesktopImageURL:forScreen:options:error:](#) (page 36)
Sets the desktop image for the given screen to the image at the specified URL.
- [desktopImageOptionsForScreen:](#) (page 16)
Returns the desktop image options for the given screen.

Performing Finder Spotlight Searches

- [showSearchResultsForQueryString:](#) (page 38)
Displays a Spotlight search results window in Finder for the specified query string.

Finder File Labels

- [fileLabelColors](#) (page 18)
Returns the corresponding array of file label colors for the file labels.
- [fileLabels](#) (page 19)
Returns the array of file labels as strings.

Class Methods

sharedWorkspace

Returns the shared `NSWorkspace` instance.

```
+ (NSWorkspace *)sharedWorkspace
```

Return Value

The `NSWorkspace` object associated with the process.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

DesktopImage

iChatStatusFromApplication

MyPhoto

Quartz Composer WWDC 2005 TextEdit

SourceView

Declared In

`NSWorkspace.h`

Instance Methods

absolutePathForAppBundleWithIdentifier:

Returns the absolute file-system path of an application bundle.

```
- (NSString *)absolutePathForAppBundleWithIdentifier:(NSString *)bundleIdentifier
```

Parameters

bundleIdentifier

The bundle identifier string. This value corresponds to the value in the `CFBundleIdentifier` key of the application's `Info.plist` file. For example, the bundle identifier of the TextEdit application is `com.apple.TextEdit`.

Return Value

The file system path to the application bundle identified by *bundleIdentifier*, or `nil` if the bundle cannot be found.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

Quartz Composer WWDC 2005 TextEdit

Declared In

`NSWorkspace.h`

activateFileViewerSelectingURLs:

Activates the Finder, and opens one or more windows selecting the specified files.

```
- (void)activateFileViewerSelectingURLs:(NSArray *)fileURLs
```

Parameters

fileURLs

The files to select and display in the Finder.

Availability

Available in Mac OS X v10.6 and later.

Declared In

`NSWorkspace.h`

activeApplication

Returns a dictionary with information about the current active application.

```
- (NSDictionary *)activeApplication
```

Return Value

A dictionary with information about the application. The dictionary contains as many of the keys described in [Table 1](#) (page 41) as are available.

Special Considerations

It is strongly suggested that you use the `NSRunningApplication` methods `currentApplication` or `active` to retrieve this information in post Mac OS X v10.6 targeted applications.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [runningApplications](#) (page 35)
- [launchedApplications](#) (page 26)

Related Sample Code

`iChatStatusFromApplication`

Declared In

`NSWorkspace.h`

desktopImageOptionsForScreen:

Returns the desktop image options for the given screen.

- (NSDictionary *)desktopImageOptionsForScreen:(NSScreen *)screen

Parameters

screen

The screen for which to get the desktop image options.

Return Value

A dictionary containing key-value pairs specified in “[Desktop Image Dictionary Keys](#)” (page 45).

Availability

Available in Mac OS X v10.6 and later.

See Also

- [desktopImageURLForScreen:](#) (page 16)
- [setDesktopImageURL:forScreen:options:error:](#) (page 36)

Related Sample Code

`DesktopImage`

Declared In

`NSWorkspace.h`

desktopImageURLForScreen:

Returns the URL for the desktop image for the given screen.

- (NSURL *)desktopImageURLForScreen:(NSScreen *)screen

Parameters

screen

The screen for which to get the desktop image.

Return Value

The desktop image.

Availability

Available in Mac OS X v10.6 and later.

See Also

- [desktopImageOptionsForScreen:](#) (page 16)
- [setDesktopImageURL:forScreen:options:error:](#) (page 36)

Related Sample Code

AnimatedTableView

DesktopImage

Declared In

NSWorkspace.h

duplicateURLs:completionHandler:

Duplicates the specified URLs asynchronously in the same manner as the Finder..

```
- (void)duplicateURLs:(NSArray *)URLs completionHandler:(void (^)(NSDictionary
    *newURLs, NSError *error))handler
```

Parameters

URLs

The array of URLs to duplicate.

handler

The completion handler block object. If completionHandler is not nil, it will be called when the operation is complete, on the same dispatch queue that was used for the duplicateURLs:completionHandler: call. The completionHandler may be nil if you are not interested in the results.

The block takes two arguments:

newURLs

A dictionary parameter that maps the given URLs to their new URLs locations. Files that could not be duplicated will not be present in the dictionary.

error

If the operation succeeded for every file, the error parameter will be nil. If it failed for one or more files, the error parameter will describe the overall result of the operation in a manner suitable for presentation to the user.

Discussion

This methods may show a progress indicator, or other user interface elements, at AppKit's discretion.

In Mac OS X 10.6, this method require that the main run loop be run in a common mode.

Availability

Available in Mac OS X v10.6 and later.

See Also

- [recycleURLs:completionHandler:](#) (page 34)

Declared In

NSWorkspace.h

extendPowerOffBy:

Requests the system wait for the specified amount of time before turning off the power or logging out the user.

- (NSInteger)extendPowerOffBy:(NSInteger)*requested*

Parameters

requested

The number of milliseconds to wait before turning off the power or logging off the user.

Return Value

The number of milliseconds granted by the system.

Discussion

Currently unimplemented.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSWorkspace.h

fileLabelColors

Returns the corresponding array of file label colors for the file labels.

- (NSArray *)fileLabelColors

Return Value

An array of colors.

Discussion

This array has the same number of elements as [fileLabels](#) (page 19), and the color at a given index corresponds to the label at the same index.

You can listen for notifications named [NSWorkspaceDidChangeFileLabelsNotification](#) (page 53) to be notified when file labels change which may result in changes to the order of the `fileLabelColors`.

Availability

Available in Mac OS X v10.6 and later.

See Also

- [fileLabels](#) (page 19)

Declared In

NSWorkspace.h

fileLabels

Returns the array of file labels as strings.

- (NSArray *)fileLabels

Return Value

An array of strings.

Discussion

You can listen for notifications named [NSWorkspaceDidChangeFileLabelsNotification](#) (page 53) to be notified when file labels change.

Availability

Available in Mac OS X v10.6 and later.

See Also

- [fileLabelColors](#) (page 18)

Declared In

NSWorkspace.h

filenameExtension:isValidForType:

Returns whether the specified filename extension is appropriate for the Uniform Type Identifier.

- (BOOL)filenameExtension:(NSString *)filenameExtension isValidForType:(NSString *)typeName

Parameters

filenameExtension

A string containing the filename extension.

typeName

A string containing the Uniform Type Identifier.

Return Value

YES if *filenameExtension* is a valid extension for *typeName*, NO otherwise

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSWorkspace.h

fullPathForApplication:

Returns the full path for the specified application.

- (NSString *)fullPathForApplication:(NSString *)appName

Parameters

appName

The name of the application.

Return Value

The full path for the application, or `nil` if the specified application was not found.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSWorkspace.h`

getFileSystemInfoForPath:isRemovable:isWritable:isUnmountable:description:type:

Describes the file system at *fullPath*.

```
- (BOOL)getFileSystemInfoForPath:(NSString *)fullPath isRemovable:(BOOL
*)removableFlag isWritable:(BOOL *)writableFlag isUnmountable:(BOOL
*)unmountableFlag description:(NSString **)description type:(NSString
**)fileSystemType
```

Parameters

fullPath

The path to the file-system mount point.

removableFlag

On input, a boolean variable; on return, this variable contains YES if the file system is on removable media.

writableFlag

On input, a boolean variable; on return, this variable contains YES if the file system is writable.

unmountableFlag

On input, a boolean variable; on return, this variable contains YES if the file system is unmountable.

description

On input, a pointer to a string object variable; on return, if the method was successful, this variable contains a string object that describes the file system. You should not rely on this description for program logic but can use it in message strings. Values can include "hard," "nfs," and "foreign."

fileSystemType

On input, a pointer to a string object variable; on return, if the method was successful, this variable contains the file-system type. Values can include "HFS," "UFS," or other values.

Return Value

YES if the information was successfully returned, otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSWorkspace.h`

getInfoForFile:application:type:

Retrieves information about the specified file.

```
- (BOOL)getInfoForFile:(NSString *)fullPath application:(NSString **)appName
type:(NSString **)type
```

Parameters*fullPath*

The full path to the desired file.

appName

The application the system would use to open the file.

type

On input, a pointer to a string object variable; on return, if the method is successful, this variable contains a string object with the filename extension or encoded HFS file type of the file.

Return Value

YES if the information was retrieved successfully; otherwise, NO if the file could not be found or the application was not associated with the file.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [iconForFile:](#) (page 21)
- [iconForFiles:](#) (page 22)

Declared In

NSWorkspace.h

hideOtherApplications

Hides all applications other than the sender.

- (void)hideOtherApplications

Discussion

The user can hide all applications except the current one by Command-Option-clicking on an application's Dock icon.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSWorkspace.h

iconForFile:

Returns an image containing the icon for the specified file.

- (NSImage *)iconForFile:(NSString *)fullPath

Parameters*fullPath*

The full path to the file.

Return Value

The icon associated with the file.

Discussion

The returned image has an initial size of 32 pixels by 32 pixels.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [getInfoForFile:application:type:](#) (page 20)
- [iconForFileType:](#) (page 22)
- [iconForFiles:](#) (page 22)

Related Sample Code

DesktopImage

DictionaryController

SourceView

SpotlightAPI

Spotlighter

Declared In

NSWorkspace.h

iconForFiles:

Returns an image containing the icon for the specified files.

```
- (NSImage *)iconForFiles:(NSArray *)fullPaths
```

Parameters

fullPaths

An array of NSString objects, each of which contains the full path to a file.

Return Value

The icon associated with the group of files.

Discussion

If *fullPaths* specifies one file, that file's icon is returned. If *fullPaths* specifies more than one file, an icon representing the multiple selection is returned.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [iconForFile:](#) (page 21)
- [iconForFileType:](#) (page 22)

Declared In

NSWorkspace.h

iconForFileType:

Returns an image containing the icon for files of the specified type.

- (UIImage *)iconForFileType:(NSString *)*fileType*

Parameters

fileType

The file type, which may be either a filename extension or an encoded HFS file type.

Return Value

The icon associated with files of the given type.

Discussion

The returned image has an initial size of 32 pixels by 32 pixels.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [iconForFile:](#) (page 21)
- [iconForFiles:](#) (page 22)

Related Sample Code

ButtonMadness

IconCollection

MyPhoto

SourceView

ZipBrowser

Declared In

NSWorkspace.h

isFilePackageAtPath:

Determines whether the specified path is a file package.

- (BOOL)isFilePackageAtPath:(NSString *)*fullPath*

Parameters

fullPath

The full path to examine.

Return Value

YES if the path identifies a file package; otherwise, NO if the path does not exist, is not a directory, or is not a file package.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSWorkspace.h

launchApplication:

Launches the specified application.

```
- (BOOL)launchApplication:(NSString *)appName
```

Parameters

appName

The name of the application to open.

Return Value

YES if the application was successfully launched or was already running; otherwise, NO.

Discussion

The *appName* parameter need not be specified with a full path and, in the case of an application wrapper, may be specified with or without the `.app` extension, as described in [Use of .app Extension](#).

Before this method begins, it posts an [NSWorkspaceWillLaunchApplicationNotification](#) (page 49) to the `NSWorkspace` object's notification center. When the operation is complete, it posts an [NSWorkspaceDidLaunchApplicationNotification](#) (page 49).

Availability

Available in Mac OS X v10.0 and later.

See Also

- [launchApplication:showIcon:autoLaunch:](#) (page 24)

Related Sample Code

QTAudioExtractionPanel

Declared In

NSWorkspace.h

launchApplication:showIcon:autoLaunch:

Launches the specified application using additional options.

```
- (BOOL)launchApplication:(NSString *)appName showIcon:(BOOL)showIcon
      autoLaunch:(BOOL)autoLaunch
```

Parameters

appName

The name of the application to open.

showIcon

If NO, the application's icon is not placed on the screen. (The icon still exists, though.)

autoLaunch

If YES, the autoLaunch default is set as though the specified application were autoLaunched at startup.

Return Value

YES if the application was successfully launched or was already running; otherwise, NO.

Discussion

This method is provided to enable daemon-like applications that lack a normal user interface. Its use is not generally encouraged.

Returns YES if the application is successfully launched or already running, and NO if it can't be launched.

Before this method begins, it posts an [NSWorkspaceWillLaunchApplicationNotification](#) (page 49) to the `NSWorkspace` object's notification center. When the operation is complete, it posts an [NSWorkspaceDidLaunchApplicationNotification](#) (page 49).

Availability

Available in Mac OS X v10.0 and later.

See Also

- [launchApplication:](#) (page 23)

Declared In

`NSWorkspace.h`

launchApplicationAtURL:options:configuration:error:

Launches the app at the specified URL.

```
- (NSRunningApplication *)launchApplicationAtURL:(NSURL
        *)url options:(NSWorkspaceLaunchOptions)options configuration:(NSDictionary
        *)configuration error:(NSError **)error
```

Parameters

url

The application URL.

options

Options to use when launching the application. See “[NSWorkspaceLaunchOptions](#)” (page 45) for possible values.

configuration

A dictionary containing the configuration options. Possible key-value pairs are described in “[Workspace Launch Configuration Options](#)” (page 42)

error

Return Value**Discussion****Availability**

Available in Mac OS X v10.6 and later.

Declared In

`NSWorkspace.h`

launchAppWithBundleIdentifier:options:additionalEventParamDescriptor:launchIdentifier:

Launches the application corresponding to the specified *bundleIdentifier*.

```
- (BOOL)launchAppWithBundleIdentifier:(NSString *)bundleIdentifier
        options:(NSWorkspaceLaunchOptions)options
        additionalEventParamDescriptor:(NSAppleEventDescriptor *)descriptor
        launchIdentifier:(NSNumber **)identifier
```

Parameters*bundleIdentifier*

A bundle identifier string. This value corresponds to the value in the `CFBundleIdentifier` key of the application's `Info.plist` file. For example, the bundle identifier of the TextEdit application is `com.apple.TextEdit`.

options

Options to use when launching the application. Values for this parameter are described in [“NSWorkspaceLaunchOptions”](#) (page 45).

descriptor

Additional options specified in an AppleEvent-style descriptor. For example, you could use this parameter to specify additional documents to open when the application is launched.

identifier

On input, a pointer to a number object variable. On return, the variable contains a number object with a unique identifier for the launch attempt. You can use this value to distinguish individual launch requests. This parameter may be `nil`.

Return Value

YES if the application was found and launched; otherwise, NO.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [openURLs:withAppBundleIdentifier:options:additionalEventParamDescriptor:launchIdentifiers:](#) (page 32)

Declared In

`NSWorkspace.h`

launchedApplications

Returns an array of dictionaries, one entry for each running application.

- (NSArray *)launchedApplications

Return Value

An array of `NSDictionary` objects. Each dictionary contains as many of the keys described in [Table 1](#) (page 41) as are available.

Special Considerations

It is strongly suggested that you use the `NSWorkspace` [runningApplications](#) (page 35) class method and `NSRunningApplications` class to retrieve this information in post Mac OS X v10.6 targeted applications.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [runningApplications](#) (page 35)
- [activeApplication](#) (page 15)

Declared In

`NSWorkspace.h`

localizedDescriptionForType:

Returns the localized description for the specified Uniform Type Identifier

```
- (NSString *)localizedDescriptionForType:(NSString *)typeName
```

Parameters

typeName

A string containing the Uniform Type Identifier.

Return Value

An NSString containing the localized description of *typeName*.

Discussion

The localized description is suitable for displaying to the user.

Availability

Available in Mac OS X v10.5 and later.

Related Sample Code

CocoaSlides

Declared In

NSWorkspace.h

mountedLocalVolumePaths

Returns the mount points of all local volumes, not just the removable ones returned by [mountedRemovableMedia](#) (page 27).

```
- (NSArray *)mountedLocalVolumePaths
```

Return Value

An array of NSString objects, each of which contains the full pathname of the mount point for any local volumes.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

SourceView

Declared In

NSWorkspace.h

mountedRemovableMedia

Returns the full pathnames of all currently mounted removable disks.

```
- (NSArray *)mountedRemovableMedia
```

Return Value

An array of NSString objects, each of which contains the full pathname of a mounted removable disk.

Discussion

If the computer provides an interrupt or other notification when the user inserts a disk into a drive, the Finder will mount the disk immediately. However, if no notification is given, the Finder won't be aware that a disk needs to be mounted. On such systems, an application should invoke either [mountNewRemovableMedia](#) (page 56) or [checkForRemovableMedia](#) (page 55) before invoking [mountedRemovableMedia](#) (page 27). Either of these methods cause the Finder to poll the drives to see if a disk is present. If a disk has been inserted but not yet mounted, these methods will cause the Finder to mount it.

The Disk button in an Open or Save panel invokes [mountedRemovableMedia](#) (page 27) and [mountNewRemovableMedia](#) (page 56) as part of its operation, so most applications won't need to invoke these methods directly.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [checkForRemovableMedia](#) (page 55)
- [mountNewRemovableMedia](#) (page 56)

Related Sample Code

CocoaDVDPlayer

Declared In

NSWorkspace.h

noteFileSystemChanged:

Informs the `NSWorkspace` object that the file system changed at the specified path.

```
- (void)noteFileSystemChanged:(NSString *)path
```

Parameters

path

The full path that changed.

Discussion

The `NSWorkspace` object then gets the status of all the files and directories it is interested in and updates itself appropriately. This method is used by many objects that write or delete files.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [fileSystemChanged](#) (page 55)

Related Sample Code

Quartz Composer WWDC 2005 TextEdit

Declared In

NSWorkspace.h

notificationCenter

Returns the notification center for workspace notifications.

- (NSNotificationCenter *)notificationCenter

Return Value

The notification center object.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

CocoaDVDPlayer

iChatStatusFromApplication

Declared In

NSWorkspace.h

openFile:

Opens the specified file specified using the default application associated with its type.

- (BOOL)openFile:(NSString *)fullPath

Parameters

fullPath

The full path to the file.

Return Value

YES if the file was successfully opened; otherwise, NO.

Discussion

The sending application is deactivated before the request is sent.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [openFile:fromImage:at:inView:](#) (page 30)
- [openFile:withApplication:](#) (page 30)
- [openFile:withApplication:andDeactivate:](#) (page 31)
- [openTempFile:](#) (page 57)

Related Sample Code

CIAnnotation

DictionaryController

iSpend

Quartz Composer WWDC 2005 TextEdit

UIElementInspector

Declared In

NSWorkspace.h

openFile:fromImage:at:inView:

Opens a file using the default application for its type and animates the action using a custom icon.

```
- (BOOL)openFile:(NSString *)fullPath fromImage:(NSImage *)anImage at:(NSPoint)point  
inView:(NSView *)aView
```

Parameters

fullPath

The full path to the file.

anImage

The icon for the file.

point

The point in *aView* at which to display the icon.

aView

The view in which to display the icon.

Return Value

YES if the file was successfully opened; otherwise, NO.

Discussion

The Finder provides an animation before opening the file to give the user feedback that the file is to be opened. To provide this animation, *anImage* should contain an icon for the file, and its image should be displayed at *point*, specified in the coordinates of *aView*. Currently provides the same functionality as [openFile:](#) (page 29).

The sending application is deactivated before the request is sent.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [openFile:](#) (page 29)
- [openFile:withApplication:](#) (page 30)
- [openFile:withApplication:andDeactivate:](#) (page 31)
- [openTempFile:](#) (page 57)

Declared In

NSWorkspace.h

openFile:withApplication:

Opens a file using the specified application.

```
- (BOOL)openFile:(NSString *)fullPath withApplication:(NSString *)appName
```

Parameters

fullPath

The full path to the file.

appName

The name of the application to use when opening the file.

Return Value

YES if the file was successfully opened; otherwise, NO.

Discussion

The *appName* parameter need not be specified with a full path and, in the case of an application wrapper, may be specified with or without the `.app` extension, as described in Use of .app Extension. The sending application is deactivated before the request is sent.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [openFile:](#) (page 29)
- [openFile:withApplication:andDeactivate:](#) (page 31)

Related Sample Code

CIVideoDemoGL
 QTExtractAndConvertToAIFF
 QTExtractAndConvertToMovieFile
 Quartz Composer WWDC 2005 TextEdit
 WhackedTV

Declared In

NSWorkspace.h

openFile:withApplication:andDeactivate:

Opens the specified file and optionally deactivates the sending application.

```
- (BOOL)openFile:(NSString *)fullPath withApplication:(NSString *)appName
  andDeactivate:(BOOL)flag
```

Parameters

fullPath

The full path to the file.

appName

The name of the application to use when opening the file.

flag

If YES, the sending application is deactivated before the request is sent, allowing the opening application to become the active application.

Return Value

YES if the file was successfully opened; otherwise, NO.

Discussion

The *appName* parameter need not be specified with a full path and, in the case of an application wrapper, may be specified with or without the `.app` extension, as described in Use of .app Extension. If *appName* is `nil`, the default application for the file's type is used.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [openFile:](#) (page 29)
- [openFile:withApplication:](#) (page 30)
- [application:openFile:](#) (NSApplication delegate method)

Related Sample Code

Core Data HTML Store

DispatchFractal

Declared In

NSWorkspace.h

openURL:

Opens the location at the specified URL.

```
- (BOOL)openURL:(NSURL *)url
```

Parameters*url*

A URL specifying the location to open.

Return Value

YES if the location was successfully opened; otherwise, NO.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

NSOperationSample

ObjectPath

PhotoSearch

PredicateEditorSample

QTCompressionOptionsWindow

Declared In

NSWorkspace.h

openURLs:withAppBundleIdentifier:options:additionalEventParamDescriptor:launchIdentifiers:

Opens one or more files from an array of URLs.

```
- (BOOL)openURLs:(NSArray *)urls withAppBundleIdentifier:(NSString *)bundleIdentifier
    options:(NSWorkspaceLaunchOptions)options
    additionalEventParamDescriptor:(NSAppleEventDescriptor *)descriptor
    launchIdentifiers:(NSArray **)identifiers
```

Parameters*urls*

An array of NSURL objects, each one identifying a URL for the application to open.

bundleIdentifier

A bundle identifier string or `nil` to use the default system bindings. This value corresponds to the value in the `CFBundleIdentifier` key of the application's `Info.plist` file. For example, the bundle identifier of the TextEdit application is `com.apple.TextEdit`.

options

Options to use when launching the application. Values for this parameter are described in [“NSWorkspaceLaunchOptions”](#) (page 45).

descriptor

Additional options specified in an AppleEvent-style descriptor. For example, you could use this parameter to specify additional documents to open when the application is launched.

identifiers

On input, a pointer to an array object variable. On return, the variable contains an array of `NSNumber` objects. Each number object contains a unique identifier (one for each URL) for the launch attempt. You can use these values to distinguish individual launch requests. This parameter may be `nil`.

Return Value

YES if the application was found and launched; otherwise, NO.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [launchAppWithBundleIdentifier:options:additionalEventParamDescriptor:launchIdentifier:](#) (page 25)

Related Sample Code

NewsReader

Declared In

NSWorkspace.h

performFileOperation:source:destination:files:tag:

Performs a file operation on a set of files in a particular directory.

```
- (BOOL)performFileOperation:(NSString *)operation source:(NSString *)source
    destination:(NSString *)destination files:(NSArray *)files tag:(NSInteger *)tag
```

Parameters*operation*

The file operation to perform. The possible values for this parameter are described in [“File Operations”](#) (page 43).

source

The full path to the directory containing the files on which to operate.

destination

The full path to the destination directory of the operation.

files

An array of `NSString` objects specifying the names of the files and directories to be manipulated. Each string must not contain any path information other than the name of the file or directory. In other words, all of the files and directories must be located in the source directory and not in one of its subdirectories.

tag

On input, an integer variable; on return, this variable contains a negative integer if the operation fails, 0 if the operation was performed synchronously and succeeded, or a positive integer if the operation was performed asynchronously. If the value is a positive integer, the value is a tag that identifies the requested file operation.

Return Value

YES if the operation succeeded; otherwise, NO.

Discussion

Some operations—such as moving, copying, and linking files—require a destination directory to be specified. If not, *destination* should be the empty string (@" "). Before this method returns, it posts an [NSWorkspaceDidPerformFileOperationNotification](#) (page 52) to the NSWorkspace object's notification center.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Quartz Composer WWDC 2005.TextEdit

Declared In

NSWorkspace.h

preferredFilenameExtensionForType:

Returns the preferred filename extension for the specified Uniform Type Identifier.

```
- (NSString *)preferredFilenameExtensionForType:(NSString *)typeName
```

Parameters

typeName

A string containing the Uniform Type Identifier.

Return Value

The appropriate filename extension for *typeName*, or *nil* if no extension could be determined.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSWorkspace.h

recycleURLs:completionHandler:

Moves the specified URLs to the trash in the same manner as the Finder.

```
- (void)recycleURLs:(NSArray *)URLs completionHandler:(void (^)(NSDictionary *newURLs, NSError *error))handler
```

Parameters

URLs

The array of URLs to move to the trash.

handler

The completion handler block object. If `completionHandler` is not `nil`, it will be called when the operation is complete, on the same dispatch queue that was used for the `recycleURLs:completionHandler:` call. The `completionHandler` may be `nil` if you are not interested in the results.

The block takes two arguments:

newURLs

A dictionary parameter that maps the given URLs to their new URLs locations in the trash. Files that could not be moved to the trash will not be present in the dictionary.

error

If the operation succeeded for every file, the error parameter will be `nil`. If it failed for one or more files, the error parameter will describe the overall result of the operation in a manner suitable for presentation to the user.

Discussion

This methods may show a progress indicator, or other user interface elements, at AppKit's discretion.

In Mac OS X 10.6, this method require that the main run loop be run in a common mode.

Availability

Available in Mac OS X v10.6 and later.

See Also

- [duplicateURLs:completionHandler:](#) (page 17)

Declared In

NSWorkspace.h

runningApplications

Returns an array of `NSRunningApplication` representing the running applications.

- (NSArray *)runningApplications

Return Value

An array of `NSRunningApplication` instances.

Discussion

The order of the array is unspecified, but it is stable, meaning that the relative order of particular applications will not change across multiple calls to `runningApplications`. See *NSRunningApplication Class Reference* for more information on `NSRunningApplication`.

Similar to the `NSRunningApplication` classes's properties, this property will only change when the main run loop is run in a common mode. Instead of polling, use key-value observing to be notified of changes to this array property.

This property is thread safe, in that it may be called from background threads and the result is returned atomically.

This property is observable using key-value observing.

Availability

Available in Mac OS X v10.6 and later.

Declared In

NSRunningApplication.h

selectFile:inFileViewerRootedAtPath:

Selects the file specified by *fullPath*.

```
- (BOOL)selectFile:(NSString *)fullPath inFileViewerRootedAtPath:(NSString *)rootFullPath
```

Parameters

fullPath

The full path of the file to select.

rootFullPath

If a path is specified, a new file viewer is opened. If you specify an empty string (@" ") for this parameter, the file is selected in the main viewer.

Return Value

YES if the file was successfully selected; otherwise, NO.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

EnhancedAudioBurn

Declared In

NSWorkspace.h

setDesktopImageURL:forScreen:options:error:

Sets the desktop image for the given screen to the image at the specified URL.

```
- (BOOL)setDesktopImageURL:(NSURL *)url forScreen:(NSScreen *)screen options:(NSDictionary *)options error:(NSError **)error
```

Parameters

url

A file URL to the image. The URL must not be nil.

screen

The screen to set the desktop image on.

options

The options dictionary may contain any of the [“Desktop Image Dictionary Keys”](#) (page 45) keys, which control how the image is scaled on the screen.

error

A error that is returned by-reference if setting the image fails.

Return Value

YES if the image was set as the desktop, otherwise NO. If NO is returned, the *error* parameter provides additional information.

Discussion

You should not present a user interface for picking the options. Instead, choose appropriate defaults and allow the user to adjust them in the System Preference Pane.

Availability

Available in Mac OS X v10.6 and later.

See Also

- [desktopImageOptionsForScreen](#): (page 16)
- [desktopImageURLForScreen](#): (page 16)

Related Sample Code

AnimatedTableView

DesktopImage

Declared In

NSWorkspace.h

setIcon:forFile:options:

Sets the icon for the file or directory at the specified path.

```
- (BOOL)setIcon:(NSImage *)image forFile:(NSString *)fullPath
options:(NSWorkspaceIconCreationOptions)options
```

Parameters

image

The image to use as the icon for the file or directory.

fullPath

The full path of the file or directory.

options

The icon representations to generate from the image. You specify this value by combining the appropriate “[Workspace icon creation options](#)” (page 48) constants, using the C bitwise OR operator. Specify 0 if you want to generate icons in all available icon representation formats.

Return Value

YES if the icon was set; otherwise, NO.

Discussion

The *image* can be an arbitrary image, with or without transparency. This image is automatically scaled (as needed) to generate the icon representations. The file or folder must exist and be writable by the user.

It is recommended that applications include the `NSExclude10_4ElementsIconCreationOption` option for compatibility with pre-Mac OS X v10.3 Finder. Icons that include the high resolution elements prevent custom icons from being displayed on earlier systems.

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSWorkspace.h

showSearchResultsForQueryString:

Displays a Spotlight search results window in Finder for the specified query string.

```
- (BOOL)showSearchResultsForQueryString:(NSString *)queryString
```

Parameters*queryString*

The string to search for.

Return Value

YES if the communication with Finder was successful, otherwise NO.

Discussion

Finder becomes the active application, if possible. The user can further refine the search via the Finder user interface.

Availability

Available in Mac OS X v10.6 and later.

Declared In

NSWorkspace.h

type:conformsToType:

Returns a Boolean indicating that the first Uniform Type Identifier conforms to the second Uniform Type Identifier.

```
- (BOOL)type:(NSString *)firstTypeName conformsToType:(NSString *)secondTypeName
```

Parameters*firstTypeName*

A string containing the Uniform Type Identifier that should conform to *secondTypeName*.

secondTypeName

A string containing a Uniform Type Identifier.

Return Value

YES if *firstTypeName* conforms to the uniform type identifier hierarchy of *secondTypeName*, NO otherwise.

Discussion

Use this method instead of comparing Uniform Identifier Types for equality. See *Uniform Type Identifiers Overview* for information about Uniform Type Identifier conformance.

This method will always return YES if the two strings are equal. It is appropriate to use this method with other type names, including those declared in `CFBundleTypeNameInfo.plist` entries.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSWorkspace.h

typeOfFile:error:

Returns the uniform type identifier of the specified file, if it can be determined..

```
- (NSString *)typeOfFile:(NSString *)absoluteFilePath error:(NSError **)outError
```

Parameters

absoluteFilePath

The absolute path of the file.

outError

If the Uniform Type Identifier of the file at *absolutePath* can't be determined, *outError* contains an `NSError` object that describes why.

Return Value

An `NSString` containing the uniform type identifier of the file at *absoluteFilePath*. If no UTI can be determined the return value is `nil`.

Discussion

If the file at the specified path is a symbolic link, the type of the symbolic link is returned.

Availability

Available in Mac OS X v10.5 and later.

Related Sample Code

CocoaSlides

Declared In

`NSWorkspace.h`

unmountAndEjectDeviceAtPath:

Unmounts and ejects the device at the specified path.

```
- (BOOL)unmountAndEjectDeviceAtPath:(NSString *)path
```

Parameters

path

The path to the device.

Return Value

YES if the device was unmounted; otherwise, NO.

Discussion

When this method begins, it posts an [NSWorkspaceWillUnmountNotification](#) (page 52) to the `NSWorkspace` object's notification center. When it is finished, it posts an [NSWorkspaceDidUnmountNotification](#) (page 52).

The [unmountAndEjectDeviceAtURL:error:](#) (page 40) is preferable as it will provide more detailed error information.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [unmountAndEjectDeviceAtURL:error:](#) (page 40)

Related Sample Code

CocoaDVDPlayer

Declared In

NSWorkspace.h

unmountAndEjectDeviceAtURL:error:

Attempts to eject the volume mounted at the given path.

- (BOOL)unmountAndEjectDeviceAtURL:(NSURL *)url error:(NSError **)error

Parameters*url*

The URL of the volume to eject.

error

If the operation fails, this error contains more information about the failure.

Return Value

YES if the volume was unmounted and ejected successfully, otherwise NO, for example, if the volume is not ejectable.

Availability

Available in Mac OS X v10.6 and later.

See Also- [unmountAndEjectDeviceAtPath:](#) (page 39)**Declared In**

NSWorkspace.h

URLForApplicationToOpenURL:

Returns the URL to the default application that would be used to open the given URL.

- (NSURL *)URLForApplicationToOpenURL:(NSURL *)url

Parameters*url*

The URL of the file to open.

Return ValueThe URL of the default application that would open the specified *url*. Returns nil if no application is able to open the url, or if the file url does not exist.**Discussion**

This is the programmatic equivalent of double clicking a document in the Finder.

Availability

Available in Mac OS X v10.6 and later.

Declared In

NSWorkspace.h

URLForApplicationWithBundleIdentifier:

Returns the URL for the application with the specified identifier.

```
- (NSURL *)URLForApplicationWithBundleIdentifier:(NSString *)bundleIdentifier
```

Parameters

bundleIdentifier

A bundle identifier specifying an application.

Return Value

The URL of the application, or `nil` if no application has the bundle identifier.

Discussion

This uses various (currently unspecified) heuristics in case multiple apps have the same bundle ID.

Availability

Available in Mac OS X v10.6 and later.

Declared In

NSWorkspace.h

Constants

The following table describes keys for an `NSDictionary` object containing information about an application. This dictionary is returned by [activeApplication](#) (page 15) and [launchedApplications](#) (page 26), and is also provided in the `userInfo` of `NSWorkspace` notifications for application launch and termination.

Note: It is strongly suggested that you use the `NSWorkspacerunningApplications` (page 35) class method and `NSRunningApplication` class to retrieve this information in post Mac OS X v10.6 targeted applications. Rather than [activeApplication](#) (page 15) and [launchedApplications](#) (page 26).

These constants are considered legacy.

Table 1 `userInfo` dictionary keys for `activeApplication` and `launchedApplications` and notifications for application launch and termination.

Key	Value
@ <code>"NSApplicationPath"</code>	The full path to the application, as a <code>NSString</code> object.
@ <code>"NSApplicationName"</code>	The application's name, as an <code>NSString</code> object.
@ <code>"NSApplicationBundleIdentifier"</code>	The application's bundle identifier, as an <code>NSString</code> object.
@ <code>"NSApplicationProcessIdentifier"</code>	The application's process id, as an <code>NSNumber</code> object.
@ <code>"NSApplicationProcessSerialNumber-High"</code>	The high long of the process serial number (PSN), as an <code>NSNumber</code> object.
@ <code>"NSApplicationProcessSerialNumber-Low"</code>	The low long of the process serial number (PSN), as an <code>NSNumber</code> object.

File Types

These constants specify different types of files returned by [getInfoForFile:application:type:](#) (page 20).

```
NSString *NSPlainFileType;
NSString *NSDirectoryFileType;
NSString *NSApplicationFileType;
NSString *NSFileSystemFileType;
NSString *NSShellCommandFileType;
```

Constants

`NSPlainFileType`

Plain (untyped) file

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

Declared in `NSWorkspace.h`.

`NSDirectoryFileType`

Directory

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

Declared in `NSWorkspace.h`.

`NSApplicationFileType`

Cocoa application

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

Declared in `NSWorkspace.h`.

`NSFileSystemFileType`

File-system mount point

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

Declared in `NSWorkspace.h`.

`NSShellCommandFileType`

Executable shell command

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

Declared in `NSWorkspace.h`.

Declared In

`NSWorkspace.h`

Workspace Launch Configuration Options

The following keys can be used in the configuration dictionary of the [launchApplicationAtURL:options:configuration:error:](#) (page 25) method. Each key is optional, and if omitted, default behavior is applied.

```
NSString * const NSWorkspaceLaunchConfigurationAppleEvent;
NSString * const NSWorkspaceLaunchConfigurationArguments;
NSString * const NSWorkspaceLaunchConfigurationEnvironment;
NSString * const NSWorkspaceLaunchConfigurationArchitecture;
```

Constants

`NSWorkspaceLaunchConfigurationAppleEvent`

The value is the first `NSAppleEventDescriptor` to send to the new application. If an instance of the application is already running, this is sent to that application.

Available in Mac OS X v10.6 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceLaunchConfigurationArguments`

The value is an `NSArray` of `NSStrings`, passed to the new application in the `argv` parameter. Ignored if a new instance of the application is not launched.

Available in Mac OS X v10.6 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceLaunchConfigurationEnvironment`

The value is an `NSDictionary`, mapping `NSStrings` to `NSStrings`, containing environment variables to set for the new app. Ignored if a new instance of the application is not launched.

Available in Mac OS X v10.6 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceLaunchConfigurationArchitecture`

The value is an `NSNumber` containing an `Mach-O Architecture` constant. Ignored if a new instance of the application is not launched.

Available in Mac OS X v10.6 and later.

Declared in `NSWorkspace.h`.

File Operations

These constants specify different types of file operations used by [performFileOperation:source:destination:files:tag:](#) (page 33).

```
NSString *NSWorkspaceMoveOperation;
NSString *NSWorkspaceCopyOperation;
NSString *NSWorkspaceLinkOperation;
NSString *NSWorkspaceCompressOperation;
NSString *NSWorkspaceDecompressOperation;
NSString *NSWorkspaceEncryptOperation;
NSString *NSWorkspaceDecryptOperation;
NSString *NSWorkspaceDestroyOperation;
NSString *NSWorkspaceRecycleOperation;
NSString *NSWorkspaceDuplicateOperation;
```

Constants

`NSWorkspaceMoveOperation`

Move file to destination. Behaves the same as `movePath:toPath:handler:.`

Available in Mac OS X v10.0 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceCopyOperation`

Copy file to destination. Behaves the same as `copyPath:toPath:handler:.`

Available in Mac OS X v10.0 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceLinkOperation`

Create hard link to file in destination. Behaves the same as `linkPath:toPath:handler:.`

Available in Mac OS X v10.0 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceCompressOperation`

Compress file. This operation always returns an error.

Available in Mac OS X v10.0 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceDecompressOperation`

Decompress file. This operation always returns an error.

Available in Mac OS X v10.0 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceEncryptOperation`

Encrypt file. This operation always returns an error.

Available in Mac OS X v10.0 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceDecryptOperation`

Decrypt file. This operation always returns an error.

Available in Mac OS X v10.0 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceDestroyOperation`

Destroy file. Behaves the same as `removeFileAtPath:handler:.`

Available in Mac OS X v10.0 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceRecycleOperation`

Move file to trash. The file is moved to the trash folder on the volume containing the file using the same semantics as `NSWorkspaceMoveOperation`. If a file with the same name currently exists in the trash folder, the new file is renamed. If no trash folder exists on the volume containing the file, the operation fails.

Available in Mac OS X v10.0 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceDuplicateOperation`

Duplicate file in source directory.

Available in Mac OS X v10.0 and later.

Declared in `NSWorkspace.h`.

Declared In

`NSWorkspace.h`

Desktop Image Dictionary Keys

The following keys may be specified or returned in the options dictionary for [setDesktopImageURL:forScreen:options:error:](#) (page 36).

```
NSString * const NSWorkspaceDesktopImageScalingKey;
NSString * const NSWorkspaceDesktopImageAllowClippingKey;
NSString * const NSWorkspaceDesktopImageFillColorKey;
```

Constants

`NSWorkspaceDesktopImageScalingKey`

The value is an `NSNumber` containing an `NSImageScaling` constant as declared in `NSCell`. If this is not specified, `NSImageScaleProportionallyUpOrDown` is used. `NSImageScaleProportionallyDown` is not currently supported.

Available in Mac OS X v10.6 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceDesktopImageAllowClippingKey`

The value is an `NSNumber` containing a `BOOL`, which affects the interpretation of Proportional scaling types. A `NO` value will make the image fully visible, but there may be empty space on the sides or top and bottom. A `YES` value will cause the image to fill the entire screen, but the image may be clipped. If this is not specified, `NO` is assumed. Non-proportional scaling types ignore this value.

Available in Mac OS X v10.6 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceDesktopImageFillColorKey`

The value is an `NSColor`, which is used to fill any empty space around the image. If not specified, a default value is used. Currently, only colors that use or can be converted to use `NSCalibratedRGBColorSpace` are supported, and any alpha value is ignored.

Available in Mac OS X v10.6 and later.

Declared in `NSWorkspace.h`.

NSWorkspaceLaunchOptions

These constants define launch options you can pass to

[launchAppWithBundleIdentifier:options:additionalEventParamDescriptor:](#)

[launchIdentifier:](#) (page 25) and

[openURLs:withAppBundleIdentifier:options:additionalEventParamDescriptor:](#)

[launchIdentifiers:](#) (page 32).

```
enum {
    NSWorkspaceLaunchAndPrint = 0x00000002,
    NSWorkspaceLaunchInhibitingBackgroundOnly = 0x00000080,
    NSWorkspaceLaunchWithoutAddingToRecents = 0x00000100,
    NSWorkspaceLaunchWithoutActivation = 0x00000200,
    NSWorkspaceLaunchAsync = 0x00010000,
    NSWorkspaceLaunchAllowingClassicStartup = 0x00020000,
    NSWorkspaceLaunchPreferringClassic = 0x00040000,
    NSWorkspaceLaunchNewInstance = 0x00080000,
    NSWorkspaceLaunchAndHide = 0x00100000,
    NSWorkspaceLaunchAndHideOthers = 0x00200000,
    NSWorkspaceLaunchDefault = NSWorkspaceLaunchAsync |
    NSWorkspaceLaunchAllowingClassicStartup
};
typedef NSUInteger NSWorkspaceLaunchOptions;
```

Constants

`NSWorkspaceLaunchAndPrint`

Print items instead of opening them.

Available in Mac OS X v10.3 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceLaunchInhibitingBackgroundOnly`

Causes launch to fail if the target is background-only.

Available in Mac OS X v10.3 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceLaunchWithoutAddingToRecents`

Do not add the application or documents to the Recents menu.

Available in Mac OS X v10.3 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceLaunchWithoutActivation`

Launch the application but do not bring it into the foreground.

Available in Mac OS X v10.3 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceLaunchAsync`

Launch the application and return the results asynchronously.

Available in Mac OS X v10.3 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceLaunchAllowingClassicStartup`

Start up the Classic compatibility environment, if it is required by the application.

Available in Mac OS X v10.3 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceLaunchPreferringClassic`

Force the application to launch in the Classic compatibility environment.

Available in Mac OS X v10.3 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceLaunchNewInstance`

Create a new instance of the application, even if one is already running.

Available in Mac OS X v10.3 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceLaunchAndHide`

Tell the application to hide itself as soon as it has finished launching.

Available in Mac OS X v10.3 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceLaunchAndHideOthers`

Hide all applications except the newly launched one.

Available in Mac OS X v10.3 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceLaunchDefault`

Launch the application asynchronously and launch it in the Classic environment, if required.

Available in Mac OS X v10.3 and later.

Declared in `NSWorkspace.h`.

Volume Mounting Notification User Info Keys

The following keys are available in the `userInfo` parameter of the notification named [NSWorkspaceDidRenameVolumeNotification](#) (page 51).

```
NSString * const NSWorkspaceVolumeLocalizedNameKey;  
NSString * const NSWorkspaceVolumeURLKey;
```

Constants

`NSWorkspaceVolumeLocalizedNameKey`

NSString containing the user-visible name of the volume.

Available in Mac OS X v10.6 and later.

Declared in `NSWorkspace.h`.

`NSWorkspaceVolumeURLKey`

NSURL containing the mount path of the volume.

Available in Mac OS X v10.6 and later.

Declared in `NSWorkspace.h`.

NSWorkspaceDidRenameVolumeNotification User Info Keys

The following keys are available in the `userInfo` parameter of the notification named [NSWorkspaceDidRenameVolumeNotification](#) (page 51).

```
NSString * const NSWorkspaceVolumeOldLocalizedNameKey;
NSString * const NSWorkspaceVolumeOldURLKey;
```

Constants

`NSWorkspaceVolumeOldLocalizedNameKey`
 NSString containing the old user-visible name of the volume
 Available in Mac OS X v10.6 and later.
 Declared in `NSWorkspace.h`.

`NSWorkspaceVolumeOldURLKey`
 NSURL containing the old mount path of the volume
 Available in Mac OS X v10.6 and later.
 Declared in `NSWorkspace.h`.

NSWorkspaceApplicationKey User Info Key

This constant is supplied in the `userInfo` dictionary of various notifications.

```
NSString * const NSWorkspaceApplicationKey;
```

Constants

`NSWorkspaceApplicationKey`
 The value corresponding to this key is an instance of `NSRunningApplication` that reflects the affected application.
 Available in Mac OS X v10.6 and later.
 Declared in `NSWorkspace.h`.

Workspace icon creation options

These constants describe the `NSWorkspaceIconCreationOptions` values used by [setIcon:forFile:options:](#) (page 37). You can combine these using the C bitwise OR operator.

```
enum {
    NSExcludeQuickDrawElementsIconCreationOption = 1 << 1,
    NSExclude10_4ElementsIconCreationOption     = 1 << 2
};
typedef NSUInteger NSWorkspaceIconCreationOptions;
```

Constants

`NSExcludeQuickDrawElementsIconCreationOption`
 Suppress generation of the QuickDraw format icon representations that are used Mac OS X v10.0 through v10.4.
 Available in Mac OS X v10.4 and later.
 Declared in `NSWorkspace.h`.

`NSExclude10_4ElementsIconCreationOption`
 Suppress generation of the new higher resolution icon representations that are supported in Mac OS X v10.4.
 Available in Mac OS X v10.4 and later.
 Declared in `NSWorkspace.h`.

Notifications

All `NSWorkspace` notifications are posted to the `NSWorkspace` object's own notification center, not the application's default notification center. Access this center using the `NSWorkspace` object's `notificationCenter` (page 29) method.

NSWorkspaceWillLaunchApplicationNotification

Posted when the Finder is about to launch an application.

The notification object is the shared `NSWorkspace` instance. In Mac OS X v10.6 and later the `userInfo` dictionary contains the `NSWorkspaceApplicationKey` (page 48) key with a corresponding instance of `NSRunningApplication` that represents the affected application.

In Mac OS X v10.5 and earlier the `userInfo` dictionary contains the keys and values described in [Table 1](#) (page 41).

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSWorkspace.h`

NSWorkspaceDidLaunchApplicationNotification

Posted when a new application has started up.

The notification object is the shared `NSWorkspace` instance. In Mac OS X v10.6 and later the `userInfo` dictionary contains the `NSWorkspaceApplicationKey` (page 48) key with a corresponding instance of `NSRunningApplication` that represents the affected application.

In Mac OS X v10.5 and earlier the `userInfo` dictionary contains the keys and values described in [Table 1](#) (page 41).

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSWorkspace.h`

NSWorkspaceDidTerminateApplicationNotification

Posted when an application finishes executing.

The notification object is the shared `NSWorkspace` instance. In Mac OS X v10.6 and later the `userInfo` dictionary contains the `NSWorkspaceApplicationKey` (page 48) key with a corresponding instance of `NSRunningApplication` that represents the affected application.

In Mac OS X v10.5 and earlier the `userInfo` dictionary contains the keys and values described in [Table 1](#) (page 41).

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSWorkspace.h

NSWorkspaceSessionDidBecomeActiveNotification

Posted after a user session is switched in. This allows an application to re-enable some processing when a switched out session gets switched back in, for example.

The notification object is the shared `NSWorkspace` instance. The notification does not contain a *userInfo* dictionary.

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSWorkspace.h

NSWorkspaceSessionDidResignActiveNotification

Posted before a user session is switched out. This allows an application to disable some processing when its user session is switched out, and re-enable when that session gets switched back in, for example.

The notification object is the shared `NSWorkspace` instance. The notification does not contain a *userInfo* dictionary.

If an application is launched in an inactive session, `NSWorkspaceSessionDidResignActiveNotification` is sent after `NSApplicationWillFinishLaunchingNotification` and before sending `NSApplicationDidFinishLaunchingNotification`.

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSWorkspace.h

NSWorkspaceDidHideApplicationNotification

Posted when the Finder is about to hide an application.

The notification object is the shared `NSWorkspace` instance. In Mac OS X v10.6 and later the *userInfo* dictionary contains the `NSWorkspaceApplicationKey` (page 48) key with a corresponding instance of `NSRunningApplication` that represents the affected application.

Availability

Available in Mac OS X v10.6 and later.

Declared In

NSWorkspace.h

NSWorkspaceDidUnhideApplicationNotification

Posted when the Finder is about to unhide an application.

The notification object is the shared `NSWorkspace` instance. In Mac OS X v10.6 and later the `userInfo` dictionary contains the `NSWorkspaceApplicationKey` (page 48) key with a corresponding instance of `NSRunningApplication` that represents the affected application.

Availability

Available in Mac OS X v10.6 and later.

Declared In

`NSWorkspace.h`

NSWorkspaceDidActivateApplicationNotification

Posted when the Finder is about to activate an application.

The notification object is the shared `NSWorkspace` instance. In Mac OS X v10.6 and later the `userInfo` dictionary contains the `NSWorkspaceApplicationKey` (page 48) key with a corresponding instance of `NSRunningApplication` that represents the affected application.

Availability

Available in Mac OS X v10.6 and later.

Declared In

`NSWorkspace.h`

NSWorkspaceDidDeactivateApplicationNotification

Posted when the Finder is about to deactivate an application.

The notification object is the shared `NSWorkspace` instance. In Mac OS X v10.6 and later the `userInfo` dictionary contains the `NSWorkspaceApplicationKey` (page 48) key with a corresponding instance of `NSRunningApplication` that represents the affected application.

Availability

Available in Mac OS X v10.6 and later.

Declared In

`NSWorkspace.h`

NSWorkspaceDidRenameVolumeNotification

Posted when a volume changes its name and/or mount path. These typically change simultaneously, in which case only one notification is posted.

The notification object is the shared `NSWorkspace` instance. The `userInfo` dictionary contains keys in [“NSWorkspaceDidRenameVolumeNotification User Info Keys”](#) (page 47) and [“Volume Mounting Notification User Info Keys”](#) (page 47).

Availability

Available in Mac OS X v10.6 and later.

Declared In

`NSWorkspace.h`

NSWorkspaceDidMountNotification

Posted when a new device has been mounted.

The notification object is the shared `NSWorkspace` instance.

In Mac OS X v10.5 and earlier the *userInfo* dictionary contains a key `@NSDevicePath` that returns the path where the device was mounted, as a string.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSWorkspace.h`

NSWorkspaceWillUnmountNotification

Posted when the Finder is about to unmount a device.

This notification will not be delivered if a volume was forcibly and immediately made unavailable, such as when a FireWire drive is simply unplugged, because there is no chance to deliver it before the volume becomes unavailable.

The notification object is the shared `NSWorkspace` instance. The *userInfo* dictionary contains a key `@NSDevicePath` that returns the path where the device was mounted, as a string.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSWorkspace.h`

NSWorkspaceDidUnmountNotification

Posted when the Finder did unmount a device.

This notification will not be delivered if a volume was forcibly and immediately made unavailable, such as when a FireWire drive is simply unplugged, because there is no chance to deliver it before the volume becomes unavailable.

The notification object is the shared `NSWorkspace` instance. The *userInfo* dictionary contains a key `@NSDevicePath` that returns the path where the device was mounted, as a string.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSWorkspace.h`

NSWorkspaceDidPerformFileOperationNotification

Posted when a file operation has been performed in the receiving application.

The notification object is the shared `NSWorkspace` instance. The *userInfo* dictionary contains a key `@NSOperationNumber` with a `NSNumber` object containing an integer indicating the type of file operation

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSWorkspace.h

NSWorkspaceDidChangeFileLabelsNotification

Posted when the Finder file labels or colors change.

The notification object is the shared `NSWorkspace` instance. The notification does not contain a *userInfo* dictionary.

Availability

Available in Mac OS X v10.6 and later.

Declared In

NSWorkspace.h

NSWorkspaceActiveSpaceDidChangeNotification

Posted when a Spaces change has occurred.

The notification object is the shared `NSWorkspace` instance. The notification does not contain a *userInfo* dictionary.

Availability

Available in Mac OS X v10.6 and later.

Declared In

NSWorkspace.h

NSWorkspaceDidWakeNotification

Posted when the machine wakes from sleep.

The notification object is the shared `NSWorkspace` instance. The notification does not contain a *userInfo* dictionary.

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSWorkspace.h

NSWorkspaceWillPowerOffNotification

Posted when the user has requested a logout or that the machine be powered off.

The notification object is the shared `NSWorkspace` instance. This notification does not contain a *userInfo* dictionary.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSWorkspace.h

NSWorkspaceWillSleepNotification

Posted before the machine goes to sleep. An observer of this message can delay sleep for up to 30 seconds while handling this notification.

The notification object is the shared `NSWorkspace` instance. The notification does not contain a *userInfo* dictionary.

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSWorkspace.h

NSWorkspaceScreensDidSleepNotification

Posted when the machine's screen goes to sleep.

The notification object is the shared `NSWorkspace` instance. The notification does not contain a *userInfo* dictionary.

Few applications are likely to be interested in this notification, but they may be useful for certain hardware-based drawing decisions, for example when using OpenGL.

Availability

Available in Mac OS X v10.6 and later.

Declared In

NSWorkspace.h

NSWorkspaceScreensDidWakeNotification

Posted when the machine's screens wake.

The notification object is the shared `NSWorkspace` instance. The notification does not contain a *userInfo* dictionary.

Few applications are likely to be interested in this notification, but they may be useful for certain hardware-based drawing decisions, for example when using OpenGL.

Availability

Available in Mac OS X v10.6 and later.

Declared In

NSWorkspace.h

Deprecated NSWorkspace Methods

A method identified as deprecated has been superseded and may become unsupported in the future.

Deprecated in Mac OS X v10.6

checkForRemovableMedia

Polls the system's drives for any disks that have been inserted but not yet mounted. (Deprecated in Mac OS X v10.6.)

- (void)checkForRemovableMedia

Discussion

This method doesn't wait until such disks are mounted; instead, it requests that the disk be mounted asynchronously and returns immediately. Currently has no effect.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

See Also

- [mountNewRemovableMedia](#) (page 56)
- [mountedRemovableMedia](#) (page 27)

Declared In

NSWorkspace.h

fileSystemChanged

Returns a Boolean value indicating whether a change to the file system has been registered with a [noteFileSystemChanged](#) (page 56) message since the last [fileSystemChanged](#) (page 55) message. (Deprecated in Mac OS X v10.6.)

- (BOOL)fileSystemChanged

Return Value

Currently, this method always returns NO.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

Declared In

NSWorkspace.h

findApplications

Examines all applications and updates the records of registered services and file types. (Deprecated in Mac OS X v10.6.)

- (void)findApplications

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

Declared In

NSWorkspace.h

mountNewRemovableMedia

Returns the full pathnames of any newly mounted disks. (Deprecated in Mac OS X v10.6.)

- (NSArray *)mountNewRemovableMedia

Return Value

An array of NSString objects, each of which contains the full pathname to a newly mounted disk.

Discussion

This method polls the system's drives for any disks that have been inserted but not yet mounted and waits until the new disks have been mounted. This method posts an [NSWorkspaceDidMountNotification](#) (page 52) to the NSWorkspace object's notification center when it is finished. Currently provides the same functionality as [mountedRemovableMedia](#) (page 27).

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

See Also

- [checkForRemovableMedia](#) (page 55)

- [mountedRemovableMedia](#) (page 27)

Declared In

NSWorkspace.h

noteFileSystemChanged

Informs the NSWorkspace object that the file system has changed. (Deprecated in Mac OS X v10.6.)

- (void)noteFileSystemChanged

Deprecated NSWorkspace Methods

Discussion

The `NSWorkspace` object then gets the status of all the files and directories it is interested in and updates itself appropriately. This method is used by many objects that write or delete files.

The `NSDocument` and `NSSavePanel` objects use this method when saving a file. If you create a file directly, you should call [noteFileSystemChanged](#) (page 56) so that the Finder can update the folder if it is open.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

See Also

- [fileSystemChanged](#) (page 55)

Declared In

`NSWorkspace.h`

noteUserDefaultsChanged

Informs the `NSWorkspace` object that the defaults database has changed. (Deprecated in Mac OS X v10.6.)

- (void)noteUserDefaultsChanged

Discussion

The `NSWorkspace` object then reads all the defaults it is interested in and reconfigures itself appropriately. For example, this method is used by the Preferences application to notify the Finder whether the user prefers to see hidden files. Currently has no effect.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

See Also

- [userDefaultsChanged](#) (page 58)

Declared In

`NSWorkspace.h`

openTempFile:

Opens the specified temporary file using the default application for its type. (Deprecated in Mac OS X v10.6.)

- (BOOL)openTempFile:(NSString *)fullPath

Parameters

fullPath

The full path to the temporary file.

Return Value

YES if the file was successfully opened; otherwise, NO.

Deprecated NSWorkspace Methods

Discussion

The sending application is deactivated before the request is sent. Using this method instead of one of the `openFile:...` methods lets the receiving application know that it should delete the file when it no longer needs it. Currently provides the same functionality as `openFile:` (page 29).

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

See Also

- [openFile:](#) (page 29)
- [openFile:fromImage:at:inView:](#) (page 30)
- [openFile:withApplication:](#) (page 30)
- [openFile:withApplication:andDeactivate:](#) (page 31)

Declared In

NSWorkspace.h

slideImage:from:to:

Animates a sliding image from one point to another. (Deprecated in Mac OS X v10.6.)

```
- (void)slideImage:(NSImage *)image from:(NSPoint)fromPoint to:(NSPoint)toPoint
```

Parameters

image

The image to animate.

fromPoint

The starting point, in screen coordinates.

toPoint

The ending point, in screen coordinates.

Discussion

Currently unimplemented.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

Declared In

NSWorkspace.h

userDefaultsChanged

Returns a Boolean value indicating whether a change to the defaults database has been registered with a [noteUserDefaultsChanged](#) (page 57) message since the last [userDefaultsChanged](#) (page 58) message. (Deprecated in Mac OS X v10.6.)

```
- (BOOL)userDefaultsChanged
```

Deprecated NSWorkspace Methods

Return Value

Currently, this method always returns NO.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

Declared In

NSWorkspace.h

Document Revision History

This table describes the changes to *NSWorkspace Class Reference*.

Date	Notes
2009-11-17	Added suggestion to use <code>NSRunningApplications</code> rather than <code>activeApplication</code> and <code>launchedApplications</code> .
2009-07-12	Added Declared In path.
2009-05-27	Updated for Mac OS X v10.6.
2007-04-27	Added Uniform Type Identifier methods.
2007-04-03	Clarified return value of <code>absolutePathForAppBundleWithIdentifier:</code> when the bundle does not exist.
2006-05-23	Added the return type of <code>launchAppWithBundleIdentifier:options:additionalEventParamDescriptor:launchIdentifier:</code> .
	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

absolutePathForAppBundleWithIdentifier: **instance method 15**
activateFileViewerSelectingURLs: **instance method 15**
activeApplication **instance method 15**

C

checkForRemovableMedia **instance method 55**

D

Desktop Image Dictionary Keys **45**
desktopImageOptionsForScreen: **instance method 16**
desktopImageURLForScreen: **instance method 16**
duplicateURLs:completionHandler: **instance method 17**

E

extendPowerOffBy: **instance method 18**

F

File Operations **43**
File Types **42**
fileLabelColors **instance method 18**
fileLabels **instance method 19**
filenameExtension:isValidForType: **instance method 19**
fileSystemChanged **instance method 55**
findApplications **instance method 56**

fullPathForApplication: **instance method 19**

G

getFileSystemInfoForPath:isRemovable:isWritable:isUnmountable:description:type: **instance method 20**
getInfoForFile:application:type: **instance method 20**

H

hideOtherApplications **instance method 21**

I

iconForFile: **instance method 21**
iconForFiles: **instance method 22**
iconForFileType: **instance method 22**
isFilePackageAtPath: **instance method 23**

L

launchApplicationAtURL:options:configuration:error: **instance method 25**
launchApplication: **instance method 23**
launchApplication:showIcon:autoLaunch: **instance method 24**
launchAppWithBundleIdentifier:options:additionalEventParamDescriptor:launchIdentifier: **instance method 25**
launchedApplications **instance method 26**
localizedDescriptionForType: **instance method 27**

M

mountedLocalVolumePaths **instance method** 27
 mountedRemovableMedia **instance method** 27
 mountNewRemovableMedia **instance method** 56

N

noteFileSystemChanged **instance method** 56
 noteFileSystemChanged: **instance method** 28
 noteUserDefaultsChanged **instance method** 57
 notificationCenter **instance method** 29
 NSApplicationFileType **constant (Deprecated in Mac OS X v10.6)** 42
 NSDirectoryFileType **constant (Deprecated in Mac OS X v10.6)** 42
 NSExclude10_4ElementsIconCreationOption **constant** 48
 NSExcludeQuickDrawElementsIconCreationOption **constant** 48
 NSFilesystemFileType **constant (Deprecated in Mac OS X v10.6)** 42
 NSPlainFileType **constant (Deprecated in Mac OS X v10.6)** 42
 NSShellCommandFileType **constant (Deprecated in Mac OS X v10.6)** 42
 NSWorkspaceActiveSpaceDidChangeNotification **notification** 53
 NSWorkspaceApplicationKey **constant** 48
 NSWorkspaceApplicationKey User Info Key **constant** 48
 NSWorkspaceCompressOperation **constant** 44
 NSWorkspaceCopyOperation **constant** 44
 NSWorkspaceDecompressOperation **constant** 44
 NSWorkspaceDecryptOperation **constant** 44
 NSWorkspaceDesktopImageAllowClippingKey **constant** 45
 NSWorkspaceDesktopImageFillColorKey **constant** 45
 NSWorkspaceDesktopImageScalingKey **constant** 45
 NSWorkspaceDestroyOperation **constant** 44
 NSWorkspaceDidActivateApplicationNotification **notification** 51
 NSWorkspaceDidChangeFileLabelsNotification **notification** 53
 NSWorkspaceDidDeactivateApplicationNotification **notification** 51
 NSWorkspaceDidHideApplicationNotification **notification** 50
 NSWorkspaceDidLaunchApplicationNotification **notification** 49
 NSWorkspaceDidMountNotification **notification** 52

NSWorkspaceDidPerformFileOperationNotification **notification** 52
 NSWorkspaceDidRenameVolumeNotification **notification** 51
 NSWorkspaceDidRenameVolumeNotification User Info Keys 47
 NSWorkspaceDidTerminateApplicationNotification **notification** 49
 NSWorkspaceDidUnhideApplicationNotification **notification** 50
 NSWorkspaceDidUnmountNotification **notification** 52
 NSWorkspaceDidWakeNotification **notification** 53
 NSWorkspaceDuplicateOperation **constant** 44
 NSWorkspaceEncryptOperation **constant** 44
 NSWorkspaceLaunchAllowingClassicStartup **constant** 46
 NSWorkspaceLaunchAndHide **constant** 47
 NSWorkspaceLaunchAndHideOthers **constant** 47
 NSWorkspaceLaunchAndPrint **constant** 46
 NSWorkspaceLaunchAsync **constant** 46
 NSWorkspaceLaunchConfigurationAppleEvent **constant** 43
 NSWorkspaceLaunchConfigurationArchitecture **constant** 43
 NSWorkspaceLaunchConfigurationArguments **constant** 43
 NSWorkspaceLaunchConfigurationEnvironment **constant** 43
 NSWorkspaceLaunchDefault **constant** 47
 NSWorkspaceLaunchInhibitingBackgroundOnly **constant** 46
 NSWorkspaceLaunchNewInstance **constant** 47
 NSWorkspaceLaunchOptions 45
 NSWorkspaceLaunchPreferringClassic **constant** 46
 NSWorkspaceLaunchWithoutActivation **constant** 46
 NSWorkspaceLaunchWithoutAddingToRecents **constant** 46
 NSWorkspaceLinkOperation **constant** 44
 NSWorkspaceMoveOperation **constant** 43
 NSWorkspaceRecycleOperation **constant** 44
 NSWorkspaceScreensDidSleepNotification **notification** 54
 NSWorkspaceScreensDidWakeNotification **notification** 54
 NSWorkspaceSessionDidBecomeActiveNotification **notification** 50
 NSWorkspaceSessionDidResignActiveNotification **notification** 50
 NSWorkspaceVolumeLocalizedNameKey **constant** 47
 NSWorkspaceVolumeOldLocalizedNameKey **constant** 48
 NSWorkspaceVolumeOldURLKey **constant** 48

NSWorkspaceVolumeURLKey **constant** [47](#)
 NSWorkspaceWillLaunchApplicationNotification **notification** [49](#)
 NSWorkspaceWillPowerOffNotification **notification** [53](#)
 NSWorkspaceWillSleepNotification **notification** [54](#)
 NSWorkspaceWillUnmountNotification **notification** [52](#)

O

openFile: **instance method** [29](#)
 openFile:fromImage:at:inView: **instance method** [30](#)
 openFile:withApplication: **instance method** [30](#)
 openFile:withApplication:andDeactivate: **instance method** [31](#)
 openTempFile: **instance method** [57](#)
 openURL: **instance method** [32](#)
 openURLs:withAppBundleIdentifier:options:additionalEventParamDescriptor:launchIdentifiers: **instance method** [32](#)

P

performFileOperation:source:destination:files:tag: **instance method** [33](#)
 preferredFilenameExtensionForType: **instance method** [34](#)

R

recycleURLs:completionHandler: **instance method** [34](#)
 runningApplications **instance method** [35](#)

S

selectFile:inFileViewerRootedAtPath: **instance method** [36](#)
 setDesktopImageURL:forScreen:options:error: **instance method** [36](#)
 setIcon:forFile:options: **instance method** [37](#)
 sharedWorkspace **class method** [14](#)
 showSearchResultsForQueryString: **instance method** [38](#)
 slideImage:from:to: **instance method** [58](#)

T

type:conformsToType: **instance method** [38](#)
 typeOfFile:error: **instance method** [39](#)

U

unmountAndEjectDeviceAtPath: **instance method** [39](#)
 unmountAndEjectDeviceAtURL:error: **instance method** [40](#)
 URLForApplicationToOpenURL: **instance method** [40](#)
 URLForApplicationWithBundleIdentifier: **instance method** [41](#)
 userDefaultsChanged **instance method** [58](#)

V

Volume Mounting Notification User Info Keys [47](#)

W

Workspace icon creation options [48](#)
 Workspace Launch Configuration Options [42](#)