API Reference

AppKit › NSWorkspace

Class

NSWorkspace

An NSWorkspace object can launch other apps and perform a variety of file-handling services.

Language

Swift | Objective-C

SDK

macOS 10.0+

On This Page

Overview

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

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

You can use the workspace object to:

- Open, manipulate, and get information about files and devices
- Track changes to the file system, devices, and the user database
- Get and set Finder information for files
- Launch apps

Legacy Constants

Table 1 describes keys for an NSDictionary object containing information about an app. This dictionary is returned by activeApplication and launchedApplications, and is also provided in the userInfo of NSWorkspace notifications for app launch and termination.

Note that these constants are considered legacy.

> Note
>
> It is strongly suggested that you use the NSWorkspace class's runningApplications method and the NSRunningApplication class to retrieve this information in apps target macOS 10.6 and later, rather than the activeApplication and launchedApplications methods.

Table 1 userInfo dictionary keys for activeApplication and launchedApplications and notifications for app launch and termination

Key	Value
@"NSApplicationPath"	The full path to the app, as a NSString object.
@"NSApplicationName"	The app's name, as an NSString object.

`@"NSApplicationBundleIdentifier"`	The app’s bundle identifier, as an `NSString` object.
`@"NSApplicationProcessIdentifier"`	The app's process id, as an `NSNumber` object.
`@"NSApplicationProcessSerialNumberHigh"`	The high long of the process serial number (PSN), as an `NSNumber` object.
`@"NSApplicationProcessSerialNumberLow"`	The low long of the process serial number (PSN), as an `NSNumber` object.

Symbols

Accessing the Shared NSWorkspace Instance

`+ sharedWorkspace`

Returns the shared `NSWorkspace` instance.

Accessing the NSWorkspace Notification Center

`notificationCenter`

Returns the notification center for workspace notifications.

Opening Files

`– openURL:`

Opens the location at the specified URL.

`– openFile:`

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

`– openFile:withApplication:`

Opens a file using the specified app.

`– openFile:withApplication:andDeactivate:`

Opens the specified file and optionally deactivates the sending app.

~~`– openFile:fromImage:at:inView:`~~

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

Deprecated

Manipulating Applications

`– launchApplication:`

Launches the specified app.

`– launchApplication:showIcon:autolaunch:`

Launches the specified app using additional options.

`– launchApplicationAtURL:options:configuration:error:`

Launches the app at the specified URL.

– `hideOtherApplications`
    Hides all applications other than the sender.

Manipulating Files

– `duplicateURLs:completionHandler:`
    Duplicates the specified URLS asynchronously in the same manner as the Finder.

– `recycleURLs:completionHandler:`
    Moves the specified URLs to the trash in the same manner as the Finder.

~~– `performFileOperation:source:destination:files:tag:`~~
    Performs a file operation on a set of files in a particular directory.
    Deprecated

– `activateFileViewerSelectingURLs:`
    Activates the Finder, and opens one or more windows selecting the specified files.

– `selectFile:inFileViewerRootedAtPath:`
    Selects the file specified by `fullPath`.

Manipulating Uniform Type Identifier Information

– `typeOfFile:error:`
    Returns the uniform type identifier of the specified file, if it can be determined..

– `localizedDescriptionForType:`
    Returns the localized description for the specified Uniform Type Identifier

– `preferredFilenameExtensionForType:`
    Returns the preferred filename extension for the specified Uniform Type Identifier.

– `filenameExtension:isValidForType:`
    Returns whether the specified filename extension is appropriate for the Uniform Type Identifier.

– `type:conformsToType:`
    Returns a Boolean indicating that the first Uniform Type Identifier conforms to the second Uniform Type Identifier.

– `URLForApplicationWithBundleIdentifier:`
    Returns the URL for the app with the specified identifier.

Requesting Information

– `getInfoForFile:application:type:`
    Retrieves information about the specified file.

– `URLForApplicationToOpenURL:`
    Returns the URL to the default app that would be used to open the given URL.

– `fullPathForApplication:`
    Returns the full path for the specified app.

–
`getFileSystemInfoForPath:isRemovable:isWritable:isUnmountable:description:type:`
    Describes the file system at `fullPath`.

– `isFilePackageAtPath:`
    Determines whether the specified path is a file package.

`frontmostApplication`
Returns the frontmost app, which is the app that receives key events.

`runningApplications`
Returns an array of `NSRunningApplication` representing the running apps.

`menuBarOwningApplication`
Returns the app that owns the currently displayed menu bar.

Managing Icons

– `iconForFile:`
Returns an image containing the icon for the specified file.

– `iconForFileType:`
Returns an image containing the icon for files of the specified type.

– `iconForFiles:`
Returns an image containing the icon for the specified files.

– `setIcon:forFile:options:`
Sets the icon for the file or directory at the specified path.

Unmounting a Device

– `unmountAndEjectDeviceAtPath:`
Unmounts and ejects the device at the specified path.

– `unmountAndEjectDeviceAtURL:error:`
Attempts to eject the volume mounted at the given path.

Working with Bundles

– `absolutePathForAppBundleWithIdentifier:`
Returns the absolute file-system path of an app bundle.

–
`launchAppWithBundleIdentifier:options:additionalEventParamDescriptor:launchIdentifier:`
Launches the app corresponding to the specified `bundleIdentifier`.

–
`openURLs:withAppBundleIdentifier:options:additionalEventParamDescriptor:launchIdentifiers:`
Opens one or more files from an array of URLs.

Managing the Desktop Image

– `desktopImageURLForScreen:`
Returns the URL for the desktop image for the given screen.

– `setDesktopImageURL:forScreen:options:error:`
Sets the desktop image for the given screen to the image at the specified URL.

– `desktopImageOptionsForScreen:`
Returns the desktop image options for the given screen.

Performing Finder Spotlight Searches

– `showSearchResultsForQueryString:`
Displays a Spotlight search results window in Finder for the specified query string.

Finder File Labels

`fileLabelColors`

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

`fileLabels`

Returns the array of file labels as strings.

Tracking Changes to the File System

– `noteFileSystemChanged:`

Informs the workspace object that the file system changed at the specified path.

Requesting Additional Time Before Logout

– `extendPowerOffBy:`

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

Supporting Accessibility

`accessibilityDisplayShouldDifferentiateWithoutColor`

A Boolean value that indicates whether the app should avoid conveying information through color alone.

`accessibilityDisplayShouldIncreaseContrast`

A Boolean value that indicates whether the app should present a high contrast user interface.

`accessibilityDisplayShouldReduceTransparency`

A Boolean value that indicates whether the app should avoid using semitransparent backgrounds.

Deprecated Methods

~~– `openTempFile:`~~

Opens the specified temporary file using the default app for its type.

Deprecated

~~– `findApplications`~~

Examines all apps and updates the records of registered services and file types.

Deprecated

~~– `noteUserDefaultsChanged`~~

Informs the `NSWorkspace` object that the defaults database has changed.

Deprecated

~~– `slideImage:from:to:`~~

Animates a sliding image from one point to another.

Deprecated

~~– `checkForRemovableMedia`~~

Polls the system's drives for any disks that have been inserted but not yet mounted.

Deprecated

~~– `noteFileSystemChanged`~~

Informs the `NSWorkspace` object that the file system has changed.

Deprecated

~~– `fileSystemChanged`~~

Returns a Boolean value indicating whether a change to the file system has been registered with a `noteFileSystemChanged` message since the last `fileSystemChanged` message.

Deprecated

~~– userDefaultsChanged~~

Returns a Boolean value indicating whether a change to the defaults database has been registered with a `noteUserDefaultsChanged` message since the last `userDefaultsChanged` message.

Deprecated

~~– mountNewRemovableMedia~~

Returns the full pathnames of any newly mounted disks.

Deprecated

~~– mountedRemovableMedia~~

Returns the full pathnames of all currently mounted removable disks.

Deprecated

~~– mountedLocalVolumePaths~~

Returns the mount points of all local volumes, not just the removable ones returned by `mountedRemovableMedia`.

Deprecated

~~– activeApplication~~

Returns a dictionary with information about the current active app.

Deprecated

~~– launchedApplications~~

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

Deprecated

Constants

File Types

These constants specify different types of files returned by the `getInfoForFile:application:type:` method.

Workspace Launch Configuration Options

The following keys can be used in the configuration dictionary of the `launchApplicationAtURL:options:configuration:error:` method. Each key is optional, and if omitted, default behavior is applied.

File Operations

These constants specify different types of file operations used by `performFileOperation:source:destination:files:tag:`.

Desktop Image Dictionary Keys

The following keys may be specified or returned in the options dictionary for `setDesktopImageURL:forScreen:options:error:`.

`NSWorkspaceLaunchOptions`

These constants define launch options you can pass to `launchAppWithBundleIdentifier:options:additionalEventParamDescriptor:launchIdentifier:` and `openURLs:withAppBundleIdentifier:options:additionalEventParamDescriptor:launchIdentifiers:`.

Volume Mounting Notification User Info Keys

The following keys are available in the `userInfo` parameter of the notification named `NSWorkspaceDidRenameVolumeNotification`.

NSWorkspaceDidRenameVolumeNotification User Info Keys

The following keys are available in the `userInfo` parameter of the notification named `NSWorkspaceDidRenameVolumeNotification`.

NSWorkspaceApplicationKey User Info Key

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

`NSWorkspaceIconCreationOptions`

These constants describe the `NSWorkspaceIconCreationOptions` values used by `setIcon:forFile:options:`. You can combine these using the C bitwise OR operator.

Notifications

All `NSWorkspace` notifications are posted to the `NSWorkspace` object's own notification center, not the app's default notification center. Access this center using the `NSWorkspace` object's `notificationCenter` method.

`NSWorkspaceWillLaunchApplicationNotification`

Posted when the Finder is about to launch an app.

`NSWorkspaceDidLaunchApplicationNotification`

Posted when a new app has started up.

`NSWorkspaceDidTerminateApplicationNotification`

Posted when an app finishes executing.

`NSWorkspaceSessionDidBecomeActiveNotification`

Posted after a user session is switched in.

`NSWorkspaceSessionDidResignActiveNotification`

Posted before a user session is switched out.

`NSWorkspaceDidHideApplicationNotification`

Posted when the Finder hid an app.

`NSWorkspaceDidUnhideApplicationNotification`

Posted when the Finder unhid an app.

`NSWorkspaceDidActivateApplicationNotification`

Posted when the Finder is about to activate an app.

`NSWorkspaceDidDeactivateApplicationNotification`

Posted when the Finder deactivated an app.

`NSWorkspaceDidRenameVolumeNotification`

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

`NSWorkspaceDidMountNotification`

Posted when a new device has been mounted.

`NSWorkspaceWillUnmountNotification`

Posted when the Finder is about to unmount a device.

`NSWorkspaceDidUnmountNotification`

Posted when the Finder did unmount a device.

~~`NSWorkspaceDidPerformFileOperationNotification`~~

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

Deprecated

`NSWorkspaceDidChangeFileLabelsNotification`
Posted when the Finder file labels or colors change.

`NSWorkspaceActiveSpaceDidChangeNotification`
Posted when a Spaces change has occurred.

`NSWorkspaceDidWakeNotification`
Posted when the machine wakes from sleep.

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

`NSWorkspaceWillSleepNotification`
Posted before the machine goes to sleep.

`NSWorkspaceScreensDidSleepNotification`
Posted when the machine's screen goes to sleep.

`NSWorkspaceScreensDidWakeNotification`
Posted when the machine's screens wake.

`NSWorkspaceAccessibilityDisplayOptionsDidChangeNotification`
Posted when any of the accessibility display options change.

Instance Properties

`accessibilityDisplayShouldInvertColors`

`accessibilityDisplayShouldReduceMotion`

Instance Methods

`– openURL:options:configuration:error:`

`– openURLs:withApplicationAtURL:options:configuration:error:`

Relationships

Inherits From

`NSObject`