

---

# NSOpenPanel

**Inherits From:** NSSavePanel : NSObject

**Conforms To:** NSObject (NSObject)

**Declared In:** AppKit/NSOpenPanel.h

**Note:** The inheritance and conformance information shown above applies only to NSOpenPanel on OpenStep for Windows. On Mach, NSOpenPanel inherits from (in this order) NSSavePanel, NSPanel, NSWindow, NSResponder, and NSObject.

## Class Description

NSOpenPanel provides the Open dialog on OpenStep for Windows or the Open panel on the OpenStep for Mach user interface. Applications use the Open panel (or dialog) as a convenient way to query the user for the name of a file to open. The Open panel can only be run modally.

Most of this class's behavior is defined by its superclass, NSSavePanel. NSOpenPanel adds to this behavior by:

- Letting you specify the types (by file-name extension) of the items that will appear in the panel
- Letting the user select files, directories, or both
- Letting the user select multiple items at a time

Typically, you access an NSOpenPanel by invoking the **openPanel** method. When the class receives an **openPanel** message, it tries to reuse an existing panel rather than create a new one. If a panel is reused, its attributes are reset to the default values so that the effect is the same as receiving a new panel. Because Open dialogs and panels may be reused, you shouldn't modify the instance returned by **openPanel** except through the methods listed below (and through those inherited from NSSavePanel). For example, you can set the panel's title and whether it allows multiple selection, but not the arrangement of the buttons within the panel. If you must modify the Open panel substantially, create and manage your own instance using the **alloc...** and **init...** methods rather than the **openPanel** method.

The following code example shows the NSOpenPanel displaying only files with extensions of ".td" and allowing multiple selection. If the user makes a selection and clicks the OK button (that is, **runModalForDirectory:file:types:** returns NSOKButton), this method opens each selected file:

```
- (void)openDoc:(id)sender
{
    int result;
    NSArray *fileTypes = [NSArray arrayWithObject:@"td"];
    NSOpenPanel *oPanel = [NSOpenPanel openPanel];
```

```
[oPanel setAllowsMultipleSelection:YES];
result = [oPanel runModalForDirectory:NSHomeDirectory() file:nil
           types:fileTypes];
if (result == NSOKButton) {
    NSArray *filesToOpen = [oPanel filenames];
    int i, count = [filesToOpen count];
    for (i=0; i<count; i++) {
        NSString *aFile = [filesToOpen objectAtIndex:i];
        id currentDoc = [[ToDoDoc alloc] initWithFile:aFile];
    }
}
```

### Method Types

Obtaining the shared instance	+ openPanel
Running the panel modally	– runModalForDirectory:file:types: – runModalForTypes:
Getting the user selection	– filenames
Allowing browser selections	– setCanChooseFiles: – canChooseFiles – setCanChooseDirectories: – canChooseDirectories
Allowing multiple selections	– setAllowsMultipleSelection: – allowsMultipleSelections

### Class Methods

#### openPanel

+ (NSOpenPanel \*)**openPanel**

Returns a "recycled" NSOpenPanel or, if one doesn't yet exist, creates it before returning it. New and recycled NSOpenPanels are reset to default values, which include selection of single files only.

---

## Instance Methods

### **allowsMultipleSelection**

– (BOOL)**allowsMultipleSelection**

Returns whether the NSOpenPanel’s browser allows the user to open multiple files (and directories) at a time. If multiple files or directories are allowed, then the **filename** method—inherited from NSSavePanel—returns a non-**nil** value only if one and only one file is selected. By contrast, NSOpenPanel’s **filenames** method always returns the selected files, even if only one file is selected.

**See also:** – **filename**(NSSavePanel), – **filenames**, – **setAllowsMultipleSelection:**

### **canChooseDirectories**

– (BOOL)**canChooseDirectories**

Returns whether the Open dialog or panel allows the user to choose directories to open.

**See also:** – **setCanChooseDirectories:**

### **canChooseFiles**

– (BOOL)**canChooseFiles**

Returns whether the Open dialog or panel allows the user to choose files to open.

**See also:** – **setCanChooseFiles:**

### **filenames**

– (NSArray \*)**filenames**

Returns an array containing the absolute paths (as NSString objects) of the selected files and directories. If multiple selections aren’t allowed, the array contains a single name. The **filenames** method is preferable over NSSavePanel’s **filename** to get the name or names of files and directories that the user has selected.

### **runModalForDirectory:file:types:**

– (int)**runModalForDirectory:**(NSString \*)*path*  
    **file:**(NSString \*)*name*  
    **types:**(NSArray \*)*fileTypes*

Displays the NSOpenPanel and begins its event loop. The NSOpenPanel displays the files in *path* that match the types in *fileTypes* (an NSArray of file extensions as NSString objects). If *path* is **nil** the directory

the application is "currently in" is substituted (on Mach, this is the user's home directory); in later invocations of the shared NSOpenPanel, the directory is by default the last directory where a file was opened, unless specified otherwise with **setDirectory:**. If all files and subdirectories should be included in the next column of a browser, *fileTypes* should be **nil**. If *name* specifies a particular file, it is selected when the Open dialog or panel is presented to the user; otherwise, *name* should be **nil**. This method returns NSOKButton (if the user clicks the OK button) or NSCancelButton (if the user clicks the Cancel button).

**See also:** – **runModalForTypes:**

### **runModalForTypes:**

– (int)**runModalForTypes:**(NSArray \*)*fileTypes*

Invokes the **runModalForDirectory:file:types:** method, using **nil** for file and directory (see description of **runModalForDirectory:file:types:**). The *fileTypes* argument is an NSArray containing the extensions of files to be shown in the browser. Returns NSOKButton (if the user clicks the OK button) or NSCancelButton (if the user clicks the Cancel button).

### **setAllowsMultipleSelection:**

– (void)**setAllowsMultipleSelection:**(BOOL)*flag*

Sets whether the user can select multiple files (and directories) at one time for opening.

**See also:** – **allowsMultipleSelection**

### **setCanChooseDirectories:**

– (void)**setCanChooseDirectories:**(BOOL)*flag*

Sets whether the user can select directories in the NSOpenPanel's browser. When a directory is selected, the OK button is enabled only if *flag* is YES.

**See also:** – **canChooseDirectories**

### **setCanChooseFiles:**

– (void)**setCanChooseFiles:**(BOOL)*flag*

Sets whether the user can select files in the NSOpenPanel's browser or type the files to be accepted.

**See also:** – **canChooseFiles**