| | |
|---|---|
| id)finishLaunching | Activates the application, opens any files specified by the ªNSOpenº user default, and unhighlights the application's icon in the Workspace Manager. This method is invoked by run before it starts the event loop. When this method begins, it posts the notification NSApplicationWillFinishLaunchingNotification with the receiving object to the default notification center. When it successfully completes, it posts the notification NSApplicationDidFinishLaunchingNotification. If you override finishLaunching, the subclass method should invoke the superclass method. |
| id)activateIgnoringOtherApps:(BOOL)flag | Makes this the active application. If flag is NO, the application is activated only if no other application is currently active. |
| id)deactivate | Deactivates the application. |
| OOL)isActive | Returns whether this is the active application. |
| id)abortModal | Aborts the event loop started by runModalForWindow:. |
| SModalSession)beginModalSessionForWindow:(NSWindow *)theWindow | Sets up a modal session with theWindow. |
| id)endModalSession:(NSModalSession)session | |

| | |
|---|---|
| id)sendEvent:(NSEvent *)theEvent | Dispatches events to other objects. When sending the activ this method posts the notifications NSApplicationWill NSApplicationDidBecomeActive with the receiving ob notification center. When sending the deactivate applic NSApplicationWillResignActiveNotification and NSApplicationDidResignActiveNotification notificatio object to the default notification center. |
| id)stop:(id)sender | Stops the main event loop. |
| id)stopModal | Stops the modal event loop. |
| id)stopModalWithCode:(int)returnCode | Stops the event loop started by runModalForWindow: and runModalForWindow: will return. |
| SEvent *)currentEvent | Returns the current event. |
| id)discardEventsMatchingMask:(unsigned int)mask beforeEvent:(NSEvent *)lastEvent | Removes from the event queue all events matching mask t before lastEvent. |
| SEvent *)nextEventMatchingMask:(unsigned int)mask untilDate:(NSDate *)expiration inMode:(NSString *)mode dequeue:(BOOL)flag | Returns the next event matching mask, or nil if no such event is found before the expiration date. If fla is YES, the event is removed from the queue. The mod NSRunLoop mode that determines what other ports are timers may fire while the NSApplication is waiting for |
| id)postEvent:(NSEvent *)event atStart:(BOOL)flag | Adds event to the beginning of the application's event que to the end otherwise. |
| OOL)sendAction:(SEL)aSelector to:(id)aTarget from:(id)sender | Sends an action message to aTarget or up the responder chain. |
| )targetForAction:(SEL)aSelector | Returns the object that receives the action message aSelec |
| OOL)tryToPerform:(SEL)aSelector with:(id)anObject | Attempts to send a message to the application or the delegate. |
| id)setApplicationIconImage:(NSImage *)anImage | Sets the application's icon to anImage. |
| SImage *)applicationIconImage | Returns the NSImage used for the application's icon. |
| id)hide:(id)sender | Hides all the application's windows. When this method be notification NSApplicationWillHideNotification with t the default notification center. When it completes succ notification NSApplicationDidHideNotification. |

| | |
|---|---|
| SWindow *)keyWindow | Returns the key window. |
| SWindow *)mainWindow | Returns the main window. |
| SWindow *)makeWindowsPerform:(SEL)aSelector inOrder:(BOOL)flag | Sends the aSelector message to the application's NSWind order if flag is YES, otherwise in the order of the array method returns. |
| id)miniaturizeAll:(id)sender | Miniaturizes all the receiver's application windows. |
| id)preventWindowOrdering | Suppresses the usual window ordering in handling the mo event. |
| id)setWindowsNeedUpdate:(BOOL)flag | Sets whether the application's windows need updating wh finished processing the current event. This method is e making sure menus are updated to reflect changes not i actions. |
| id)updateWindows | Sends an update message to on-screen NSWindows. Wher sends the notification NSApplicationWillUpdateNotifi receiving object to the default notification center. Whe completes, it sends the notification NSApplicationDidU |
| SArray *)windows | Returns an array of the application's NSWindows. |
| SWindow *)windowWithWindowNumber:(int)windowNum | Returns the NSWindow object corresponding to windowN |
| id)orderFrontColorPanel:(id)sender | Brings up the color panel. |
| id)orderFrontDataLinkPanel:(id)sender | Shows the shared instance of the data link panel, creating |
| id)orderFrontHelpPanel:(id)sender | Shows the application's help panel or the default one. |
| id)runPageLayout:(id)sender | Runs the application's page layout panel. |
| SMenu *)mainMenu | Returns the id of the application's main menu. |
| id)setMainMenu:(NSMenu *)aMenu | Makes aMenu the application's main menu. |
| id)addWindowsItem:(id)aWindow title:(NSString *)aString filename:(BOOL)isFilename | Adds a Windows menu item for aWindow. |
| id)arrangeInFront:(id)sender | Orders all registered NSWindows to the front. |
| id)changeWindowsItem:(id)aWindow title:(NSString *)aString filename:(BOOL)isFilename | Changes the Windows menu item for aWindow. |
| d)removeWindowsItem:(id)aWindow | Removes the Windows menu item for aWindow. |
| id)setWindowsMenu:(id)aMenu | Sets the Windows menu. |

| | |
|---|---|
| id)setServicesMenu:(NSMenu *)aMenu | Sets the Services menu. |
| )validRequestorForSendType:(NSString *)sendType returnType:(NSString *)returnType | Indicates whether the NSApplication can send and receive |
| SDPSContext *)context | Returns the NSApplication's Display PostScript context. |
| id)reportException:(NSException *)anException | Logs the given exception by calling NSLog(). |
| id)terminate:(id)sender | Frees the NSApplication object and exits the application. |
| )delegate | Returns the NSApplication's delegate. |
| id)setDelegate:(id)anObject | Makes anObject the NSApplication's delegate. |
| OOL)application:(id)sender openFileWithoutUI:(NSString *)filename | Sent directly by sender to the delegate. Opens the specifie file to run without a user interface.Work with the file w programmatic control of sender, rather than under keyb user. Returns YES or NO to indicate whether the file w |
| OOL)application:(NSApplication *)application openFile:(NSString *)filename | Sent directly by application to the delegate. Like application:openFileWithoutUI:, but brings up the user application. |
| OOL)application:(NSApplication *)application openTempFile:(NSString *)filename | Sent directly by application to the delegate. Like application:openFile:, but a file opened through this m temporary it's the application's responsibility to remove the file at the ap |
| id)applicationDidBecomeActive:(NSNotification *)aNotification | Sent by the default notification center to the delegate aNot NSApplicationDidBecomeActiveNotification. If the de method, it's automatically registered to receive the not |
| id)applicationDidFinishLaunching:(NSNotification *)aNotification | Sent by the default notification center to the delegate aNot NSApplicationDidFinishLaunchingNotification. If the this method, it's automatically registered to receive the |
| id)applicationDidHide:(NSNotification *)aNotification | Sent by the default notification center to the delegate aNot NSApplicationDidHideNotification. If the delegate im it's automatically registered to receive the notification. |
| id)applicationDidResignActive:(NSNotification *)aNotification | |

NSApplicationDidUpdateNotification. If the delegate i
method, it's automatically registered to receive the not

OOL)applicationOpenUntitledFile:(NSApplication *)application
Sent directly by application to the delegate. Like applicati
a new, untitled document.

OOL)applicationShouldTerminate:(id)sender Sent directly by sender to the delegate. Returns YES if the
terminate.

id)applicationWillBecomeActive:(NSNotification *)aNotification
Sent by the default notification center to the delegate aNot
NSApplicationWillBecomeActiveNotification. If the d
method, it's automatically registered to receive this no

id)applicationWillFinishLaunching:(NSNotification *)aNotification
Sent by the default notification center to the delegate aNot
NSApplicationWillFinishLaunchingNotification. If the
this method, it's automatically registered to receive thi

id)applicationWillHide:(NSNotification *)aNotification
Sent by the default notification center to the delegate aNot
NSApplicationWillHideNotification. If the delegate im
it's automatically registered to receive this notification

id)applicationWillResignActive:(NSNotification *)aNotification
Sent by the default notification center to the delegate aNot
NSApplicationWillResignActiveNotification. If the de
method, it's automatically registered to receive this no

id)applicationWillUnhide:(NSNotification *)aNotification
Sent by the default notification center to the delegate aNot
NSApplicationWillUnhideNotification. If the delegate
method, it's automatically registered to receive the not

id)applicationWillUpdate:(NSNotification *)aNotification
Sent by the default notification center to the delegate aNot
NSApplicationWillUpdateNotification. If the delegate
method, it's automatically registered to receive this no