

iveDocumentReturns the active document

ectionOwnerReturns the editor of the currently selected object

nectSourceReturns the object that's the source of the connection

nectDestinationReturns the object that's the destination of the connection

ool)isConnectingReturns YES if connection lines are being displayed

pConnectingRemoves any connection lines from the screen

playConnectionBetween:sourceCauses Interface Builder to draw connection lines between
and:destination
source and destination

ool)isTestingInterfaceReturns YES if Interface Builder is in Test mode

isterDocumentController:aControllerAdds aController to the list of objects to be notified when documents are opened or saved

registerDocumentController:aControllerRemoves aController from the list of document controllers

destinationImplement to return the object that's the destination of the connection

ablishConnectionImplement to connect the source and destination objects

eImplement to release the storage for the connector object

InstantiateImplement to verify the identities of the connector's source and destination objects

d:(NXTypedStream *)streamImplement to unarchive the connector object from stream

ewObject:oldImplement to update a connector by replacing its old

to:newsource or destination object with a new object

ourceImplement to return the object that's the source of the connection

te:(NXTypedStream *)streamImplement to archive the connector object to stream

chMarks the document as edited

DocumentPathIn:(char *)bufferPlaces the document's path in buffer

type:(NXAtom)type
inPasteboard:(Pasteboard *)aPasteboard
Objects:(List *)objectListCopies the objects in objectList to the specified pasteboard
type:(NXAtom)type
inPasteboard:(Pasteboard *)aPasteboard
st *)pasteType:(NXAtom)typeAlerts the document object that objects were pasted
fromPasteboard:(Pasteboard *)pboard
parent:theParent
BOOL)objectIsMember:anObjectReturns YES if anObject is a part of the object hierarchy
Objects:(List *)objectListPlaces the objects from the object hierarchy into objectList
ParentForObject:anObjectReturns the object above anObject in the object hierarchy

BOOL)setName:(const char *)nameSets the name associated with the anObject
for:anObject
NameIn:(char *)name Places the name associated with anObject in the buffer
for:anObjectname

Connector:aConnectorAdds a connector object to Interface Builder's list
moveConnector:aConnectorRemoves aConnector from the list of connectors
Connectors:(List *)aListPlaces in aList connector objects whose sources
forSource:aSourceare aSource
Connectors:(List *)aListPlaces in aList connector objects whose destinations
forDestination:aDestinationare aDestination
Connectors:(List *)aListPlaces in aList the connector objects of class
forSource:aSourcefilterClass whose sources are aSource
filterClass:filterClass
Connectors:(List *)aListPlaces in aList the connector objects of class
forDestination:aDestinationfilterClass whose destinations are aDestination
filterClass:filterClass

SelectionFrom:anEditorRegisters anEditor as the editor that owns the selection
torDidClose:anEditorInforms the document object that anEditor is no longer
for:anObjectactive
Editor:(BOOL)createItReturns the editor object for anObject
for:anObject
BOOL)openEditorFor:anObjectOpens the editor object for anObject

drawObject:anObjectRedraws the selected object by opening required editors

With:anObjectImplement to initialize a newly allocated editor
inDocument:aDocument

documentImplement this method to return the active document

editedObjectImplement to return the object that's being edited

windowImplement to return the editor window

setObject:anObjectImplement to redraw anObject

(BOOL)wantsSelectionImplement to return YES if the editor is willing to become the selection owner

selectObjects:(List *)objectListImplement to draw the objects in objectList as selected

makeSelectionVisible:(BOOL)showItImplement to make the current selection visible

(BOOL)copySelectionImplement to copy the selected object(s) to the pasteboard

(BOOL)deleteSelectionImplement to delete the selected object(s)

(BOOL)pasteInSelectionImplement to paste object(s) from pasteboard into selection

(NXAtom)acceptsTypeFrom:(const NXAtom *)typeList

Implement to return the pasteboard type your editor accepts

closeImplement to close the editor and free its resources

openSubeditorFor:anObjectImplement to open the subeditor for anObject

closeSubeditorsImplement to close all subeditors

moveToFrontImplement to bring the editor's window to the front

(BOOL)activateImplement to activate the editor

commitSenderImplement in subclass to commit changes made in the Inspector panel

loadData:senderImplement in subclass to load data into inspector's display

(BOOL)wantsButtonsReturns whether the inspector requires Interface Builder to display OK and Revert buttons

nst char *)getSizeInspectorClassNameReturns the class name of the receiver's size inspector

SelectionInto:(List *)objectListImplement to place the selected objects into objectList
drawSelectionImplement to redraw the objects in the selection
signed)selectionCountImplement to return the number of objects in the editor's selection