
NSDataLink

Inherits From:	NSObject
Conforms To:	NSCoding, NSCopying NSObject (NSObject)
Declared In:	AppKit/NSDataLink.h

Class Description

An NSDataLink object (or *data link*) defines a single link between a selection in a source document and a dependent, dynamically updated selection in a destination document.

A data link is typically created when linkable data is copied to the pasteboard. First, an NSSelection object describing the data is created. Then a link to that selection is created using **initWithSourceSelection:managedBy:supportingTypes:**. The link can then be written to the pasteboard using **writeToPasteboard:**. Usually, after the link has been written to the pasteboard (or saved to a file using **writeToFile:**) the link is released because it's generally of no further use to the source application.

Once the data and link have been written to the pasteboard, they can be added to a destination document by an object that can respond to a message to Paste and Link. The object responding to this message will paste the data as usual. The destination application will then read the link from the pasteboard using **initWithPasteboard:**, create an NSSelection describing the linked data within the destination document, and will add the link by sending **addLink:at:** to the document's NSDataLinkManager object (also known as a *data link manager* or simply *link manager*).

When the link is added to the destination document's link manager, it becomes a *destination link*. At that time, the data link's object establishes a connection with the source document's link manager, which automatically creates a *source link* in the source application; the source link refers to the source selection.

A link that isn't managed by a link manager is a *broken link*. (Both source and destination links have link managers.) All links are broken links when they are created. Links can be explicitly broken (ensuring that they cause no updates) using the **break** method. Broken links (that aren't former source links) can be hooked up as destination links with the **addLink:at:** method. The disposition of a link (destination, source, or broken) can be retrieved with the **disposition** method. Most of the messages defined by the NSDataLink class can be sent to a link of any disposition, but some only make sense when sent to a link with a specific disposition; these are so noted in their method descriptions.

Links of all dispositions (except links to files) maintain an NSSelection object referring to the link's selection in the source document; this selection is returned by the **sourceSelection** method. Links directly

to files represent entire files rather than selections in a document; these links are created with **initWithFile:** and have no source selection.

Source and destination links also maintain an **NSSelection** describing the location of the data in the destination document; this selection is returned by the **destinationSelection** method.

See the **NSSelection** class description for more information on **NSSelection** objects.

Adopted Protocols

NSCoding	– encodeWithCoder: – initWithCoder:
NSCopying	– copyWithZone:

Method Types

Creating an NSDataLink	– initWithFile: – initWithSourceSelection:managedBy:supportingTypes: – initWithContentsOfFile: – initWithPasteboard: – copyWithZone:
Exporting the link	– saveLinkIn: – writeToFile: – writeToPasteboard:
Getting information about the link	– disposition – linkNumber – manager
Getting information about the link's source	– lastUpdateTime – openSource – sourceApplicationName – sourceFilename – sourceSelection – types
Getting information about the link's destination	– destinationApplicationName – destinationFilename – destinationSelection

Changing the link

- break
- noteSourceEdited
- setUpdateMode:
- updateMode
- updateDestination

Instance Methods

break

- (BOOL)**break**

Breaks the link so the data referred to by its selection will not get updated.

copyWithZone:

- @protocol NSCopying
- (id)**copyWithZone:**(NSZone *)*zone*

Returns a copy of the receiving data link allocated from *zone*. The copy is essentially linked to the source data, but not hooked up to the destination document. The copy has a copy of the receiver's source selection, has no destination selection, and its disposition is NSLinkBroken.

See also: – **addLink:at:** (NSDataLinkManager)

destinationApplicationName

- (NSString *)**destinationApplicationName**

Returns the name of the application that owns the destination document.

See also: – **destinationFilename**, – **sourceApplicationName**

destinationFilename

- (NSString *)**destinationFilename**

Returns the file name of the destination document.

See also: – **destinationApplicationName**, – **sourceFilename**

destinationSelection

– (NSSelection *)**destinationSelection**

Returns the destination selection.

See also: – **sourceSelection**

disposition

– (NSDataLinkDisposition)**disposition**

Identifies the link as a destination link, a source link, or a broken link by returning one of the following values:

NSLinkInDestination

NSLinkInSource

NSLinkBroken

initWithFile:

– (id)**initWithFile:**(NSString *)*filename*

Initializes a new instance corresponding to the entire file *filename*.

See also: – **addLink:at:** (NSDataLinkManager), – **writeToPasteboard:**, – **sourceSelection**

initWithSourceSelection:managedBy:supportingTypes:

– (id)**initWithSourceSelection:**(NSSelection *)*selection*
 managedBy:(NSDataLinkManager *)*linkManager*
 supportingTypes:(NSArray *)*newTypes*

Initializes a newly allocated instance corresponding to a selection in the source document *selection*. *linkManager* is the source document's link manager. *newTypes* is an array of the types that *linkManager*'s delegate is willing to provide (using **copyToPasteboard:at:cheapCopyAllowed:**) when a destination of the link requests the data described by *selection*.

See also: – **dataLinkManager:startTrackingLink:** (NSDataLinkManager delegate)

initWithContentsOfFile:

– (id)**initWithContentsOfFile:**(NSString *)*filename*

Initializes a new instance from *filename*, a link that was previously saved using the **saveLinkIn:** or **writeToFile:** method.

See also: – **saveLinkIn:**, – **writeToFile:**

initWithPasteboard:

– (id)**initWithPasteboard:**(NSPasteboard *)*pasteboard*

Initializes a new instance from *pasteboard*. The new link is generally used by adding it to a destination document's link manager with **addLink:at:**.

For this method to succeed, a link must have been placed on the pasteboard using **writeToPasteboard:**, or the file name of a saved link (data of type NSFileNamesPboardType with an extension of NSDataLinkFilenameExtension) must be on the pasteboard.

See also: – **saveLinkIn:**

lastUpdateTime

– (NSDate *)**lastUpdateTime**

Returns the last time the link was updated.

linkNumber

– (NSDataLinkNumber)**linkNumber**

Returns the link's number. This number is constant through the life of the document, and unique among the document's links.

manager

– (NSDataLinkManager *)**manager**

Returns the link's manager, or **nil** if it doesn't have one (for instance, if the link is broken).

noteSourceEdited

– (void)**noteSourceEdited**

Notifies a source link that the data referred to by its source selection has changed. If the link's destination link has been set to update continuously, the destination will be updated.

This message only has meaning if sent to a source link.

openSource

– (BOOL)**openSource**

Opens the source document of the link and makes the source selection visible. This message only has meaning when sent to a destination link.

saveLinkIn:

– (BOOL)**saveLinkIn:**(NSString *)*directoryName*

Saves the link in a file specified by the user through an NSSavePanel. The NSSavePanel's initial directory is set to *directoryName*.

See also: – initWithContentsOfFile:

setUpdateMode:

– (void)**setUpdateMode:**(NSDataLinkUpdateMode)*mode*

Sets the link's update mode to *mode*, which must be one of the following:

NSUpdateContinuously

NSUpdateWhenSourceSaved

NSUpdateManually

NSUpdateNever

A mode of NSUpdateContinuously updates the link's destination data every time the source is edited.

NSUpdateWhenSourceSaved updates the link's destination data every time the source is saved.

NSUpdateManually updates the link's destination data every time an **updateDestination** message is sent to the destination link; this message can be sent programmatically or by the data link panel.

NSUpdateNever keeps the link from ever updating; once a destination link has been set to this mode, it can't be set back to any other mode until it is broken. (This mode is used for link buttons, for example.)

This message only has meaning when sent to a destination link or to a broken link.

See also: – `updateMode`

sourceApplicationName

– (NSString *)**sourceApplicationName**

Returns the name of the application that owns the source document.

See also: – `sourceFilename`, – `destinationApplicationName`

sourceFilename

– (NSString *)**sourceFilename**

Returns the file name of the source document.

See also: – `sourceApplicationName`, – `destinationFilename`

sourceSelection

– (NSSelection *)**sourceSelection**

Returns the source selection.

See also: – `destinationSelection`

types

– (NSArray *)**types**

Returns the types that the source document can provide.

updateDestination

– (BOOL)**updateDestination**

Updates the data referred to by the link's destination selection with the contents referred to by the source selection.

updateMode

– (NSDataLinkUpdateMode)**updateMode**

Returns the link's update mode, which determines when the data referred to by the link's destination selection will be updated.

The update modes are listed and described in the method description for **setUpdateMode:**.

writeToFile:

– (BOOL)**writeToFile:**(NSString *)*filename*

Writes the link into *filename*, returning NO if the file can't be written.

See also: – **saveLinkIn:**, – **initWithContentsOfFile:**

writeToPasteboard:

– (void)**writeToPasteboard:**(NSPasteboard *)*pasteboard*

Writes the link onto *pasteboard*. When a link is written to a pasteboard, the type **NSDataLinkPboardType** must be included in the pasteboard's types.

See also: – **initWithPasteboard:**