

NSDataLinkManager

Inherits From: NSObject
Conforms To: appkit/NSDataLinkManager.h

Initializing and Freeing a Link Manager

- (id)**initWithDelegate:**(id)*anObject* Initializes and returns a newly allocated instance, designating *anObject* as the delegate.
- (id)**initWithDelegate:fromFile:**(id)*anObject* (NSString *)*path* Initializes and returns a newly allocated instance designating *anObject* as the delegate. The document's file is specified by the full path *path*.

Adding and Removing Links

- (BOOL)**addLink:**(NSDataLink *)*link* **at:**(NSSelection *)*selection* Adds the link *link* to the document, indicating that the data in the document described by *selection* is dependent upon the link.
- (BOOL)**addLinkAsMarker:**(NSDataLink *)*link* **at:**(NSSelection *)*selection* Incorporates *link* into the document as a marker in the location of the destination document described by *selection*.
- (NSDataLink *)**addLinkPreviouslyAt:**(NSSelection *)*oldSelection* **fromPasteboard:**(NSPasteboard *)*pasteboard* **at:**(NSSelection *)*selection* Creates and adds a new destination link corresponding to the same source

data as the link described by the destination selection *oldSelection* with the new link's destination selection provided in *selection*; the document's links must have been written to the pasteboard *pasteboard*.

- (void)**breakAllLinks** Breaks all the destination links in the document.
- (void)**writeLinksToPasteboard:**(NSPasteboard *)*pasteboard* Writes all the link manager's links to *pasteboard*.

Informing the Link Manager of Document Status

- (void)**noteDocumentClosed** Informs link manager that document has been closed.
- (void)**noteDocumentEdited** Informs link manager that document has been edited.
- (void)**noteDocumentReverted** Informs link manager that changes have been reverted.
- (void)**noteDocumentSaved** Informs link manager that document has been saved.
- (void)**noteDocumentSavedAs:**(NSString *)*path* Informs link manager that document has been saved in the file specified by the full pathname *path*.
- (void)**noteDocumentSavedTo:**(NSString *)*path* Informs link manager that document has been saved in the file specified by the full pathname *path*.

Getting and Setting Information about the Link Manager

- (id)**delegate** Returns the data link manager's delegate.
- (BOOL)**delegateVerifiesLinks** Returns YES if delegate is asked to verify updates.
- (NSString *)**filename** Returns the filename for the link manager's document.
- (BOOL)**interactsWithUser** Tells whether the link manager displays panels if link errors occur.
- (BOOL)**isEdited** Returns YES if the document was edited since the last save.
- (void)**setDelegateVerifiesLinks:**(BOOL)*flag* Sets whether the delegate is asked to verify updates.
- (void)**setInteractsWithUser:**(BOOL)*flag* Sets whether the link manager displays panels if link errors occur.

Getting and Setting Information about the Manager's Links

- (BOOL)**areLinkOutlinesVisible** Returns YES if outlines are visible.
- (NSEnumerator *)**destinationLinkEnumerator** Returns an enumerator of the destination's source links.
- (NSDataLink *)**destinationLinkWithSelection:(NSSelection *)*destSel***
Returns the destination link for the selection *destSel*.
- (void)**setLinkOutlinesVisible:(BOOL)*flag*** Sets whether outlines are visible.
- (NSEnumerator *)**sourceLinkEnumerator** Returns an enumerator of the receiver's source links.

Methods Implemented by the Delegate

- (BOOL)**copyToPasteboard:(NSPasteboard *)*pasteboard***
at:(NSSelection *)*selection*
cheapCopyAllowed:(BOOL)*flag* Implemented by the link manager's delegate to supply the source data described by *selection* on the pasteboard *pasteboard*. If *flag* is YES, the system guarantees that no events will be processed by the application before the delegate is requested to provide the specified data; in this case, the application doesn't necessarily have to write any data representations to the pasteboard. This method should return YES upon success, or NO if the selection can't be resolved.
- (void)**dataLinkManager:(NSDataLinkManager *)*sender***
didBreakLink:(NSDataLink *)*link* Informs the delegate that the destination link *link* was broken and thus data based on the link's destination selection will no longer be updated.
- (BOOL)**dataLinkManager:(NSDataLinkManager *)*sender***
isUpdateNeededForLink:(NSDataLink *)*link* Returns YES if the source data identified by *link*'s source selection has been modified since the link's last update time.
- (void)**dataLinkManager:(NSDataLinkManager *)*sender***
startTrackingLink:(NSDataLink *)*link* Informs the delegate that a destination document has established a data link *link* to the link manager's document and is tracking it.
- (void)**dataLinkManager:(NSDataLinkManager *)*sender***
stopTrackingLink:(NSDataLink *)*link* Informs the delegate that a destination is no longer tracking the source link *link*.
- (void)**dataLinkManagerCloseDocument:(NSDataLinkManager *)*sender***
Closes documents opened without the user interface.

- (void)**dataLinkManagerDidEditLinks:**(NSDataLinkManager *)*sender*
Informs the delegate that link data has been modified; the delegate should use this notification to mark the document as edited.
- (void)**dataLinkManagerRedrawLinkOutlines:**(NSDataLinkManager *)*sender*
Directs the delegate to redraw objects with link outlines.
- (BOOL)**dataLinkManagerTracksLinksIndividually:**(NSDataLinkManager *)*sender*
Returns whether the receiver is willing to track links individually.
- (BOOL)**importFile:**(NSString *)*filename*
at:(NSSelection *)*selection*
Imports the file *filename* at the destination described by *selection*. Returns YES upon success, or NO if the selection can't be resolved.
- (BOOL)**pasteFromPasteboard:**(NSPasteboard *)*pasteboard*
at:(NSSelection *)*selection*
Pastes the updated data that has been made available on *pasteboard*. The destination for the data is described by *selection*, which was supplied to the link manager as an argument to the **addLink:at:** method. Returns YES upon success, or NO if the selection can't be resolved.
- (BOOL)**showSelection:**(NSSelection *)*selection*
Shows the source data for the specified selection *selection*. Returns YES upon success, or NO if the selection can't be resolved.
- (NSWindow *)**windowForSelection:**(NSSelection *)*selection*
Returns the NSWindow object for the given *selection*, or **nil** if the selection can't be resolved.