	initWithDelegate:
	initWithDelegate:fromFile:
	free
Adding and removing links	addLink:at:
	addLinkAsMarker:at:
	writeLinksToPasteboard:
	addLinkPreviouslyAt:fromPasteboard:at:
	breakAllLinks
Informing the link manager of document status	
	documentClosed
	documentEdited
	documentReverted
	documentSaved
	documentSavedAs:
	documentSavedTo:
Getting and setting information about the link manager	
	filename
	isEdited
	setLinksVerifiedByDelegate:
	areLinksVerifiedByDelegate
	delegate
	setInteractsWithUser:
	interactsWithUser
Getting and setting information about the manager's links	
	setLinkOutlinesVisible:
	areLinkOutlinesVisible
	findDestinationLinkWithSelection:
	prepareEnumerationState:forLinksOfType:
	nextLinkUsing:

addLink:(NXDataLink *)link at:(NXSelection *)selection

Adds the link link to the document, indicating that the data in the document described by selection is dependent upon the link. This method is invoked as part of the Paste and Link command to actually link in the data that was just pasted. It can also be used at other times for example, to link to files that are dragged into the document.

addLinkAsMarker:(NXDataLink *)link at:(NXSelection *)selection

Incorporates link into the document as a marker. This method is used to implement link buttons th
link's source, but are never asked to receive data from the source document. The link button in the
is described by selection. This method adds the link and, upon success, sets its the link's update m
NX_UpdateNever. Returns self upon success, nil otherwise.

The named images ªNXLinkButtonº and ªNXLinkButtonHº can be used (through NXImage's find
to represent ordinary and highlighted link buttons, respectively. These images are shared, so you m
NXImage's copy method) if you need to scale them to a different size.

addLink:at:

(NXDataLink *)addLinkPreviouslyAt:(NXSelection *)oldSelection
fromPasteboard:(Pasteboard *)pasteboard
at:(NXSelection *)selection

Creates and adds a new destination link corresponding to the same source data as the link describe
selection oldSelection. The new link's destination selection is provided in selection. This method
data that is already linked. It's similar to copying the old link and adding it using addLink:at:, exc
destination selection rather than the old link. Before invoking this method, the document's links m
pasteboard pasteboard using writeLinksToPasteboard:. Returns the new link if it's successfully ad
can't be added or no link for oldSelection existed.

(BOOL)areLinkOutlinesVisible

Used to inform the link manager's delegate of whether link outlines should be drawn around linke
When the delegate receives a dataLinkManagerRedrawLinkOutlines: message, it should query the
areLinkOutlinesVisible message. If this message returns YES, the delegate should call the NXFra
function to draw a distinctive link outline around the dependent data.

setLinkOutlinesVisible:

(BOOL)areLinksVerifiedByDelegate

Return YES if the link manager's delegate will be asked to verify whether data based on the delega
needs to be updated. If so, the delegate should implement the dataLinkManager:isUpdateNeededF
Returns NO by default, but the application can change this by sending the link manager a setLinks
message.

breakAllLinks

Breaks all the destination links in the document by sending each link a break message. This metho
by the application's data link panel in response to user input. Returns self.

break (NXDataLink), pickedBreakAllLinks: (NXDataLinkPanel) documentClosed

An application should send this message to the link manager to inform it that the manager's docum
Returns self.

documentEdited

An application should send this message to the link manager to inform it that the manager's docum
the delegate doesn't track source links individually, this method marks all source links as dirty, ind
dependent destination data will eventually need to be updated. Returns self.

dataLinkManagerTracksLinksIndividually: (NXDataLinkManager delegate)

documentReverted

An application should send this message to the link manager to inform it that the manager's docum
to the last saved copy. This method then restores the link manager and its links to their last saved s
the link manager received a documentSaved or documentSavedAs: message). Returns self.

documentSaved

An application should send this message to the link manager to inform it that the manager's docum
This method stores the document's destination links and, if necessary, initiates updates of other do
upon the document's source links. Returns self.

documentSavedAs:(const char *)path

An application should send this message to the link manager to inform it that the manager's docum
the file specified by the full pathname path. This method stores the document's destination links.
manager's source links, since the documents for those links are not dependent upon the newly save

documentSavedTo:(const char *)path

An application should send this message to the link manager to inform it that a copy of the manage
saved to the file specified by the full pathname path. This method stores the appropriate link infor
file, and returns self.

(const char *)filename

Returns the name of the file for the link manager's document. This is the name that was set with th
fromFile: or documentSavedAs: method.

Notifies the link managers of dependent documents that the link manager is going away, and frees
held by the link manager.

init

There is no need to call this method use one of the other init... methods to initialize a newly allocat
NXDataLinkManager instance for a new document.

initWithDelegate:fromFile:

initWithDelegate:anObject

Initializes and returns a newly allocated NXDataLinkManager instance for a new document. The l
delegate, specified by anObject, will be expected to provide source data, paste destination data, and
manager keep links up-to-date. Before data in the document can be linked to, the document will ha
link manager will have to be informed of the document's name by a documentSavedAs: message.

initWithDelegate:fromFile:

initWithDelegate:anObject fromFile:(const char *)path

Initializes a newly allocated NXDataLinkManager instance for a new document. The link manage
by anObject, will be expected to to provide source data, paste destination data, and help the data lin
up-to-date. The document's file is specified by the full path path. The file must exist or initializat

See ªMethods Implemented by the Delegateº at the end of this class specification for information a
delegate should implement to assist the link manager.

Returns the new link manager upon success frees the allocated storage and returns nil if initializati

initWithDelegate:fromFile:

(BOOL)interactsWithUser

Returns YES if the link manager should display alert panels when problems with links occur, NO i
suppressed. This value is set with the setInteractsWithUser: method the default value is YES.

(BOOL)isEdited

Returns YES if the document has been edited since the last save, or NO if the file for the documen
document's edited state is set by the documentEdited method, and cleared by documentSaved and

(NXDataLink *)nextLinkUsing:(NXLinkEnumerationState *)state

with later invocations of nextLinkUsing:. srcOrDest must be either NX_LinkInDestination or NX_
indicate whether the nextLinkUsing: method is to return the next destination link or the next sourc
Returns self if there is one or more links of the requested type, or nil if there is none.

setInteractsWithUser:(BOOL)flag

Instructs the link manager as to whether it should display alert panels when problems with links oc
default value), alert panels will be displayed. Returns self.

interactsWithUser

setLinkOutlinesVisible:(BOOL)flag

Sets the internal flag indicating to the link manager's delegate whether link outlines ought to be di
be returned by areLinkOutlinesVisible.

If the link manager's delegate implements the dataLinkManagerRedrawLinkOutlines: method, this
to the delegate and it should either display link outlines using NXFrameLinkRect() or erase link ou
previously displayed, based on the return value of areLinkOutlinesVisible.

Returns self.

setLinksVerifiedByDelegate:(BOOL)flag

Sets whether the update status of links will be individually verified by the link manager's delegate.
delegate must implement the dataLinkManager:isUpdateNeededForLink: method to tell the link m
a source link needs to be updated.

By default, the update status of an individual link isn't verified by the delegate, so the link manage
on its last update time. An example where this verification could be incorrect might be a link to a
query itself doesn't change, the link manager might return that data is up-to-date, even though the
the query might have changed.

areLinksVerifiedByDelegate

writeLinksToPasteboard:(Pasteboard *)pasteboard

Writes all the link manager's links to the pasteboard pasteboard in preparation for an invocation of
fromPasteboard:at:, which will expect to find one link matching its specified selection.

The links are written with Pasteboard's addTypes:num:owner: method, which doesn't change the p
change count, using a private pasteboard type.

copyToPasteboard:(Pasteboard *)pasteboard
    at:(NXSelection *)selection

This method should return self upon success, or nil if the selection can't be resolved.

pasteFromPasteboard:at: (NXDataLinkManager delegate), declareTypes:num:owner: (Pasteboard
provideData: (Pasteboard owner)

(NXSelection *)createSelection

Never invoked by the system.

dataLinkManager:(NXDataLinkManager *)sender
    didBreakLink:(NXDataLink *)link

If this method is implemented by the delegate, it will be invoked to inform the delegate that the de
broken and thus data based on link's destination selection will no longer be updated.

The link shouldn't be be sent a free message at this time, because the method that invoked dataLin
didBreakLink: may still reference the link. However, the link can be freed with Application's dela
Alternatively, the link could be kept around for a while in order to allow the break operation to be
requested, the link could be re-added with addLink:at:.

break (NXDataLink), destinationSelection (NXDataLink)

(BOOL)dataLinkManager:(NXDataLinkManager *)sender
    isUpdateNeededForLink:(NXDataLink *)link

A delegate that sends a setLinksVerifiedByDelegate: message to the link manager (indicating that
individual links will be verified by the delegate) should implement this method and return YES if t
identified by link's source selection has been modified since the link's last update time.

lastUpdateTime (NXDataLink)

dataLinkManager:(NXDataLinkManager *)sender
    startTrackingLink:(NXDataLink *)link

Informs the delegate that another document has established a data link to the link manager's docun
need only implement this method if it returns YES in response to a dataLinkManagerTracksLinksI
link is a newly added source link the data that it applies to is identified by link's source selection.

dataLinkManagerTracksLinksIndividually:

dataLinkManager:(NXDataLinkManager *)sender
    stopTrackingLink:(NXDataLink *)link

Informs the delegate that the former source link link is no longer linked to the document. There ar
link might be removed the destination document could get closed, the link could be explicitly brok
application might have died.

dataLinkManagerTracksLinksIndividually:

dataLinkManagerDidEditLinks:(NXDataLinkManager *)sender

Informs the delegate that link data has been modified. Since the link data is stored alongside the d
should be considered part of the document, the delegate should use this notification to mark the do

dataLinkManagerRedrawLinkOutlines:(NXDataLinkManager *)sender

If the delegate implements this method, it will be invoked any time the manager is instructed to sh
through setLinkOutlinesVisible:. This method should query the link manager with areLinkOutline
link outlines should be displayed. If so, it should invoke NXFrameLinkRect() to draw a distinctiv
linked data otherwise it should display the data without outlines.

(BOOL)dataLinkManagerTracksLinksIndividually:
(NXDataLinkManager *)sender

If the delegate implements this method it should return whether it's willing to track links individua
doesn't implement this method, links are not individually tracked. If the delegate implements this
YES, it should also implement dataLinkManager:startTrackingLink: and dataLinkManager:stopTr
the links in use.

Many applications do not need to track links individually, but there are several situations where it
when a link is used. For example, the delegate may want to individually track links in order to co
destination documents each time data for a link's source selection is modified an individual link ca
updateDestination message whenever a modification is made that affects the destination.

Additionally, many links may be placed on the pasteboard when data is copied, but few of those li
get used. If the application must store selection-state information in the document, it should only
their associated links) that actually get used this method is used to find out if the delegate wants to
link gets used.

importFile:(const char *)filename at:(NXSelection *)selection

If the application has added a link based on an entire file (that is, used addLink:at: to incorporate a
initLinkedToFile:), the delegate must implement this method to import the filename file at the dest
selection. This method should return self upon success, or nil if the selection can't be resolved.

pasteFromPasteboard:(Pasteboard *)pasteboard at:(NXSelection *)selection

If the application has added an ordinary destination link (that is, used addLink:at: to incorporate a
initFromPasteboard: or a related method), the delegate must implement this method to paste the up
been made available on the pasteboard. The destination for the data is described by selection, whi
link manager as an argument to the addLink:at: method.

The data is read from the pasteboard just as it is for any ordinary paste see the Pasteboard class sp
information on reading data from a pasteboard. This method should return self upon success, or n
be resolved.

specified selection selection. This method should scroll the document so the selected data is visibl
highlight the selected data using the function NXFrameLinkRect() with the argument isDestinatior
method should return self upon success, or nil if the selection can't be resolved.

windowForSelection:(NXSelection *)selection

In an application that serves as a link source, the delegate should implement this method to return t
the given selection, or nil if the selection can't be resolved.