

MiscLinkedListNode

Inherits From:	Object
Declared In:	misckit/MiscLinkedListNode.h
Protocols:	NXTransport

Class Description

The MiscLinkedListNode class is a class used by the MiscLinkedList to implement a doubly linked list. In most cases, you will not need to concern yourself with this class. If you were to make a custom subclass of the MiscLinkedList, however, you might find it useful to also make a subclass of MiscLinkedListNode to work with it. For example, if you were using MiscLinkedLists to keep track of graphic objects in a drawing program, you might want to have the linked list keep track of which nodes are "selected". To do this, you would probably subclass the MiscLinkedListNode and add an instance variable to track selection status. (This tends to make more sense than requiring the graphic to know if it is selected, since typically graphics are selected from a list but don't±or shouldn't±have a notion of being in a list. NeXT's Draw is a bad example of this sort of thing.) Once creating a node that knows selection status, you would use this new node along with a subclass of MiscLinkedList that understands how to manipulate selected nodes.

The MiscLinkedListNode class knows three things: the object it "contains" and which MiscLinkedListNodes are the next and previous nodes in the MiscLinkedList to which this node belongs. When a MiscLinkedListNode is created, it is

normally given an object to contain via the `±initObject:` method. To change or access the contained object, you can use the `±getObject`, `±setObject:`, or `±freeObject` methods.

The previous `MiscLinkedListNode` in the `MiscLinkedList` is accessed via the `±getPrevious` and `±setPrevious:` methods. The next node in the list is accessed via `±getNext` and `±setNext:` methods.

`MiscLinkedListNode` also implements `±read:` and `±write:` as well as the `NXTransport` protocol, allowing it to be archived onto a stream or sent to another application via distributed objects. Note that typically a `MiscLinkedList` is needed to oversee the archiving and make sure that the pointers are set up properly. (See the descriptions of `±read:` and `±write:` for more information.)

Instance Variables

id `the_object`;
id `the_next_node`;
id `the_previous_node`;

<code>the_object</code>	The object contained by this <code>MiscLinkedListNode</code> .
<code>the_next_node</code>	The next node in the <code>MiscLinkedList</code> to which this node belongs.
<code>the_previous_node</code>	The previous node in the <code>MiscLinkedList</code> to which this node belongs.

Method Types

Initializing a `MiscLinkedListNode`: `± initObject:`

Accessing contained object: ± freeObject
 ± getObject
 ± setObject:

Accessing next node: ± getNext
 ± setNext:

Accessing previous node: ± getPrevious
 ± setPrevious:

Archiving a MiscLinkedListNode: ± read:
 ± write:

Instance Methods

freeObject
 - freeObject

Free *the_object* and set it to nil. Returns *self*.

See also: **-getObject**, **-initWithObject:**, and **-setObject:**

getNext
 - getNext

Returns the next MiscLinkedListNode in the MiscLinkedList to which the receiver belongs.

See also: **-getPrevious**, **±setNext:**, and **±setPrevious:**

getObject

- **getObject**

Returns the id of the object contained by the receiver.

See also: **-freeObject**, **-initObject:**, and **-setObject:**

getPrevious

- **getPrevious**

Returns the previous MiscLinkedListNode in the MiscLinkedList to which the receiver belongs.

See also: **-getNext**, **±setNext:** and **±setPrevious:**

initObject

- **initObject:***this_object*

Initializes a new instance of MiscLinkedListNode and sets *the_object* to *this_object*. Returns *self*.

See also: **-freeObject**, **-getObject**, and **-setObject:**

read

- **read:**(NXTypedStream *)*stream*

Reads a MiscLinkedListNode from the stream *stream*. This method leaves *the_next_node* and *the_previous_node* empty. Typically, a MiscLinkedList object will set these instance variables as appropriate while unarchiving itself. Returns *self*.

See also: **-write:**

setNext

- **setNext:***this_node*

Set *the_next_node* to *this_node*. Returns *this_node*.

See also: **-getNext**, **-getPrevious**, and **-setPrevious**:

setObject

- **setObject:***this_object*

Set *the_object* to *this_object*. Returns *this_object*.

See also: **-freeObject**, **-getObject**, and **±initObject**:

setPrevious

- **setPrevious:***this_node*

Set *the_previous_node* to *this_node*. Returns *this_node*.

See also: **-getNext**, **-getPrevious**, and **-setNext**:

write

- **write:**(NXTypedStream *)*stream*

Writes a MiscLinkedListNode to the stream *stream*. This method does not archive *the_next_node* or *the_previous_node*. Typically, a MiscLinkedList object will archive itself in order from first to last, so there is no need for this data to be archived. Returns *self*.

See also: **-read**:

