Text (MiscExtensions)

Declared In: Text_MiscExtensions.h

Depends On: <appkit/appkit.h>

Category Description

This category is intendet to simplify the work with NeXTs Text class. Currently it only adds methods that will allow you to cut/copy and paste to a specified pasteboard. This is very useful because it is nasty to currupt the main pasteboard when dealing with the normal cut/copy/paste methods.

The mecahnism of cut/copy/paste is very powerful when combined with the Texts find method.

In the future we should have better methods for finding lines, strings that spand multiple runs (different fonts etc.) and many other things that we are used to from the MiscStrings.

Note: The implementation requires a nasty hack because it uses a yet *undocumented* pasteboard type for RTFD text selections. The source for MailHelper (by Izumi Ohzawa and Manuel Alberto Ricart) brought me to this idea.
It was tested using NeXTSTEP 3.2 and should work on all the other 3.x releases too.

Remember that making selections inside on text object using **selectAll:** and its companions causes other text objects to forget about their selections if they are both in the same window (firstResponder mechanism).

All the text run handling might not work on NSJ. I'm not able to test on a Japanese release.

Method Types

Cutting and Pasting
- copyTo:
- cutTo:
- pasteFrom:

Class Methods

Instance Methods

substringFromLine:

- **substringFromLine:**(int)*line*

Method description here.

See also: **- myReference**

findNextWord

- **findNextWord**

Method description here.

See also: **- myReference**

findPerviousWord

- **findPerviousWord**

Method description here.

See also: **- myReference**

findText:ignoreCase:backwards:wrap:font:

- (BOOL)**findText:**(const char *)*string* **ignoreCase:**(BOOL)*ignoreCaseflag* **backwards:**(BOOL)*backwardsflag* **wrap:**(BOOL)*wrapflag* **font:***aSharedFont*

Searches for a given text but requires that the found portion of text is typeset in a special *font*. This should always be the shared font object for a certain typeface.

See also: - findText:ignoreCase:backwards:wrap:

fontAtPosition:
 - fontAtPosition:(int)*position*

Returns the shared font object for a given position. Invalid *positions* will be handled as the run search method does.

See also: - fontsInsideSelection:, **- textRunForPosition:**

fontsInsideSelection
 - fontsInsideSelection

Returns a List of Font objects. It is the receivers responsibility to free the List. Don't free the objects inside the list. Those are the shared font objects!
If you need details information use the NXRuns directly.

Note: I'm not sure about this method. It might disapear in favour of a general **selTextRunCount** method. Suggestions please.

See also: - fontAtPosition:, **- textRunForPosition:**

textRunForPosition:
 - (NXRun *)**textRunForPosition:**(int)*position*

Returns the NXRun which holds the information on the specified text position. If the number is out of range it either returns the first or the last NXRun, depending on the given location.
The *position* references the number of the character like the other text methods do (0 = first,...)

See also: - firstTextBlock

hasEmptySelection
 - (BOOL)**hasEmptySelection**

This method returns YES is the text has no useful selection. Which means that either the selection has useless offsets or starts and ends at the same spot.

See also: - myReference

copyTo:

- copyTo:*aPasteboard*

This method returns YES is the text has no useful selection. Which means that either the selection has useless offsets or starts and ends at the same spot.

See also: - myReference

cutTo:

- cutTo:*aPasteboard*

Well here we will just copy the selection to the given pasteboard and then delete the selection. Basically this is a cut.

See also: - cut:

pasteFrom:

- pasteFrom:*aPasteboard*

Replaces the current selection with the contents of the specified pasteboard.

See also: - paste:, **- readSelectionFromPasteboard:**