	init
	free
Defining the printing rectangle	setMarginLeft:right:top:bottom:
	getMarginLeft:right:top:bottom:
	setOrientation:andAdjust:
	orientation
	setPaperRect:andAdjust:
	paperRect
	setPaperType:andAdjust:
	paperType
Page range	setFirstPage:
	firstPage
	setLastPage:
	lastPage
	setAllPages:
	isAllPages
	currentPage
Pagination and scaling	setHorizPagination:
	horizPagination
	setVertPagination:
	vertPagination
	setScalingFactor:
	scalingFactor
Positioning the image on the page	setHorizCentered:
	isHorizCentered
	setVertCentered:
	isVertCentered
	setPagesPerSheet:
	pagesPerSheet
Print job attributes	initializeJobDefaults

setPaperFeed:
paperFeed

Specifying the printer+ setDefaultPrinter:
+ getDefaultPrinter
setPrinter:
printer

Spooling setOutputFile:
outputFile
setContext:
context

Archiving read:
write:

setPrinter:, printer

setPrinter:, printer

(DPSContext)context

Returns the Display PostScript context used for printing.

(int)copies

Returns the number of copies that will be printed.

(int)currentPage

Returns the page number of the page currently being printed. This method is valid only while prin

(int)firstPage

Frees the PrintInfo object.

```
getMarginLeft:(NXCoord *)leftMargin
        right:(NXCoord *)rightMargin
        top:(NXCoord *)topMargin
        bottom:(NXCoord *)bottomMargin
```

Returns, by reference, the sizes of the four page margins measured in points.

setMarginLeft:right:top:bottom:

```
(int)horizPagination
```

Returns a constant that represents the manner in which an image is distributed horizontally among description, above, for a description of the pagination constants.

setHorizPagination:

```
init
```

Initializes the PrintInfo object after memory for it has been allocated through alloc or allocFromZo

```
initializeJobDefaults
```

Called before each print job (specifically, before the Print panel is displayed), this method initializo PrintInfo attributes:

```
(BOOL)isAllPages
```

Returns whether all pages will be printed. If NO, only those pages that fall within [firstPage, lastP

setAllPages:

```
(BOOL)isHorizCentered
```

margin.

setVertCentered:

(const char **)jobFeatures

Returns a pointer to an array of pointers that contains the keys to the job features table, the hash tal additional printing-job attributes. You would use these keys as arguments to methods such as valu removeJobFeature:. It's the caller's responsibility to free the pointer to the array, but not the point

setJobFeature:toValue:, valueForJobFeature:, removeJobFeature:

(int)lastPage

Returns the page number of the last page that will be printed, as set through setLastPage:, or MAX been explicitly set. If all pages are being printed, the last page value is ignored during printing. If reversed, this value gives the page number of the first page that will be printed.

setLastPage:, setAllPages:, setReversePageOrder:

(char)orientation

Returns the page orientation as NX_PORTRAIT or NX_LANDSCAPE.

setOrientation:andAdjust:, setPaperType:andAdjust:, setPaperRect:andAdjust:

(const char *)outputFile

Returns the name of the file to which the generated PostScript code is sent. If this is NULL, the co temporary file.

setOutputFile:

(char)pageOrder

Returns a constant that denotes the order in which pages are printed. See setPageOrder: for the pag

setPageOrder:

(short)pagesPerSheet

Returns the number of pages per sheet of paper.

Returns a pointer to a rectangle that gives the size of the paper, measured in points. Note that the r
for its size field the origin of the paper is always (0.0, 0.0).

(const char *)paperType

Returns the paper type. If the type is unknown, an empty string is returned.

(NXPrinter *)printer

Returns the NXPrinter that's used for printing.

read:(NXTypedStream *)stream

Reads the PrintInfo from the typed stream stream.

removeJobFeature:(const char *)key

Removes, from the job-features hash table, the element that corresponds to key.

jobFeatures, setJobFeature:toValue:, valueForJobFeature:

(BOOL)reversePageOrder

Returns YES if the PrintInfo dictates that pages will be printed in reverse order (in other words, if t
through setPageOrder: is reversed).

setReversePageOrder:, setPageOrder:

(float)scalingFactor

Returns the factor by which the image is scaled.

setAllPages:(BOOL)flag

Sets whether all the pages of the document are to be printed (as opposed to a subset given by the fi
values).

setContext:(DPSContext)aContext

setFirstPage:(int)anInt

Sets the page number of the first page that will be printed.

setHorizCentered:(BOOL)flag

Sets whether the image is centered horizontally on a page if flag is NO, the image is flush against t
image spills over more than one page horizontally, then flag is ignored and the image is always aga

setHorizPagination:(int)mode

Sets the way in which a document is divided horizontally into pages. See the class description, abo
constants that you can use as the argument to this method.

setJobFeature:(const char *)feature toValue:(const char *)value

Sets the value of the given job feature. The feature is added to the job-features hash table if it isn't

setLastPage:(int)anInt

Sets the page number of the last page that will be printed.

setMarginLeft:(NXCoord)leftMargin
right:(NXCoord)rightMargin
top:(NXCoord)topMargin
bottom:(NXCoord)bottomMargin

Sets the margins. All margins are in points.

setOrientation:(char)mode andAdjust:(BOOL)flag

Sets the orientation or the page mode should be either NX_PORTRAIT or NX_LANDSCAPE.

If flag is NO, then only the orientation is affected. If flag is YES, the paper rectangle value is upda
orientation.

setOutputFile:(const char *)aString

Sets the name of the file to which the generated PostScript code is sent. If aString is NULL, the co
temporary file.

NX_UNKNOWNORDER

reversePageOrder, pageOrder

setPagesPerSheet:(short)pageCount

Sets the number of pages of the document that are printed on a single sheet of paper. This number
power of two when used by the system.

setPaperFeed:(const char *)paperFeedSlot

Sets the paper feed slot by name. If paperFeedSlot is NULL (or an empty string), any paper feed s
choose the manual feed slot, set the name to ªNXManualº. Any other name that you use must app
documented in the NXPrinter class). Returns self.

setPaperFeed:

setPaperRect:(const NXRect *)aRect andAdjust:(BOOL)flag

Sets the size of the paper, measured in points, that's to be used in printing. The origin of the rectar
0)Ðthe origin field of aRect is ignored.

If flag is NO, then only the paper rectangle is changed. If flag is YES, the orientation and paper ty
to reflect the new size.

setPaperType:(const char *)type andAdjust:(BOOL)flag

Sets the name of the paper type.

If flag is NO, only the paper type is changed. If flag is YES, the paper rectangle and orientation va
reflect the new type (given that type is a recognized paper type).

setPrinter:(NXPrinter *)aPrinter

Sets the printer that's used in subsequent printing jobs.

setReversePageOrder:(BOOL)flag

Sets whether pages are printed in reverse order. This ordering is applied to the page order mode se
setPageOrder:. Returns self.

reversePageOrder, setPageOrder:

Sets whether the default implementation of placePrintRect:offset: in the View class centers the ima
page.

setVertPagination:(int)mode

Sets the way in which a document is divided vertically into pages. See the class description, above
constants that you can use as the argument to this method.

(const char *)valueForJobFeature:(const char *)feature

Returns the value for the given printing feature, as stored in the job-features table.

jobFeatures, setJobFeature:toValue:, removeJobFeature

(int)vertPagination

Returns a constant that represents the manner in which an image is distributed vertically among pa
description, above, for a description of the pagination constants.

setVertPagination:

write:(NXTypedStream *)stream

Writes the PrintInfo to the typed stream stream.