

Defined Types

NXAcknowledge

DECLARED IN appkit/Listener.h

SYNOPSIS

typedef struct _NXAcknowledge {
 msg_header_t header;
 msg_type_t sequenceType;
 int sequence;
 msg_type_t errorType;
 int error;
NS_DEV_DOCFOR:typedef:NXAcknowledge;, } **NXAcknowledge**;

DESCRIPTION NXAcknowledge is the structure of a Listener acknowledgement message.

NXAppkitErrorTokens

DECLARED IN appkit/errors.h

SYNOPSIS

typedef enum _NXAppkitErrorTokens {
 NX_longLine = NX_APPKIT_ERROR_BASE,
 NX_nullSel,
 NX_wordTablesWrite,
 NX_wordTablesRead,
 NX_textBadRead,
 NX_textBadWrite,
 NX_powerOff,
 NX_pasteboardComm,
 NX_mallocError,
 NX_printingComm,
 NX_abortModal,
 NX_abortPrinting,
 NX_illegalSelector,
 NX_appkitVMError,
 NX_badRtfDirective,
 NX_badRtfFontTable,
 NX_badRtfStyleSheet,
 NX_newerTypedStream,
 NX_tiffError,
 NX_printPackageError,
 NX_badRtfColorTable,
 NX_journalAborted,
 NX_draggingError,
 NX_colorUnknown,
 NX_colorBadIO,
 NX_colorNotEditable,

`NX_badBitmapParams,`
`NX_windowServerComm,`
`NX_unavailableFont,`
`NX_PPDIncludeNotFound,`
`NX_PPDParseError,`
`NX_PPDIncludeStackOverflow,`
`NX_PPDIncludeStackUnderflow,`
`NX_rtfPropOverflow`
`NS_DEV_DOCFOR:typedef:NXAppkitErrorTokens;, } NXAppkitErrorTokens;`

DESCRIPTION This enumeration defines the exceptions raised by the Application Kit. (See `NX_RAISE()` for more information.) The constants are:

<code>NX_longLine</code>	Text class: line longer than 16384 characters
<code>NX_nullSel</code>	Text class: operation attempted on empty selection
<code>NX_wordTablesWrite</code>	Error occurred while writing word tables
<code>NX_wordTablesRead</code>	Error occurred while reading word tables
<code>NX_textBadRead</code>	Text class: error reading from file
<code>NX_textBadWrite</code>	Text class: error writing to file
<code>NX_powerOff</code>	Power off exception
<code>NX_pasteboardComm</code>	Communications problem with pbs server
<code>NX_mallocError</code>	malloc problem
<code>NX_printingComm</code>	Problem sending data to npd
<code>NX_abortModal</code>	abortModal message when not running modal
<code>NX_abortPrinting</code>	Printing aborted
<code>NX_illegalSelector</code>	Invalid selector passed to Application Kit
<code>NX_appkitVMError</code>	Error allocating or deallocating virtual memory
<code>NX_badRtfDirective</code>	Invalid RTF directive
<code>NX_badRtfFontTable</code>	Invalid RTF font table
<code>NX_badRtfStyleSheet</code>	Invalid RTF style sheet
<code>NX_newerTypedStream</code>	Version of typed stream more recent than software
<code>NX_tiffError</code>	Error with TIFF operation
<code>NX_printPackageError</code>	Problem loading the print package
<code>NX_badRtfColorTable</code>	Invalid RTF color table
<code>NX_journalAborted</code>	Journaling session was terminated
<code>NX_draggingError</code>	Error messaging drag service
<code>NX_colorUnknown</code>	NXColorList: unknown color name or number
<code>NX_colorBadIO</code>	NXColorList: file read/write error
<code>NX_colorNotEditable</code>	Attempt to change noneditable color list
<code>NX_badBitmapParams</code>	Inconsistent set of bitmap parameters
<code>NX_windowServerComm</code>	Communications problem with the Window Server
<code>NX_unavailableFont</code>	No default font could be found
<code>NX_PPDIncludeNotFound</code>	Include file in PPD file not found
<code>NX_PPDParseError</code>	PPD parsing error
<code>NX_PPDIncludeStackOverflow</code>	PPD include files nested too deep
<code>NX_PPDIncludeStackUnderflow</code>	PPD include file nesting mismatched
<code>NX_rtfPropOverflow</code>	RTF property stack overflow

NXBreakArray

DECLARED IN appkit/Text.h

SYNOPSIS typedef struct _NXBreakArray {

`NXChunk chunk;`
`NXLineDesc breaks[1];`
`NS_DEV_DOCFOR:typedef:NXBreakArray;, } NXBreakArray;`

DESCRIPTION An NXBreakArray holds line break information for a Text object. It's mainly an array of line descriptors. Each line descriptor contains three fields:

- 1) Line change bit (sign bit); set if this line defines a new height
- 2) Paragraph end bit (next to sign bit); set if the end of this line ends the paragraph
- 3) Number of characters in the line (low-order 14 bits).

If the line change bit is set, the descriptor is the first field of an NXHeightChange structure. Since this record is bracketed by negative short values, the breaks array can be sequentially accessed backwards and forwards.

Since the structure's first field is an NXChunk structure, NXBreakArrays can be manipulated using the functions that manage variable-sized arrays of records. See **NXChunkMalloc()** for more information.

NXCharArray

DECLARED IN appkit/Text.h

SYNOPSIS

```
NXChunk chunk;  
wchar text[1];  
NS_DEV_DOCFOR:typedef:NXCharArray;, } NXCharArray;
```

```
typedef struct _NXCharArray {
```

DESCRIPTION This structure holds holds the character array for the current line in the Text object. Since the structure's first field is an NXChunk structure, NXCharArrays can be manipulated using the functions that manage variable-sized arrays of records. See **NXChunkMalloc()** for more information.

NXCharFilterFunc

DECLARED IN appkit/Text.h

SYNOPSIS

```
(*NXCharFilterFunc)  
(unsigned short charCode,  
int flags,  
NS_DEV_DOCFOR:typedef:NXCharFilterFunc;, unsigned short charSet);
```

```
typedef unsigned short
```

DESCRIPTION The character filter function analyses each character the user enters in the Text object. See **setCharFilter:** (Text class).

NXCharMetrics

DECLARED IN appkit/afm.h

SYNOPSIS

```
short charCode;  
unsigned char numKernPairs;  
unsigned char reserved;  
float xWidth;
```

```
typedef struct {
```

```
int name;  
float bbox[4];  
int kernPairIndex;  
NS_DEV_DOCFOR:typedef:NXCharMetrics;,    } NXCharMetrics;
```

DESCRIPTION An NXCharMetrics structure stores information on a character. The fields are:

charCode	Character code, -1 if unencoded
numKernPairs	Number of kerning pairs starting with this character
xWidth	Width in x of this character
name	NameÐan index into a string table
bbox	Character bounding box
kernPairIndex	Index into NXFontMetrics.kerns array

NXChunk

DECLARED IN appkit/chunk.h

SYNOPSIS

```
short growby;  
int allocated;  
int used;  
NS_DEV_DOCFOR:typedef:NXChunk;,    } NXChunk;
```

typedef struct _NXChunk {

DESCRIPTION NXChunk structures are used to implement variable sized arrays of records. Allocation is by the given size (in bytes)Ðtypically a multiple number of records, say 10. The block of memory never shrinks, and the chunk records the current number of elements. To use NXChunks, declare a structure with an NXChunk structure as its first field. See **NXChunkMalloc()** for more information.

The fields of an NXChunk are:

growby	The increment used to enlarge the array
allocated	How many elements are currently allocated
used	How many elements are currently used

NXColorSpace

DECLARED IN appkit/graphics.h

SYNOPSIS

```
NX_CustomColorSpace = -1,  
NX_OneIsBlackColorSpace = 0,  
NX_OneIsWhiteColorSpace = 1,  
NX_RGBColorSpace = 2,  
NX_CMYKColorSpace = 5  
NS_DEV_DOCFOR:typedef:NXColorSpace;,    } NXColorSpace;
```

typedef enum _NXColorSpace {

DESCRIPTION Used to represent sample-encoding formats for a bitmap image.

NXCompositeChar

DECLARED IN appkit/afm.h

SYNOPSIS typedef struct {

int **compCharIndex**;
int **numParts**;
int **firstPartIndex**;

NS_DEV_DOCFOR:typedef:NXCompositeChar;, } **NXCompositeChar**;

DESCRIPTION An NXCompositeChar structure describes a composite character. The fields are:

compCharIndex	Index into NXFontMetrics.charMetrics
numParts	Number of parts making up this char
firstPartIndex	Index of first part in NXFontMetrics.compositeCharParts

NXCompositeCharPart

DECLARED IN appkit/afm.h

SYNOPSIS

NS_DEV_DOCFOR:typedef:NXCompositeCharPart;, typedef struct {
int **partIndex**;
float **dx**;
float **dy**;
} **NXCompositeCharPart**;

DESCRIPTION NXCompositeCharPart structures are used to describe elements of a composite character array. The fields are:

partIndex	Index into NXFontMetrics.charMetrics
dx	Displacement of part in x
dy	Displacement of part in y

NXDataLinkDisposition

DECLARED IN appkit/NXDataLink.h

SYNOPSIS typedef enum

_NXDataLinkDisposition {
 NX_LinkInDestination = 1,
 NX_LinkInSource = 2,
 NX_LinkBroken = 3

NS_DEV_DOCFOR:typedef:NXDataLinkDisposition;, } **NXDataLinkDisposition**;

DESCRIPTION Returned by NXDataLink's **disposition** method to identify a link as a destination link, a source link, or a broken link. See the NXDataLink class specification for more information on the dispositions of links.

NXDataLinkNumber

DECLARED IN appkit/NXDataLink.h

SYNOPSIS

NS_DEV_DOCFOR:typedef:NXDataLinkNumber;, typedef int **NXDataLinkNumber**;

DESCRIPTION The type returned by NXDataLink's **linkNumber** method as a persistent identifier of a destination link.

NXDataLinkUpdateMode

DECLARED IN appkit/NXDataLink.h

SYNOPSIS

typedef enum

```
_NXDataLinkUpdateMode {  
    NX_UpdateContinuously = 1,  
    NX_UpdateWhenSourceSaved = 2,  
    NX_UpdateManually = 3,  
    NX_UpdateNever = 4  
};  
NS_DEV_DOCFOR:typedef:NXDataLinkUpdateMode;, } NXDataLinkUpdateMode;
```

DESCRIPTION Used by NXDataLink's **setUpdateMode:** and **updateMode** methods to identify when a link's data is to be updated.

NXDragOperation

DECLARED IN appkit/drag.h

SYNOPSIS

typedef enum _NXDragOperation {

```
    NX_DragOperationNone = 0,  
    NX_DragOperationCopy = 1,  
    NX_DragOperationLink = 2,  
    NX_DragOperationGeneric = 4,  
    NX_DragOperationPrivate = 8,  
    NX_DragOperationAll = 15  
};  
NS_DEV_DOCFOR:typedef:NXDragOperation;, } NXDragOperation;
```

DESCRIPTION The NXDragOperation constants represent the operations that a dragging destination can perform on the data that a dragged image represents. While a dragging session is in progress, the drag operation values returned by the source and destination objects are compared to determine whether the destination object is valid, and to (automatically) set the appearance of the cursor:

- **NX_DragOperationNone.** The destination won't accept the dragged-image's data; the cursor isn't changed.
- **NX_DragOperationCopy.** The destination will copy the data; the cursor is changed to the copy cursor.
- **NX_DragOperationLink.** The destination will create some sort of link, as appropriate for the data; the cursor is changed to the link cursor.
- **NX_DragOperationGeneric.** The destination will perform a "standard" operation; the cursor is changed to the move cursor.
- **NX_DragOperationPrivate.** The source and the destination will negotiate for the data, or otherwise send special messages to each other; the cursor isn't changed.
- **NX_DragOperationAll.** This should only be used by the dragging source as the value of its drag operation mask.

See the NXDraggingDestination protocol for more information.

NXEncodedLigature

DECLARED IN `appkit/afm.h`

SYNOPSIS

typedef struct {

```

        unsigned char firstChar;
        unsigned char secondChar;
        unsigned char ligatureChar;
NS_DEV_DOCFOR:typedef:NXEncodedLigature;,    } NXEncodedLigature;

```

DESCRIPTION An NXEncodedLigature structure is used for elements of the encoded ligature array. This structure is used only for those ligatures in which all three characters are encoded. The fields are:

firstChar	Character encoding of first character
secondChar	Character encoding of second character
ligatureChar	Character encoding of ligature

NXErrorReporter

DECLARED IN `appkit/errors.h`

SYNOPSIS

```
NS_DEV_DOCFOR:typedef:NXErrorReporter;,
*errorState);
```

DESCRIPTION This is the type for a function that acts as a application's error reporter. See the description of **NXRegisterErrorReporter()** for more information.

NXFaceInfo

DECLARED IN `appkit/Font.h`

SYNOPSIS

```
typedef struct NXFaceInfo {
```

```

NXFontMetrics *fontMetrics;
int flags;
struct _fontFlags {
    unsigned int usedInDoc:1;
    unsigned int usedInPage:1;
    unsigned int usedInSheet:1;
} fontFlags;
struct _NXFaceInfo *nextFInfo;
NS_DEV_DOCFOR:typedef:NXFaceInfo;, } NXFaceInfo;

```

DESCRIPTION NXFaceInfo structures store information about a font and its usage. Its fields are:

fontMetrics	Information from the AFM file
flags	Which font information is present
fontFlags	Font usage (see below)
nextFInfo	Pointer to next record in the linked list

The fontFlags substructure records font usage so that conforming PostScript comments can be

generated for a document. Its fields are:

usedInDoc	Has the font been used in the document?
usedInPage	Has the font been used in the page?
usedInSheet	Has the font been used in the sheet? (There can be more than one page printed on a sheet of paper.)

NXFontMetrics

DECLARED IN appkit/afm.h

SYNOPSIS

typedef struct _NXFontMetrics {
 char ***formatVersion**;
 char ***name**;
 char ***fullName**;
 char ***familyName**;
 char ***weight**;
 float **italicAngle**;
 char **isFixedPitch**;
 char **isScreenFont**;
 short **screenFontSize**;
 float **fontBBox**[4];
 float **underlinePosition**;
 float **underlineThickness**;
 char ***version**;
 char ***notice**;
 char ***encodingScheme**;
 float **capHeight**;
 float **xHeight**;
 float **ascender**;
 float **descender**;
 short **hasYWidths**;
 float ***widths**;
 unsigned int **widthsLength**;
 char ***strings**;
 unsigned int **stringsLength**;
 char **hasXYKerns**;
 short ***encoding**;
 float ***yWidths**;
 NXCharMetrics ***charMetrics**;
 int **numCharMetrics**;
 NXLigature ***ligatures**;
 int **numLigatures**;
 NXEncodedLigature ***encLigatures**;
 int **numEncLigatures**;
 union {
 NXKernPair ***kernPairs**;
 NXKernXPair ***kernXPairs**;
 } **kerns**;
 int **numKernPairs**;
 NXTrackKern ***trackKerns**;
 int **numTrackKerns**;
 NXCompositeChar ***compositeChars**;
 int **numCompositeChars**;


```

        NXCompositeCharPart *compositeCharParts;
        int numCompositeCharParts;
NS_DEV_DOCFOR:typedef:NXFontMetrics;,    } NXFontMetrics;
```

DESCRIPTION The NXFontMetrics structure is used to describe a font. (See the description of **readMetrics:** in the Font class specification for more information.)

The structure's fields are:

formatVersion	Version of afm file format
name	Name of font for findfont
fullName	Full name of font
familyName	Font family name
weight	Weight of font
italicAngle	Degrees counterclockwise from vertical
isFixedPitch	Is the font monospaced?
isScreenFont	Is the font a screen font?
screenFontSize	If it is, how big is it?
fontBBox[4]	Bounding box (llx, lly, urx, ury)
underlinePosition	Distance from baseline for underlines
underlineThickness	Thickness of underline stroke
version	Version identifier
notice	Trademark or copyright
encodingScheme	Default encoding vector
capHeight	Top of `H'
xHeight	Top of `x'
ascender	Top of `d'
descender	Bottom of `p'
hasYWidths	Do any chars have non-0 y width?
widths	Character widths in x
widthsLength	
strings	Table of strings and other info
stringsLength	
hasXYKerns	Do any of the kerning pairs have nonzero dy?
encoding	256 offsets into NXCharMetrics
yWidths	Character widths in y (<i>not</i> in encoding order, but a parallel array to the NXCharMetrics array)
charMetrics	Array of NXCharMetrics
numCharMetrics	Number of elements
ligatures	Array of NXLigatures
numLigatures	Number of elements
encLigatures	Array of NXEncodedLigatures
numEncLigatures	Number of elements
kerns.kernPairs	Array of NXKernPairs
kerns.kernXPairs	Array of NXKernXPairs
numKernPairs	Number of elements
trackKerns	Array of NXTrackKerns
numTrackKerns	Number of elements
compositeChars	Array of NXCompositeChars
numCompositeChars	Number of elements
compositeCharParts	Array of NXCompositeCharParts
numCompositeCharParts	Number of elements

NXFontTraitMask

DECLARED IN appkit/FontManager.h

SYNOPSIS

NS_DEV_DOCFOR:typedef:NXFontTraitMas;;

typedef unsigned int **NXFontTraitMask**;

DESCRIPTION

A NXFontTraitMask characterizes one or more of a font's traits. It's used as an argument type for several of the methods in the FontManager class.

NXFSM

DECLARED IN

appkit/Text.h

SYNOPSIS

NS_DEV_DOCFOR:typedef:NXFSM;;

typedef struct _NXFSM {
 const struct _NXFSM ***next**;
 short **delta**;
 short **token**;
} **NXFSM**;

DESCRIPTION

NXFSM is a word definition finite-state machine transition structure used by a Text object. The fields are:

- next

Points to state to go to; NULL implies final state
- delta

If final state, this undoes lookahead
- token

If final state, negative value implies word is newline; 0 implies dark; and positive implies white space

NXHeightChange

DECLARED IN

appkit/Text.h

SYNOPSIS

NS_DEV_DOCFOR:typedef:NXHeightChange;;

typedef struct _NXHeightChange {
 NXLineDesc **lineDesc**;
 NXHeightInfo **heightInfo**;
} **NXHeightChange**;

DESCRIPTION

This structure associates line descriptors and line height information in a Text object.

NXHeightInfo

DECLARED IN

appkit/Text.h

SYNOPSIS

NS_DEV_DOCFOR:typedef:NXHeightInfo;;

typedef struct _NXHeightInfo {
 NXCoord **newHeight**;
 NXCoord **oldHeight**;
 NXLineDesc **lineDesc**;
} **NXHeightInfo**;

DESCRIPTION

This structure is used to store height information for each line of text in a Text object. The fields are

- newHeight

Line height from current position forward
- oldHeight

Height before change

NXJournalHeader

DECLARED IN appkit/NXJournaler.h

SYNOPSIS

typedef struct {
 int **version**;
 unsigned int **offsetToAppNames**;
 unsigned int **lastEventTime**;
NS_DEV_DOCFOR:typedef:NXJournalHeader;, } **NXJournalHeader**;

DESCRIPTION The NXJournalHeader type defines the header for a journaling event file. The event data begins immediately after the header.

NXKernPair

DECLARED IN appkit/afm.h

SYNOPSIS

typedef struct {
 int **secondCharIndex**;
 float **dx**;
 float **dy**;
NS_DEV_DOCFOR:typedef:NXKernPair;, } **NXKernPair**;

DESCRIPTION The NXKernPair structure describes a kerning pair element. Its fields are:

secondCharIndex	Index into NXFontMetrics.charMetrics
dx	x displacement relative to first character
dy	y displacement relative to first character

NXKernXPair

DECLARED IN appkit/afm.h

SYNOPSIS

typedef struct {
 int **secondCharIndex**;
 float **dx**;
NS_DEV_DOCFOR:typedef:NXKernXPair;, } **NXKernXPair**;

DESCRIPTION The NXKernXPair structure describes a kerning pair element. In this structure, the displacement in the y direction is assumed to be 0. The structure's fields are:

secondCharIndex	Index into NXFontMetrics.charMetrics
dx	X displacement relative to first character

NXLay

DECLARED IN appkit/Text.h

SYNOPSIS

NXCoord **x**;
NXCoord **y**;
short **offset**;
short **chars**;
id **font**;
void ***paraStyle**;
NXRun ***run**;
NXLayFlags **lFlags**;

NS_DEV_DOCFOR:

typedef:NXLay;, } **NXLay**;

typedef struct _NXLay {

DESCRIPTION

A Text object's NXLay structure represents a single sequence of text in a line and records everything needed to select or draw that piece. The fields are:

x	x coordinate of moveto
y	y coordinate of moveto
offset	Offset in line array for text
chars	Number of characters in the lay
font	Font object
parastyle	Implementation dependent style sheet information
run	Text run for this lay
lFlags	Lay flags

NXLayArray

DECLARED IN

appkit/Text.h

SYNOPSIS

NXChunk **chunk**;
NXLay **lays**[1];

NS_DEV_DOCFOR:

typedef:NXLayArray;, } **NXLayArray**;

typedef struct _NXLayArray {

DESCRIPTION

A Text object's NXLayArray structure holds the layout for the current line. Since the structure's first field is an NXChunk structure, NXLayArrays can be manipulated using the functions that manage variable-sized arrays of records. See **NXChunkMalloc()** for more information.

NXLayFlags

DECLARED IN

appkit/Text.h

SYNOPSIS

unsigned int **mustMove**:1;
unsigned int **isMoveChar**:1;

NS_DEV_DOCFOR:

typedef:NXLayFlags;, } **NXLayFlags**;

typedef struct {

DESCRIPTION

This structure records whether a text lay in a Text object needs special treatment. Its fields are:

mustMove	True if current lay follows lay with nonprinting character
isMoveChar	True if lay contains nonprinting character

NXLayInfo

DECLARED IN appkit/Text.h

SYNOPSIS

typedef struct _NXLayInfo {
 NXRect **rect**;
 NXCoord **descent**;
 NXCoord **width**;
 NXCoord **left**;
 NXCoord **right**;
 NXCoord **rightIndent**;
 NXLayArray ***lays**;
 NXWidthArray ***widths**;
 NXCharArray ***chars**;
 NXTextCache **cache**;
 NXRect ***textClipRect**;
 struct _lFlags {
 unsigned int **horizCanGrow**:1;
 unsigned int **vertCanGrow**:1;
 unsigned int **erase**:1;
 unsigned int **ping**:1;
 unsigned int **endsParagraph**:1;
 unsigned int **resetCache**:1;
 } **lFlags**;
};
NS_DEV_DOCFOR:typedef:NXLayInfo;, } **NXLayInfo**;

DESCRIPTION A Text object's NXLayInfo structure is used by the scanning and drawing functions to communicate information about lines. Its fields are:

rect	Bounds rect for current line
descent	Descent line; can be reset by the scanning function
width	Width of line
left	Coordinate visible at left side
right	Coordinate visible at right side
rightIndent	How much white space to leave at right side of line
lays	Filled with NXLay items by the scanning function
widths	Filled with character widths by the scanning function
chars	Filled with characters by the scanning function
cache	Cache of current block and run
textClipRect	If non-nil, the current clipping rectangle for drawing
lFlags.horizCanGrow	1 if the scanning function should dynamically resize x margins
lFlags.vertCanGrow	1 if the scanning function should dynamically resize y margins
lFlags.erase	Tells the drawing function whether to erase before drawing line
lFlags.ping	Tells the drawing function whether to ping the Window Server
lFlags.endsParagraph	True if this line ends the paragraph
lFlags.resetCache	Used in the scanning function to reset local caches

NXLigature

DECLARED IN appkit/afm.h

SYNOPSIS

typedef struct {
 int **firstCharIndex**;
 int **secondCharIndex**;
 int **ligatureIndex**;
};
NS_DEV_DOCFOR:typedef:NXLigature;, } **NXLigature**;

DESCRIPTION This structure correlates two characters and a ligature character. Its fields are:

firstCharIndex	Index into NXFontMetrics.charMetrics
secondCharIndex	Index into NXFontMetrics.charMetrics
ligatureIndex	Index into NXFontMetrics.charMetrics

NXLineDesc

DECLARED IN appkit/Text.h

SYNOPSIS

NS_DEV_DOCFOR:typedef:NXLineDesc;, typedef short **NXLineDesc**;

DESCRIPTION An NXLineDesc is used to identify lines in the Text object.

NXLinkEnumerationState

DECLARED IN appkit/NXDataLinkManager.h

SYNOPSIS

typedef struct {
void *a;
void *b;
NS_DEV_DOCFOR:typedef:NXLinkEnumerationState;, } **NXLinkEnumerationState**;

DESCRIPTION An **NXLinkEnumerationState** structure is prepared by NXDataLinkManager's **prepareEnumerationState:** method and then passed to the **nextLinkUsing:** method, allowing an application to retrieve the link manager's links. The contents of this structure are private.

NXMeasurementUnit

DECLARED IN appkit/PageLayout.h

SYNOPSIS

typedef enum
_NXMeasurementUnit {
NX_UnitInch,
NX_UnitCentimeter,
NX_UnitPoint,
NX_UnitPica
NS_DEV_DOCFOR:typedef:NXMeasurementUnit;, } **NXMeasurementUnit**;

DESCRIPTION These are the units of measurement that are used by the PageLayout class. They're offered to the user through the Units pop-up list in the Page Layout panel.

NXMessage

DECLARED IN appkit/Listener.h

SYNOPSIS

typedef struct _NXMessage {
msg_header_t **header**;
msg_type_t **sequenceType**;

SYNOPSIS

typedef enum {

```

    NX_RTFDErrorNone
    NX_RTFDErrorSaveAborted,
    NX_RTFDErrorUnableToWriteFile,
    NX_RTFDErrorUnableToCloseFile,
    NX_RTFDErrorUnableToCreatePackage
    NX_RTFDErrorUnableToCreateBackup,
    NX_RTFDErrorUnableToDeleteBackup,
    NX_RTFDErrorUnableToDeleteTemp,
    NX_RTFDErrorUnableToDeleteOriginal,
    NX_RTFDErrorFileDoesntExist,
    NX_RTFDErrorUnableToReadFile,
    NX_RTFDErrorInsufficientAccess,
    NX_RTFDErrorMalformedRTFD
NS_DEV_DOCFOR:typedef:NXRTFDError;, } NXRTFDError;
```

DESCRIPTION This enumeration defines the constants returned by methods that open or save RTFD documents (for example, the **openRTFDFrom:** method in the Text class). These constants divide into four group, as listed in the lists below.

No Errors

NX_RTFDErrorNone

Write Errors

- NX_RTFDErrorSaveAborted
- NX_RTFDErrorUnableToWriteFile
- NX_RTFDErrorUnableToCloseFile
- NX_RTFDErrorUnableToCreatePackage
- NX_RTFDErrorUnableToCreateBackup
- NX_RTFDErrorUnableToDeleteBackup
- NX_RTFDErrorUnableToDeleteTemp
- NX_RTFDErrorUnableToDeleteOriginal

Read Errors

- NX_RTFDErrorFileDoesntExist
- NX_RTFDErrorUnableToReadFile

Read/Write Errors

- NX_RTFDErrorInsufficientAccess
- NX_RTFDErrorMalformedRTFD

NXRun

DECLARED IN appkit/Text.h

SYNOPSIS

typedef struct _NXRun {

```

    id font;
    int chars;
    void *paraStyle;
    float textGray;
    int textRGBColor;
    unsigned char superscript;
    unsigned char subscript;
    id info;
```

NS_DEV_DOCFOR:typedef:NXRun;, } **NXRun**;

DESCRIPTION A Text object's NXRun structure represents a single sequence of text with a given format. The fields are:

font	The Font object for the run
chars	Number of characters in run
paraStyle	Implementation dependent style sheet information
textGray	Gray value of the text
textRGBColor	Text color (negative if not set)
superscript	Superscript in points
subscript	Subscript in points
info	Available for subclasses of Text
rFlags	Indicates underline, etc.

NXRunArray

DECLARED IN appkit/Text.h

SYNOPSIS typedef struct _NXRunArray {
NXChunk **chunk**;
NXRun **runs**[1];
NS_DEV_DOCFOR:typedef:NXRunArray;, } **NXRunArray**;

DESCRIPTION A Text object's NXRunArray structure holds the array of text runs. Since the structure's first field is an NXChunk structure, NXRunArrays can be manipulated using the functions that manage variable-sized arrays of records. See **NXChunkMalloc()** for more information.

NXRunFlags

DECLARED IN appkit/Text.h

SYNOPSIS typedef struct {
unsigned int **underline**:1;
unsigned int **graphic**:1;
NS_DEV_DOCFOR:typedef:NXRunFlags;, } **NXRunFlags**;

DESCRIPTION A Text object's NXRunFlags structure records whether a run contains graphics or is underlined. Its fields are:

underline	True if text is underlined
graphic	True if graphic is present

NXScreen

DECLARED IN appkit/screens.h

SYNOPSIS typedef struct _NXScreen {
int **screenNumber**;
NXRect **screenBounds**;

NXWindowDepth **depth**;
 NS_DEV_DOCFOR:typedef:NXScreen;, } **NXScreen**;

DESCRIPTION The NXScreen structure represents a screen. Its fields are:

screenNumber	A unique integer that identifies the screen
screenBounds	The screen's area, reckoned in the screen coordinate system
depth	The amount of memory the screen devotes to each pixel

NXSelPt

DECLARED IN appkit/Text.h

SYNOPSIS

```

typedef struct _NXSelPt {
    int cp;
    int line;
    NXCoord x;
    NXCoord y;
    int c1st;
    NXCoord ht;
}

```

NS_DEV_DOCFOR:typedef:NXSelPt;, } **NXSelPt**;

DESCRIPTION A Text object's NXSelPt structure represents one end of a selection. Its fields are:

cp	Character position
line	Offset of LineDesc in break table
x	x coordinate
y	y coordinate
c1st	Character position of first character on the line
ht	Line height

NXSpellCheckMode

DECLARED IN appkit/NXSpellChecker.h

SYNOPSIS

```

typedef enum {
    NX_CheckSpelling,
    NS_DEV_DOCFOR:enum:NX_CheckSpellingToEnd;,     NX_CheckSpellingToEnd,
    NS_DEV_DOCFOR:enum:NX_CheckSpellingFromStart;,     NX_CheckSpellingFromStart,
    NS_DEV_DOCFOR:enum:NX_CheckSpellingInSelection;,     NX_CheckSpellingInSelection,
    NS_DEV_DOCFOR:enum:NX_CountWords;,     NX_CountWords,
    NS_DEV_DOCFOR:enum:NX_CountWordsToEnd;,     NX_CountWordsToEnd,
    NS_DEV_DOCFOR:enum:NX_CountWordsInSelection;,     NX_CountWordsInSelection
}

```

NS_DEV_DOCFOR:typedef:NXSpellCheckMode;,} **NXSpellCheckMode**;

DESCRIPTION Used as arguments to NXSpellChecker's **checkSpelling:of:** and **checkSpelling:of:wordCount:** methods to specify the extent and nature of word checking and counting. The elements are:

NX_CheckSpelling	Checks spelling of the entire text stream
NX_CheckSpellingToEnd	Checks spelling from the current position to the end
NX_CheckSpellingFromStart	Checks spelling of the stream from top to bottom
NX_CheckSpellingInSelection	Check spelling within the selection
NX_CountWords	Counts the number of words in the entire text stream
NX_CountWordsToEnd	Counts words from the current position to the end
NX_CountWordsInSelection	Counts words in the selection

NXStreamSeekMode

DECLARED IN appkit/readOnlyTextStream.h

SYNOPSIS `typedef enum {`

NX_StreamStart,
NX_StreamCurrent,
NX_StreamEnd

```
NS_DEV_DOCFOR:typedef:NXStreamSeekMode;, } NXStreamSeekMode;
```

DESCRIPTION Used by the `NXReadOnlyTextStream` protocol during a seek on a stream. See the protocol specification for details.

NXStringOrderTable

DECLARED IN appkit/Text.h

SYNOPSIS `typedef struct {`

```
unsigned char primary[256];
unsigned char secondary[256];
unsigned char primaryCI[256];
unsigned char secondaryCI[256];
```

```
NS_DEV_DOCFOR:typedef:NXStringOrderTable;,    } NXStringOrderTable;
```

DESCRIPTION The arrays in a Text object's NXStringOrderTable structure are used for case-sensitive and case-insensitive ordering of characters. See the documentation for **NXOrderStrings()** for more information.

NXTabStop

DECLARED IN `appkit/Text.h`

SYNOPSIS `typedef struct NXTabStop {`

```
short kind;  
NXCoord x;
```

```
NS_DEV_DOCFOR:typedef:NXTabStop;, } NXTabStop;
```

DESCRIPTION This structure is used to describe a Text object's tab stops. Its fields are:

kind	Kind of tab (only NX_LEFTTAB is currently implemented)
x	x coordinate for stop

NXTextBlock

DECLARED IN appkit/Text.h

SYNOPSIS `typedef struct NXTextBlock {`

```
struct _NXTextBlock *next;
struct _NXTextBlock *prior;
```

```
struct _tbFlags {
    unsigned int malloced:1;
} tbFlags;
short chars;
wchar_t *text;
NS_DEV_DOCFOR:typedef:NXTextBlock;,    } NXTextBlock;
```

DESCRIPTION A Text object's NXTextBlock structures hold the characters of the text. Its fields are:s

next	Next block in linked list
prior	Previous block in linked list
tbFlags.mallo	True if the block was malloc'ed
chars	Number of characters in this block
text	The text in this block

NXTextCache

DECLARED IN appkit/Text.h

SYNOPSIS typedef struct _NXTextCache {

```
int curPos;
NXRun *curRun;
int runFirstPos;
NXTextBlock *curBlock;
int blockFirstPos;
```

NS_DEV_DOCFOR:typedef:NXTextCache;, } **NXTextCache**;

DESCRIPTION A Text object's NXTextCache structure describes the current text block and run. Its fields are:

curPos	Current position in text stream
curRun	Current run of text
runFirstPos	Character position of first character in current run
curBlock	Current block of text
blockFirstPos	Character position of first character in current block

NXTextFilterFunc

DECLARED IN appkit/Text.h

SYNOPSIS typedef char

```
*(*NXTextFilterFunc)
(id self,
 unsigned char *insertText,
 int *insertLength,
NS_DEV_DOCFOR:typedef:NXTextFilterFunc;,    int position);
```

DESCRIPTION A Text object's text filter function can be used to implement autoindenting and other features. See Text's **setTextFilter:** method.

NXTextFunc

DECLARED IN

appkit/Text.h

SYNOPSIS

typedef int (***NXTextFunc**)
(id *self*,
NS_DEV_DOCFOR:typedef:NXTextFunc;, NXLayoutInfo **layoutInfo*);

DESCRIPTION

This is the type for a Text object's scanning and drawing functions, as set through Text's **setScanFunc:** and **setDrawFunc:** methods.

NXTextStyle

DECLARED IN

appkit/Text.h

SYNOPSIS

typedef struct _NXTextStyle {
NXCoord **indent1st**;
NXCoord **indent2nd**;
NXCoord **lineHt**;
NXCoord **descentLine**;
short **alignment**;
short **numTabs**;
NXTabStop ***tabs**;
NS_DEV_DOCFOR:typedef:NXTextStyle;, } **NXTextStyle**;

DESCRIPTION

A Text object's NXTextStyle structure describes the text layout and tab stops. Its fields are:

indent1st	How far the first line of the paragraph is indented
indent2nd	How far the second line is indented
lineHt	Line height
descentLine	Distance to descent line from bottom of line
alignment	Alignment mode
numTabs	Number of tab stops
tabs	Array of tab stops

NXTopLevelErrorHandler

DECLARED IN

appkit/errors.h

SYNOPSIS

NS_DEV_DOCFOR:typedef:NXTopLevelErrorHandler;, typedef void
NXTopLevelErrorHandler(NXHandler **errorState*);

DESCRIPTION

This is the type for functions that act as a application's top-level error handler. See the description of **NXDefaultTopLevelErrorHandler()** for more information.

NXTrackingTimer

DECLARED IN

appkit/timer.h

SYNOPSIS

typedef struct _NXTrackingTimer {
double delay;
double period;
DPSTimedEntry te;
BOOL freeMe;

	DESCRIPTION	Encodes the depth, or amount of memory, devoted to a single pixel for a window or screen.
	wchar	
	DECLARED IN	appkit/Text.h
	SYNOPSIS	
NS_DEV_DOCFOR:	typedef:wchar;,	typedef unsigned char wchar ;
	DESCRIPTION	This is the type used for the characters within a Text object.

Symbolic Constants

Bits per Character and Integer

	DECLARED IN	appkit/nextstd.h
SYNOPSIS	NS_DEV_DOCFOR:global:NBITSCHAR;,	NBITSCHAR NBITSINT NS_DEV_DOCFOR:global:NBITSINT;,
	DESCRIPTION	These constants define the number of bits per character and the number of bits per integer, respectively.

Boolean Constants

	DECLARED IN	appkit/nextstd.h
Constant	SYNOPSIS Value	
	TRUE	1
	FALSE	0
	DESCRIPTION	These constants define boolean true and false values.

Box Borders

	DECLARED IN	appkit/Box.h
SYNOPSIS	NS_DEV_DOCFOR:global:NX_NOBORDER;,NX_NOBORDER NS_DEV_DOCFOR:global:NX_LINE;,NX_LINE NS_DEV_DOCFOR:global:NX_BEZEL;,NX_BEZEL NS_DEV_DOCFOR:global:NX_GROOVE;,NX_GROOVE	
	DESCRIPTION	These constants represent the four types of borders that can be drawn around a Box object.

Box Title Positions

DECLARED IN appkit/Box.h

SYNOPSIS NS_DEV_DOCFOR:global:NX_NOTITLE;; NX_NOTITLE
NS_DEV_DOCFOR:global:NX_ABOVETOP;; NX_ABOVETOP
 NS_DEV_DOCFOR:global:NX_ATTOP;;NX_ATTOP
 NS_DEV_DOCFOR:global:NX_BELOWTOP;;NX_BELOWTOP
 NS_DEV_DOCFOR:global:NX_ABOVEBOTTOM;;NX_ABOVEBOTTOM
 NS_DEV_DOCFOR:global:NX_ATBOTTOM;;NX_ATBOTTOM
 NS_DEV_DOCFOR:global:NX_BELOWBOTTOM;;NX_BELOWBOTTOM

DESCRIPTION These constants represent the locations where a Box's title can be placed with respect to its border. Thus, for example, NX_ABOVETOP means the title is above the top of the border, NX_ATTOP means the title breaks the top border, and so on.

Button and ButtonCell Highlight/Display Types

DECLARED IN appkit/ButtonCell.h

SYNOPSIS
NS_DEV_DOCFOR:global:NX_MOMENTARYPUSH;; NX_MOMENTARYPUSH
NS_DEV_DOCFOR:global:NX_PUSHONPUSHOFF;; NX_PUSHONPUSHOFF
 NS_DEV_DOCFOR:global:NX_TOGGLE;;NX_TOGGLE
 NS_DEV_DOCFOR:global:NX_SWITCH;;NX_SWITCH
 NS_DEV_DOCFOR:global:NX_RADIOBUTTON;;NX_RADIOBUTTON
 NS_DEV_DOCFOR:global:NX_MOMENTARYCHANGE;;NX_MOMENTARYCHANGE
 NS_DEV_DOCFOR:global:NX_ONOFF;;NX_ONOFF

DESCRIPTION These constants represent the way Buttons and ButtonCells behave when pressed, and how they display their state. See Button's **setType:** method for more information.

Button and ButtonCell Icon Positions

DECLARED IN appkit/Cell.h

SYNOPSIS NS_DEV_DOCFOR:global:NX_TITLEONLY;; NX_TITLEONLY
NS_DEV_DOCFOR:global:NX_ICONONLY;; NX_ICONONLY
 NS_DEV_DOCFOR:global:NX_ICONLEFT;;NX_ICONLEFT
 NS_DEV_DOCFOR:global:NX_ICONRIGHT;;NX_ICONRIGHT
 NS_DEV_DOCFOR:global:NX_ICONBELOW;;NX_ICONBELOW
 NS_DEV_DOCFOR:global:NX_ICONABOVE;;NX_ICONABOVE
 NS_DEV_DOCFOR:global:NX_ICONOVERLAPS;;NX_ICONOVERLAPS

DESCRIPTION These constants represent the position of a ButtonCell's icon relative to its title. See Button's **setIconPosition:** method for more information.

Cell and ButtonCell Parameters

DECLARED IN appkit/Cell.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_CELLDISABLED;; NX_CELLDISABLED
NS_DEV_DOCFOR:global:NX_CELLSTATE;; NX_CELLSTATE
NS_DEV_DOCFOR:global:NX_CELLEDITABLE;;NX_CELLEDITABLE
NS_DEV_DOCFOR:global:NX_CELLHIGHLIGHTED;;NX_CELLHIGHLIGHTED
NS_DEV_DOCFOR:global:NX_LIGHTBYCONTENTS;; NX_LIGHTBYCONTENTS
NS_DEV_DOCFOR:global:NX_LIGHTBYGRAY;;NX_LIGHTBYGRAY
NS_DEV_DOCFOR:global:NX_LIGHTBYBACKGROUND;;NX_LIGHTBYBACKGROUND
NS_DEV_DOCFOR:global:NX_ICONISKEYEQUIVALENT;;NX_ICONISKEYEQUIVALENT
NS_DEV_DOCFOR:global:NX_OVERLAPPINGICON;;NX_OVERLAPPINGICON
NS_DEV_DOCFOR:global:NX_ICONHORIZONTAL;;NX_ICONHORIZONTAL
NS_DEV_DOCFOR:global:NX_ICONLEFTORBOTTOM;;NX_ICONLEFTORBOTTOM
NS_DEV_DOCFOR:global:NX_CHANGECONTENTS;;NX_CHANGECONTENTS
NS_DEV_DOCFOR:global:NX_BUTTONINSET;;NX_BUTTONINSET

DESCRIPTION These constants represent parameters that are accessed through Cell's and ButtonCell's **setParameter:to:** and **getParameter:** methods. Only the first four constants listed above are accessible by Cell; the others apply to ButtonCells only.

Cell Data Entry Types

DECLARED IN appkit/Cell.h

SYNOPSIS NS_DEV_DOCFOR:global:NX_ANYTYPE;; NX_ANYTYPE
NS_DEV_DOCFOR:global:NX_INTTYPE;; NX_INTTYPE
NS_DEV_DOCFOR:global:NX_POSINTTYPE;;NX_POSINTTYPE
NS_DEV_DOCFOR:global:NX_FLOATTYPE;;NX_FLOATTYPE
NS_DEV_DOCFOR:global:NX_POSFLOATTYPE;;NX_POSFLOATTYPE
NS_DEV_DOCFOR:global:NX_DOUBLETYPE;;NX_DOUBLETYPE
NS_DEV_DOCFOR:global:NX_POSDOUBLETYPE;;NX_POSDOUBLETYPE

DESCRIPTION These constants represent the numeric data types that a text Cell can accept. See Cell's **setEntryType:** method for more information.

Cell Periodic Action Flag

DECLARED IN appkit/Cell.h

SYNOPSIS
NS_DEV_DOCFOR:global:NX_PERIODICMASK;; NX_PERIODICMASK

DESCRIPTION You pass this constant to Cell's **sendActionOn:** method to indicate that the Cell should send its action message periodically while the mouse is down.

Cell Types

DECLARED IN appkit/Cell.h

Constant	SYNOPSIS Cell Type	
NS_DEV_DOCFOR:global:NX_NULLCELLNo;;	NX_NULLCELL	No display

NS_DEV_DOCFOR:global:NX_TEXTCELLThe;;NX_TEXTCELL	The Cell displays text
NS_DEV_DOCFOR:global:NX_ICONCELLThe;;NX_ICONCELL	The Cell display an icon

DESCRIPTION These constants represent different types of Cell objects.

Color Panel Modes

DECLARED IN appkit/NXColorPanel.h

SYNOPSIS NS_DEV_DOCFOR:global:NX_GRAYMODE;; NX_GRAYMODE
NS_DEV_DOCFOR:global:NX_RGBMODE;; NX_RGBMODE
 NS_DEV_DOCFOR:global:NX_CMYKMODE;;NX_CMYKMODE
 NS_DEV_DOCFOR:global:NX_HSBMODE;;NX_HSBMODE
 NS_DEV_DOCFOR:global:NX_CUSTOMPALETTE;;NX_CUSTOMPALETTE
 NS_DEV_DOCFOR:global:NX_CUSTOMCOLORMODE;;NX_CUSTOMCOLORMODE
 NS_DEV_DOCFOR:global:NX_BEGINMODE;;NX_BEGINMODE

DESCRIPTION These constants represent the different Color panel modes.

Color Panel Mode Masks

DECLARED IN appkit/NXColorPanel.h

SYNOPSIS
NS_DEV_DOCFOR:global:NX_GRAYMODEMASK;; NX_GRAYMODEMASK
NS_DEV_DOCFOR:global:NX_RGBMODEMASK;; NX_RGBMODEMASK
 NS_DEV_DOCFOR:global:NX_CMYKMODEMASK;;NX_CMYKMODEMASK
 NS_DEV_DOCFOR:global:NX_HSBMODEMASK;;NX_HSBMODEMASK
 NS_DEV_DOCFOR:global:NX_CUSTOMPALETTE;;NX_CUSTOMPALETTE
 K
 NS_DEV_DOCFOR:global:NX_LISTMODEMASK;;NX_LISTMODEMASK
 NS_DEV_DOCFOR:global:NX_WHEELMODEMASK;;NX_WHEELMODEMASK
 NS_DEV_DOCFOR:global:NX_ALLMODESMASK;;NX_ALLMODESMASK

DESCRIPTION These constants provide masks for the Color panel modes.

Color Picker Insertion Order Constants

DECLARED IN appkit/NXColorPanel.h

Value	SYNOPSIS	Insertion Order
	NX_WHEEL_INSERTION	0.50
	NX_SLIDERS_INSERTION	0.51
	NX_CUSTOMPALETTE_INSERTION	0.52
	NX_LIST_INSERTION	0.53

DESCRIPTION These constants represent the insertion orders that correspond to the color pickers that are provided by the system.

Drawing Activity States

Constant	DECLARED IN	appkit/View.h	
	SYNOPSIS		
	Activity		
	NX_DRAWING	Drawing to the screen	
	NX_PRINTING	Spooling to a printer	
	NX_COPYING	Copying to a pasteboard	
	DESCRIPTION	Describes an application's current drawing activity.	

Error Base Constants

	DECLARED IN	appkit/errors.h	
	SYNOPSIS		
	NS_DEV_DOCFOR:global:NX_APPKIT_ERROR_BASE;;	NX_APPKIT_ERROR_BASE	
	NS_DEV_DOCFOR:global:NX_APP_ERROR_BASE;;		NX_APP_ERROR_BASE
	DESCRIPTION	These constants represent the base error codes for errors generated by the Application Kit and by your application. 1000 error codes are reserved for both sets of errors.	

Application Priority Levels

	DECLARED IN	appkit/Application.h	
Meaning	SYNOPSIS		
		Level	Value
	NX_BASETHRESHOLD	1	Normal execution
	NX_RUNMODALTHRESHOLD	5	An attention panel is being run
	NX_MODALRESPTHRESHOLD	10	A modal event loop is in progress
	DESCRIPTION	These constants represent the default priorities at which an application runs under the described circumstances. An application's priority setting is used to block the delivery of events that have a lesser priority value. A priority must be between 0 and 30 (inclusive).	

Events, Kit-Defined Subtypes

	DECLARED IN	appkit/Application.h	
Constant	SYNOPSIS		
	Meaning		
	NS_DEV_DOCFOR:global:NX_WINEXPOSEDA;;	NX_WINEXPOSED	A nonretained Window has been exposed
	NS_DEV_DOCFOR:global:NX_APPACT;;	NX_APPACT	The application has been activated
	NS_DEV_DOCFOR:global:NX_APPDEACT;;	NX_APPDEACT	The application has been deactivated
	NS_DEV_DOCFOR:global:NX_WINMOVEDA;;	NX_WINMOVED	A Window has moved
	NS_DEV_DOCFOR:global:NX_SCREENCHANGEDA;;	NX_SCREENCHANGED	A Window has changed screens
	DESCRIPTION	These represent events that are manufactured by the Application Kit.	

Events, System-Defined Subtype

Constant	DECLARED IN	appkit/Application.h	
	SYNOPSIS		
	Meaning		
		NX_POWEROFF	The user is turning off the computer
	DESCRIPTION	These represent events that are produced by the user's actions on the system.	

Figure Space Constant

	DECLARED IN	appkit/Font.h	
SYNOPSIS		NS_DEV_DOCFOR:global:NX_FIGSPACE;, NX_FIGSPACE	
	DESCRIPTION	This constant identifies the nonbreaking space character in the NEXTSTEP encoding vector.	

Font Attribute Constants

	DECLARED IN	appkit/afm.h	
	SYNOPSIS		
		NS_DEV_DOCFOR:global:NX_FONTHEADER;, NX_FONTHEADER	
		NS_DEV_DOCFOR:global:NX_FONTMETRICS;, NX_FONTMETRICS	
		NS_DEV_DOCFOR:global:NX_FONTWIDTHS;,NX_FONTWIDTHS	
		NS_DEV_DOCFOR:global:NX_FONTCHARDATA;,NX_FONTCHARDATA	
		NS_DEV_DOCFOR:global:NX_FONTKERNING;,NX_FONTKERNING	
		NS_DEV_DOCFOR:global:NX_FONTCOMPOSITES;,NX_FONTCOMPOSITES	
	DESCRIPTION	The Font class uses these constants to query the Window Server for font attributes. See the description of readMetrics: in the Font class specification.	

Font Conversion Constants

	DECLARED IN	appkit/FontManager.h	
Change	SYNOPSIS		
	Value		
		NX_NOFONTCHANGE	0
		NX_VIAPANEL	1
		NX_ADDTRAIT	2
		NX_SIZEUP	3
		NX_SIZEDOWN	4
		NX_HEAVIER	5
		NX_LIGHTER	6
		NX_REMOVETRAIT	7
			Type of

DESCRIPTION These constants are used as values of a FontManager's **whatToDo** instance variable. The value of this variable determines how the FontManager will convert a font when it receives a **convertFont:** message. (See the description of the FontManager's **convertFont:** method for more information.)

Font Matrix Constants

DECLARED IN appkit/Font.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_IDENTITYMATRIX;, NX_IDENTITYMATRIX
NS_DEV_DOCFOR:global:NX_FLIPPEDMATRIX;, NX_FLIPPEDMATRIX

DESCRIPTION These constants identify the orientation of the font. NX_IDENTITYMATRIX identifies a font matrix that's used for fonts that will be displayed in a View having an unflipped coordinate system. If the View has a flipped coordinate system (as is found in a Text object), use NX_FLIPPEDMATRIX.

Font Trait Constants

DECLARED IN appkit/FontManager.h

SYNOPSIS NS_DEV_DOCFOR:global:NX_ITALIC;, NX_ITALIC
NS_DEV_DOCFOR:global:NX_BOLD;, NX_BOLD
NS_DEV_DOCFOR:global:NX_UNBOLD;,NX_UNBOLD
NS_DEV_DOCFOR:global:NX_NONSTANDARDCHARSET;,NX_NONSTANDARDCHARSET
NS_DEV_DOCFOR:global:NX_NARROW;,NX_NARROW
NS_DEV_DOCFOR:global:NX_EXPANDED;,NX_EXPANDED
NS_DEV_DOCFOR:global:NX_CONDENSED;,NX_CONDENSED
NS_DEV_DOCFOR:global:NX_SMALLCAPS;,NX_SMALLCAPS
NS_DEV_DOCFOR:global:NX_POSTER;,NX_POSTER
NS_DEV_DOCFOR:global:NX_COMPRESSED;,NX_COMPRESSED

DESCRIPTION These constants are used by the FontManager to identify font traits. The list of font traits should be kept small since the more traits that are assigned to a given font, the harder it will be to map it to some other family. Some traits are mutually exclusive, such as NX_EXPANDED and NX_CONDENSED.

FontPanel View Tags

DECLARED IN appkit/FontPanel.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_FPPREVIEWFIELD;, NX_FPPREVIEWFIELD
NS_DEV_DOCFOR:global:NX_FPSIZEFIELD;, NX_FPSIZEFIELD
NS_DEV_DOCFOR:global:NX_FPREVERTBUTTON;,NX_FPREVERTBUTTON
NS_DEV_DOCFOR:global:NX_FPPREVIEWBUTTON;,NX_FPPREVIEWBUTTON
NS_DEV_DOCFOR:global:NX_FPSETBUTTON;,NX_FPSETBUTTON
NS_DEV_DOCFOR:global:NX_FPSIZETITLE;,NX_FPSIZETITLE
NS_DEV_DOCFOR:global:NX_FPCURRENTFIELD;,NX_FPCURRENTFIELD

These tags identify the View objects within a FontPanel object.

Gray Shades

DECLARED IN appkit/graphics.h

Shade	SYNOPSIS		Gray
	Value		
	NX_WHITE	1.0	
	NX_LTGRAY	2.0/3.0	
	NX_DKGRAY	1.0/3.0	
	NX_BLACK	0.0	

DESCRIPTION These constants represent the four pure (undithered) shades of gray that can be displayed on a monochrome screen.

Icon and Token Window Dimensions

DECLARED IN appkit/Window.h

Dimension	SYNOPSIS	
	Value	
	NX_ICONWIDTH	48.0
	NX_ICONHEIGHT	48.0
	NX_TOKENWIDTH	64.0
	NX_TOKENHEIGHT	64.0

DESCRIPTION These constants give the dimensions of an icon and the Window (a token-style Window) in which it's contained.

Image Representation Device Matching Constant

DECLARED IN appkit/NXImageRep.h

SYNOPSIS
NS_DEV_DOCFOR:global:NX_MATCHESDEVICE;, NX_MATCHESDEVICE

DESCRIPTION This constant is used by NXImageRep to indicate that the value of certain attributes, such as the number of colors, or bits-per-sample, will change to match the device that the image is shown on. See the NXImageRep class specification for more information.

Journaling Flag and Mask

DECLARED IN appkit/Application.h

SYNOPSIS
NS_DEV_DOCFOR:global:NX_JOURNALFLAG;, NX_JOURNALFLAG
NS_DEV_DOCFOR:global:NX_JOURNALFLAGMASK;, NX_JOURNALFLAGMASK

DESCRIPTION The flag and associated mask for setting a Window's event mask for journal events.

Journaling Listener Name

DECLARED IN appkit/NXJournaler.h

SYNOPSIS	Name	Value
NX_JOURNALREQUEST		"NXJournalerRequest"

DESCRIPTION This is the name that an Application's master journaler's Listener uses to check into the Network Name Server.

Journaling Recording Device

DECLARED IN appkit/NXJournaler.h

SYNOPSIS	NS_DEV_DOCFOR:global:NX_CODEC;;	NX_CODEC
	NS_DEV_DOCFOR:global:NX_DSP;;	NX_DSP

DESCRIPTION Used to set or return the recording device for NXJournaler's **recordDevice** and **setRecordDevice:** methods.

Journaling Status

DECLARED IN appkit/NXJournaler.h

SYNOPSIS	NS_DEV_DOCFOR:global:NX_STOPPED;;	NX_STOPPED
	NS_DEV_DOCFOR:global:NX_PLAYING;;	NX_PLAYING
	NS_DEV_DOCFOR:global:NX_RECORDING;;	NX_RECORDING
	NS_DEV_DOCFOR:global:NX_NONABORTABLEFLAG;;	NX_NONABORTABLEFLAG
	NS_DEV_DOCFOR:global:NX_NONABORTABLEMASK;;	NX_NONABORTABLEMASK

DESCRIPTION NX_STOPPED, NX_PLAYING, and NX_RECORDING are values of event status and sound status for NXJournaler's **getEventStatus:...** and **setEventStatus:...** methods. If you logically OR NX_NONABORTABLEMASK into the event status for a **setEventStatus:...** message, journaling will be made non-abortable.

Journaling Subevents

DECLARED IN appkit/NXJournaler.h

SYNOPSIS	NS_DEV_DOCFOR:global:NX_WINDRAGGED;;	NX_WINDRAGGED
	NS_DEV_DOCFOR:global:NX_MOUSELOCATION;;	NX_MOUSELOCATION
	NS_DEV_DOCFOR:global:NX_LASTJRNEVENT;;	NX_LASTJRNEVENT

DESCRIPTION Subevents of the NX_JOURNALEVENT event.

Journaling Window Encodings

DECLARED IN appkit/NXJournaler.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_KEYWINDOW;; NX_KEYWINDOW
NS_DEV_DOCFOR:global:NX_MAINWINDOW;; NX_MAINWINDOW
NS_DEV_DOCFOR:global:NX_MAINMENU;;NX_MAINMENU
NS_DEV_DOCFOR:global:NX_MOUSEDOWNWINDOW;;NX_MOUSEDOWNWINDOW
NS_DEV_DOCFOR:global:NX_APPICONWINDOW;;NX_APPICONWINDOW
NS_DEV_DOCFOR:global:NX_UNKNOWNWINDOW;;NX_UNKNOWNWINDOW

DESCRIPTION Window encodings in ^a.evt^o file used to save journaling sessions.

Listener Maximum Message Size

DECLARED IN appkit/Listener.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_MAXMESSAGE;; NX_MAXMESSAGE

DESCRIPTION The maximum size of a Speaker/Listener remote message.

Listener Maximum Parameters

DECLARED IN appkit/Listener.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_MAXMSGPARAMS;; NX_MAXMSGPARAMS

DESCRIPTION The maximum number of remote method parameters allowed in a Speaker/Listener remote message. Currently, the maximum is 20.

Listener Position Types

DECLARED IN appkit/Listener.h

Type	SYNOPSIS		Position
	Value		
	NX_TEXTPOSTYPE	0	
	NX_REGEXPRPOSTYPE	1	
	NX_LINENUMPOSTYPE	2	
	NX_CHARNUMPOSTYPE	3	
	NX_APPPOSTYPE	4	

DESCRIPTION These constants describe the acceptable values for the *posType* argument in the **msgPosition:posType:ok:** and **msgSetPosition:posType:andSelect:ok:** Speaker/Listener methods.

Listener Reserved Message Numbers

DECLARED IN appkit/Listener.h

SYNOPSIS

Message	Value
NX_SELECTORPMSG	35555
NX_SELECTORFMSG	35556
NX_RESPONSEMSG	35557
NX_ACKNOWLEDGE	35558
DESCRIPTION	Reserved values for the msg_id field in the header field of a Listener's NXMessage structure. In other words, these are reserved message numbers for the Mach messages received by a Listener.

Listener RPC Error Return Values

DECLARED IN appkit/Listener.h

SYNOPSIS
NS_DEV_DOCFOR:global:NX_INCORRECTMESSAGE;, NX_INCORRECTMESSAGE

DESCRIPTION This value is the return value for a Speaker/Listener message that is successfully sent if the selector isn't recognized on the remote side.

Listener Timeout Default

DECLARED IN appkit/Listener.h

SYNOPSIS	Number	Value
NX_SENDTIMEOUT	10000	
NX_RCVTIMEOUT	10000	

DESCRIPTION These values nominally represent the default timeout values for Speaker/Listener remote messages. However, they are generally disregarded for more reasonable values.

Mach Executable File Segment Names for Images

DECLARED IN appkit/NXImageRep.h

Constant	Segment Name
	NX_EPSSEGMENT ^a __EPS ^o
	NX_TIFFSEGMENT ^a __TIFF ^o
	NX_ICONSEGMENT ^a __ICON ^o

DESCRIPTION These constants represent the three Mach segments in which images can reside.

Matrix Selection Mode Constants

DECLARED IN appkit/Matrix.h

SYNOPSIS
NS_DEV_DOCFOR:global:NX_RADIOMODE;, NX_RADIOMODE
NS_DEV_DOCFOR:global:NX_HIGHLIGHTMODE;, NX_HIGHLIGHTMODE

NS_DEV_DOCFOR:global:NX_LISTMODE;;NX_LISTMODE
NS_DEV_DOCFOR:global:NX_TRACKMODE;;NX_TRACKMODE

DESCRIPTION These constants represent the modes of operation of a Matrix, as described in the Matrix class specification.

Modal Session Return Values

DECLARED IN appkit/Application.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_RUNSTOPPED;; NX_RUNSTOPPED
NS_DEV_DOCFOR:global:NX_RUNABORTED;; NX_RUNABORTED
NS_DEV_DOCFOR:global:NX_RUNCONTINUES;;NX_RUNCONTINUES

DESCRIPTION Return values for Application's **runModalFor:** and **runModalSession:.**

Open Panel Tag Constants

DECLARED IN appkit/OpenPanel.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_OPICONBUTTON;; NX_OPICONBUTTON
NX_OPTITLEFIELD
NX OPCANCELBUTTON
NX_OPOKBUTTON
NX_OPFORM

DESCRIPTION These constants redefine the SavePanel tag constants for the OpenPanel.

Page Layout Panel Button Tags

DECLARED IN appkit/PageLayout.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_PLICONBUTTON;;NX_PLICONBUTTON
NS_DEV_DOCFOR:global:NX_PLTITLEFIELD;; NX_PLTITLEFIELD
NS_DEV_DOCFOR:global:NX_PLPAPERSIZEBUTTON;;NX_PLPAPERSIZEBUTTON
NS_DEV_DOCFOR:global:NX_PLLAYOUTBUTTON;;NX_PLLAYOUTBUTTON
NS_DEV_DOCFOR:global:NX_PLUNITSBUTTON;;NX_PLUNITSBUTTON
NS_DEV_DOCFOR:global:NX_PLWIDTHFORM;;NX_PLWIDTHFORM
NS_DEV_DOCFOR:global:NX_PLHEIGHTFORM;;NX_PLHEIGHTFORM
NS_DEV_DOCFOR:global:NX_PLPORTLANDMATRIX;;NX_PLPORTLANDMATRIX
NS_DEV_DOCFOR:global:NX_PLSCALEFIELD;;NX_PLSCALEFIELD
NS_DEV_DOCFOR:global:NX_PLCANCELBUTTON;;NX_PLCANCELBUTTON
NS_DEV_DOCFOR:global:NX_PLOKBUTTON;;NX_PLOKBUTTON

DESCRIPTION These constants represent the tag values of the various buttons that the Page Layout panel displays.

Page Order Modes

DECLARED IN appkit/PrintInfo.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_DESCENDINGORDER;; NX_DESCENDINGORDER
NS_DEV_DOCFOR:global:NX_SPECIALORDER;; NX_SPECIALORDER
NS_DEV_DOCFOR:global:NX_ASCENDINGORDER;;NX_ASCENDINGORDER
NS_DEV_DOCFOR:global:NX_UNKNOWNORDER;;NX_UNKNOWNORDER

DESCRIPTION These constants describe the order in which pages are spooled for printing.

Page Orientation Constants

DECLARED IN appkit/PrintInfo.h

SYNOPSIS NS_DEV_DOCFOR:global:NX_PORTRAIT;; NX_PORTRAIT
NS_DEV_DOCFOR:global:NX_LANDSCAPE;; NX_LANDSCAPE

DESCRIPTION These constants represent the way a page is oriented for printing. In NX_PORTRAIT mode, the page is turned so it's higher than it is wide; NX_LANDSCAPE orients the page to be wider than high.

Pagination Modes

DECLARED IN appkit/PrintInfo.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_AUTOPAGINATION;;NX_AUTOPAGINATION
NS_DEV_DOCFOR:global:NX_FITPAGINATION;; NX_FITPAGINATION
NS_DEV_DOCFOR:global:NX_CLIPPAGINATION;;NX_CLIPPAGINATION

DESCRIPTION These constants represent the different ways in which an image is divided into pages. See the PrintInfo class specification for a fuller explanation.

Panel Button Tags

DECLARED IN appkit/Panel.h

SYNOPSIS

	Name	Value
NS_DEV_DOCFOR:global:NX_OKTAG1;;	NX_OKTAG	1
NS_DEV_DOCFOR:global:NX_CANCELTAG;;	NX_CANCELTAG	0

DESCRIPTION These constants define tags for the two buttons commonly presented by a Panel.

Panel Return Values

DECLARED IN appkit/Panel.h

SYNOPSIS

	Name	Value
NS_DEV_DOCFOR:global:NX_ALERTDEFAULT;;	NX_ALERTDEFAULT	1

NS_DEV_DOCFOR:global:NX_ALERTALTERNATE;;NX_ALERTALTERNATE	0
NS_DEV_DOCFOR:global:NX_ALERTOTHER;;NX_ALERTOTHER	-1
NS_DEV_DOCFOR:global:NX_ALERTERROR;;NX_ALERTERROR	-2

DESCRIPTION These constants define values returned by the **NXRunAlertPanel()** function and by **runModalSession:** when the modal session is run with a Panel provided by **NXGetAlertPanel()**.

Printer Table Key Length

DECLARED IN appkit/NXPrinter.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_PRINTKEYMAXLEN;; NX_PRINTKEYMAXLEN

DESCRIPTION This constant gives the maximum length of a string passed as the key to an NXPrinter printer-information table.

Printer Table States

DECLARED IN appkit/NXPrinter.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_PRINTERTABLEOK;; NX_PRINTERTABLEOK
NS_DEV_DOCFOR:global:NX_PRINTERTABLENOTFOUND;;
NX_PRINTERTABLENOTFOUND
NS_DEV_DOCFOR:global:NX_PRINTERTABLEERROR;;NX_PRINTERTABLEERROR

DESCRIPTION These constants are used to describe the state of a printer-information table stored by an NXPrinter object.

Rectangle Sides

DECLARED IN appkit/graphics.h

SYNOPSIS

Meaning

Side

NX_XMIN	Parallel to the y-axis, along the side with the smallest x values
NX_YMIN	Parallel to the x-axis, along the side with the smallest y values
NX_XMAX	Parallel to the y-axis, along the side with the greatest x values
NX_YMAX	Parallel to the x-axis, along the side with the greatest y values

DESCRIPTION These constants represent the four sides of a rectangle.

Save Panel Tag Constants

DECLARED IN appkit/SavePanel.h

SYNOPSIS

	Name	Value
NS_DEV_DOCFOR:global:NX_SPICONBUTTON150;;	NX_SPICONBUTTON	150
NS_DEV_DOCFOR:global:NX_SPTITLEFIELD151;;NX_SPTITLEFIELD		151

NS_DEV_DOCFOR:global:NX_SPBROWSER152;;NX_SPBROWSER 152
NS_DEV_DOCFOR:global:NX_SPCANCELBUTTONNX_CANCELTAG;;NX_SPCANCELBUTTON
NX_CANCELTAG
NS_DEV_DOCFOR:global:NX_SPOKBUTTONNX_OKTAG;;NX_SPOKBUTTON NX_OKTAG
NS_DEV_DOCFOR:global:NX_SPFORM155;;NX_SPFORM 155

DESCRIPTION These constants define tags for identifying views in the SavePanel.

Scroller Arrow Positions

DECLARED IN appkit/Scroller.h

SYNOPSIS	Position	Value
NS_DEV_DOCFOR:global:NX_SCROLLARROWSMAXEND0;;	NX_SCROLLARROWSMAXEND	0
NS_DEV_DOCFOR:global:NX_SCROLLARROWSMINEND1;;	NX_SCROLLARROWSMINEND	1
NS_DEV_DOCFOR:global:NX_SCROLLARROWSNONE2;;	NX_SCROLLARROWSNONE	2

DESCRIPTION These constants are used in Scroller's **setArrowsPosition:** method to set the position of the arrows within the scroller.

Scroller Part Identification Constants

DECLARED IN appkit/Scroller.h

SYNOPSIS	Part
Value	
NS_DEV_DOCFOR:global:NX_NOPART0;;	NX_NOPART 0
NS_DEV_DOCFOR:global:NX_DECPAGE1;;	NX_DECPAGE 1
NS_DEV_DOCFOR:global:NX_KNOB2;;	NX_KNOB 2
NS_DEV_DOCFOR:global:NX_INCPAGE3;;	NX_INCPAGE 3
NS_DEV_DOCFOR:global:NX_DECLINE4;;	NX_DECLINE 4
NS_DEV_DOCFOR:global:NX_INCLINE5;;	NX_INCLINE 5
NS_DEV_DOCFOR:global:NX_KNOBSLOT6;;	NX_KNOBSLOT 6
NS_DEV_DOCFOR:global:NX_JUMP6;;	NX_JUMP 6

DESCRIPTION These constants are used in Scroller's **hitPart** method to identify the part of the Scroller specified in a mouse event.

Scroller Usable Parts

DECLARED IN appkit/Scroller.h

SYNOPSIS	Usable
Parts Value	
NS_DEV_DOCFOR:global:NX_SCROLLERNOPARTS0;;	NX_SCROLLERNOPARTS 0
NS_DEV_DOCFOR:global:NX_SCROLLERONLYARROWS1;;	NX_SCROLLERONLYARROWS 1
NS_DEV_DOCFOR:global:NX_SCROLLERALLPARTS2;;	NX_SCROLLERALLPARTS 2

DESCRIPTION These constants define the usable parts of a Scroller object; see the class specification for more information.

Scroller Width and Height

DECLARED IN appkit/Scroller.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_SCROLLERWIDTH,, NX_SCROLLERWIDTH

DESCRIPTION This constant identifies the default width of a vertical Scroller and the default height of a horizontal Scroller. Currently, the constant is defined as 18.0.

Text Alignment Modes

DECLARED IN appkit/Text.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_LEFTALIGNED,, NX_LEFTALIGNED
NS_DEV_DOCFOR:global:NX_RIGHTALIGNED,, NX_RIGHTALIGNED
NS_DEV_DOCFOR:global:NX_CENTERED,,NX_CENTERED
NS_DEV_DOCFOR:global:NX_JUSTIFIED,,NX_JUSTIFIED

DESCRIPTION Used as arguments and return values for methods that specify text alignment.

Text Block Constant

DECLARED IN appkit/Text.h

SYNOPSIS NS_DEV_DOCFOR:global:NX_TEXTPER,, NX_TEXTPER

DESCRIPTION This constant identifies the number of characters to allocate for each text block in a Text object.

Text Key Constants

DECLARED IN appkit/Text.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_BACKSPACE,, NX_BACKSPACE
NS_DEV_DOCFOR:global:NX_CR,, NX_CR
NS_DEV_DOCFOR:global:NX_DELETE,,NX_DELETE
NS_DEV_DOCFOR:global:NX_BTAB,,NX_BTAB
NS_DEV_DOCFOR:global:NX_ILLEGAL,,NX_ILLEGAL
NS_DEV_DOCFOR:global:NX_RETURN,,NX_RETURN
NS_DEV_DOCFOR:global:NX_TAB,,NX_TAB
NS_DEV_DOCFOR:global:NX_BACKTAB,,NX_BACKTAB
NS_DEV_DOCFOR:global:NX_LEFT,,NX_LEFT
NS_DEV_DOCFOR:global:NX_RIGHT,,NX_RIGHT
NS_DEV_DOCFOR:global:NX_UP,,NX_UP
NS_DEV_DOCFOR:global:NX_DOWN,,NX_DOWN

DESCRIPTION These constants are used by a Text object's character filter function.

Text Tab Stop Constant

DECLARED IN appkit/Text.h

SYNOPSIS NS_DEV_DOCFOR:global:NX_LEFTTAB;; NX_LEFTTAB

DESCRIPTION This constant identifies the only type of tab currently defined for a Text object.

TIFF Compression Schemes

DECLARED IN appkit/tiff.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_TIFF_COMPRESSION_NONE;; NX_TIFF_COMPRESSION_NONE
NS_DEV_DOCFOR:global:NX_TIFF_COMPRESSION_CCITTFAX3;;
 NX_TIFF_COMPRESSION_CCITTFAX3
NS_DEV_DOCFOR:global:NX_TIFF_COMPRESSION_CCITTFAX4;;NX_TIFF_COMPRESSION_CCITTFAX4
NS_DEV_DOCFOR:global:NX_TIFF_COMPRESSION_LZW;;NX_TIFF_COMPRESSION_LZW
NS_DEV_DOCFOR:global:NX_TIFF_COMPRESSION_JPEG;;NX_TIFF_COMPRESSION_JPEG
NS_DEV_DOCFOR:global:NX_TIFF_COMPRESSION_PACKBITS;;NX_TIFF_COMPRESSION_PACKBITS

DESCRIPTION These constants represent the various TIFF (*tag image file format*) data compression schemes. See the NXBitmapImageRep class specification for their meanings.

View Autoresize Constants

DECLARED IN appkit/View.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_NOTSIZABLE;; NX_NOTSIZABLE
NS_DEV_DOCFOR:global:NX_MINXMARGINSIZABLE;;
 NX_MINXMARGINSIZABLE
NS_DEV_DOCFOR:global:NX_WIDTHSIZABLE;;NX_WIDTHSIZABLE
NS_DEV_DOCFOR:global:NX_MAXXMARGINSIZABLE;;NX_MAXXMARGINSIZABLE
NS_DEV_DOCFOR:global:NX_MINYMARGINSIZABLE;;NX_MINYMARGINSIZABLE
NS_DEV_DOCFOR:global:NX_HEIGHTSIZABLE;;NX_HEIGHTSIZABLE
NS_DEV_DOCFOR:global:NX_MAXYMARGINSIZABLE;;NX_MAXYMARGINSIZABLE

DESCRIPTION Used to describe which parts of a View (or its margins) are resized when the View's superview is resized. See the View class specification for details.

Window Button Masks

DECLARED IN appkit/Window.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_CLOSEBUTTONMASK;; NX_CLOSEBUTTONMASK
NS_DEV_DOCFOR:global:NX_MINIATURIZEBUTTONMASK;;

DESCRIPTION These determine the existence of the close button and miniaturize button in a Window's title bar. See the Window class description for more information.

Window Frame Description String Length

DECLARED IN appkit/Window.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_MAXFRAMESTRINGLENGTH;, NX_MAXFRAMESTRINGLENGTH

DESCRIPTION You use this constant to allocate a string that will contain Window frame information, as used by Window methods such as **saveFromToString:**.

Window Styles

DECLARED IN appkit/Window.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_PLAINSTYLE;, NX_PLAINSTYLE
NS_DEV_DOCFOR:global:NX_TILEDSTYLE;, NX_TILEDSTYLE
NS_DEV_DOCFOR:global:NX_MENUSTYLE;,NX_MENUSTYLE
NS_DEV_DOCFOR:global:NX_MINIWINDOWSTYLE;,NX_MINIWINDOWSTYLE
NS_DEV_DOCFOR:global:NX_MINIWORLDSTYLE;,NX_MINIWORLDSTYLE
NS_DEV_DOCFOR:global:NX_TOKENSTYLE;,NX_TOKENSTYLE
NS_DEV_DOCFOR:global:NX_RESIZEBARSTYLE;,NX_RESIZEBARSTYLE
NS_DEV_DOCFOR:global:NX_FIRSTWINSTYLE;,NX_FIRSTWINSTYLE
NS_DEV_DOCFOR:global:NX_LASTWINSTYLE;,NX_LASTWINSTYLE
NS_DEV_DOCFOR:global:NX_NUMWINSTYLES;,NX_NUMWINSTYLES

DESCRIPTION Used to describe a Window object's style. The last three constants are useful for sequencing through the list of distinct styles. See the Window class description for more information.

Window Tiers

DECLARED IN appkit/Window.h

SYNOPSIS

Value		Window
tier	NX_NORMALLEVEL	0
	NX_FLOATINGLEVEL	3
	NX_DOCKLEVEL	5
	NX_SUBMENULEVEL	10
	NX_MAINMENULEVEL	20

DESCRIPTION These constants list the window (device) tiers that are used by the Application Kit. Windows are ordered (or "layered") within tiers: The uppermost window in one tier can still be obscured by the lowest window in the next higher tier.

Workspace Name Constants

DECLARED IN appkit/Listener.h

SYNOPSIS

NS_DEV_DOCFOR:global:NX_WORKSPACEREQUEST;; NX_WORKSPACEREQUEST
NS_DEV_DOCFOR:global:NX_WORKSPACEREPLY;; NX_WORKSPACEREPLY

DESCRIPTION NX_WORKSPACEREQUEST is the name of the Workspace Manager's Listener's port; it isn't defined until an application enters the run loop. NX_WORKSPACEREPLY is private and shouldn't be meddled with.

Workspace Request Constants

DECLARED IN appkit/workspaceRequest.h

SYNOPSIS

Operation Constant	Value	File
WSM_MOVE_OPERATION	"move"	
WSM_COPY_OPERATION	"copy"	
WSM_LINK_OPERATION	"link"	
WSM_COMPRESS_OPERATION	"compress"	
WSM_DECOMPRESS_OPERATION	"decompress"	
WSM_ENCRYPT_OPERATION	"encrypt"	
WSM_DECRYPT_OPERATION	"decrypt"	
WSM_DESTROY_OPERATION	"destroy"	
WSM_RECYCLE_OPERATION	"recycle"	
WSM_DUPLICATE_OPERATION	"duplicate"	

DESCRIPTION Possible file operation arguments for the **performFileOperation:source:destination:files:options:** method. The object that responds to this method is available from Application's **workspace** method.

Global Variables

Application Object

DECLARED IN appkit/Application.h

SYNOPSIS NS_DEV_DOCFOR:global:NXApp;;, id **NXApp**;

DESCRIPTION The current application's Application object.

Break Tables

DECLARED IN appkit/Text.h

SYNOPSIS

```
NS_DEV_DOCFOR:global:NXEnglishBreakTable;;, const NXFSM *const NXEnglishBreakTable;  
NS_DEV_DOCFOR:global:NXEnglishBreakTableSize;;, const int  
    NXEnglishBreakTableSize;  
NS_DEV_DOCFOR:global:NXEnglishNoBreakTable;;,const NXFSM *const NXEnglishNoBreakTable;  
NS_DEV_DOCFOR:global:NXEnglishNoBreakTableSize;;,const int NXEnglishNoBreakTableSize;  
NS_DEV_DOCFOR:global:NXCBreakTable;;,const NXFSM *const NXCBreakTable;  
NS_DEV_DOCFOR:global:NXCBreakTableSize;;,const int NXCBreakTableSize;
```

DESCRIPTION These tables are finite state machines that determine word wrapping in a Text object.

Character Category Tables

DECLARED IN appkit/Text.h

SYNOPSIS

```
NS_DEV_DOCFOR:global:NXEnglishCharCatTable;;, const unsigned char *const NXEnglishCharCatTable;  
NS_DEV_DOCFOR:global:const;;, const unsigned char *const NXCCharCatTable;
```

DESCRIPTION These tables define the character classes used in a Text object's break and click tables.

Click Tables

DECLARED IN appkit/Text.h

SYNOPSIS

```
NS_DEV_DOCFOR:global:NXFSM;;, const NXFSM *const NXEnglishClickTable;  
NS_DEV_DOCFOR:global:NXEnglishClickTableSize;;, const int  
    NXEnglishClickTableSize;  
NS_DEV_DOCFOR:global:NXCClickTable;;,const NXFSM *const NXCClickTable;  
NS_DEV_DOCFOR:global:NXCClickTableSize;;,const int NXCClickTableSize;
```

DESCRIPTION These tables are used by a Text object as finite state machines that determine which characters are selected when the user double clicks.

Domain Name

DECLARED IN appkit/Application.h

SYNOPSIS

```
NS_DEV_DOCFOR:global:NXSystemDomainName;;, char *const NXSystemDomainName;
```

DESCRIPTION The name of the host's domain.

File Information

DECLARED IN appkit/workspaceRequest.h

SYNOPSIS

```
NS_DEV_DOCFOR:typedef:NXPlainFileType;;, NXAtom NXPlainFileType;  
NS_DEV_DOCFOR:typedef:NXDirectoryFileType;;, NXAtom NXDirectoryFileType;
```

NS_DEV_DOCFOR:typedef:NXApplicationFileType;,NXAtom **NXApplicationFileType**;
NS_DEV_DOCFOR:typedef:NXFilesystemFileType;,NXAtom **NXFilesystemFileType**;
NS_DEV_DOCFOR:typedef:NXShellCommandFileType;,NXAtom **NXShellCommandFileType**;

DESCRIPTION Values identifying a file's type using the **getInfoForFile:application:type:** method.
The object that responds to this message is available from Application's **workspace** method.

File-Name Extension for Data Links

DECLARED IN appkit/NXDataLink.h

SYNOPSIS
NS_DEV_DOCFOR:typedef:NXDataLinkFilenameExtension;, NXAtom **NXDataLinkFilenameExtension**;

DESCRIPTION The file-name suffix used for links saved to files using NXDataLink's **NXDataLinkFilenameExtension** method.

Null Object

DECLARED IN appkit/Application.h

SYNOPSIS NS_DEV_DOCFOR:global:NXNullObject;, int **NXNullObject**;

DESCRIPTION A canonical null object.

Pasteboard Names

DECLARED IN appkit/Pasteboard.h

SYNOPSIS NS_DEV_DOCFOR:global:NXGeneralPboard;,NXAtom **NXGeneralPboard**;
NS_DEV_DOCFOR:global:NXFontPboard;, NXAtom **NXFontPboard**;
NS_DEV_DOCFOR:global:NXRulerPboard;,NXAtom **NXRulerPboard**;
NS_DEV_DOCFOR:global:NXFindPboard;,NXAtom **NXFindPboard**;
NS_DEV_DOCFOR:global:NXDragPboard;,NXAtom **NXDragPboard**;

DESCRIPTION The names of the standard pasteboards. See the Pasteboard class specification introduction for more information.

Pasteboard Types

DECLARED IN appkit/Pasteboard.h

SYNOPSIS
NS_DEV_DOCFOR:global:NXAsciiPboardType;, NXAtom **NXAsciiPboardType**;
NS_DEV_DOCFOR:global:NXPostScriptPboardType;, NXAtom **NXPostScriptPboardType**;
NS_DEV_DOCFOR:global:NXTIFFPboardType;,NXAtom **NXTIFFPboardType**;
NS_DEV_DOCFOR:global:NXRTFPboardType;,NXAtom **NXRTFPboardType**;
NS_DEV_DOCFOR:global:NXFilenamePboardType;,NXAtom **NXFilenamePboardType**;
NS_DEV_DOCFOR:global:NXTabularPboardType;,NXAtom **NXTabularTextPboardType**;
NS_DEV_DOCFOR:global:NXFontPboardType;,NXAtom **NXFontPboardType**;

NS_DEV_DOCFOR:global:NXRulerPboardType;;NXAtom **NXRulerPboardType**;
NS_DEV_DOCFOR:global:NXFileContentsPboardType;;NXAtom **NXFileContentsPboardType**;
NS_DEV_DOCFOR:global:NXColorPboardType;;NXAtom **NXColorPboardType**;

DESCRIPTION Some standard pasteboard data types. See the Pasteboard class specification for more information.

Pasteboard Types

DECLARED IN appkit/NXDataLink.h

SYNOPSIS
NS_DEV_DOCFOR:global:NXDataLinkPboardType;;, NXAtom **NXDataLinkPboardType**;

DESCRIPTION A pasteboard type for copying a data link to the pasteboard. See the NXDataLink class specification for more information.

Pasteboard Types

DECLARED IN appkit/NXSelection.h

SYNOPSIS
NS_DEV_DOCFOR:global:NXSelectionPboardType;;, NXAtom **NXSelectionPboardType**;

DESCRIPTION A pasteboard type for copying selection descriptions to the pasteboard. See the NXSelection class specification for more information.

Process

DECLARED IN appkit/Application.h

SYNOPSIS NS_DEV_DOCFOR:global:NXProcessID;;, int **NXProcessID**;

DESCRIPTION The Mach process in which the current application is running.

Screen Dump Switch

DECLARED IN appkit/View.h

SYNOPSIS NS_DEV_DOCFOR:global:NXScreenDump;;, BOOL **NXScreenDump**;

DESCRIPTION If YES, objects are printed as they appear on the screen. If NO (the default), objects are printed in their default states.

Smart Cut and Paste Tables

DECLARED IN appkit/Text.h

SYNOPSIS

```
NS_DEV_DOCFOR:global:NXEnglishSmartLeftChars;;,      const unsigned char *const
NXEnglishSmartLeftChars;
NS_DEV_DOCFOR:global:NXEnglishSmartRightChars;;,      const unsigned char *const
NXEnglishSmartRightChars;
NS_DEV_DOCFOR:global:NXCSmartLeftChars;;,const unsigned char *const NXCSmartLeftChars;
NS_DEV_DOCFOR:global:NXCSmartRightChars;;,const unsigned char *const NXCSmartRightChars;
```

DESCRIPTION These arrays are suitable as arguments for a Text object's **setPreSelSmartTable:** and **setPostSelSmartTable:** methods. When the user pastes text into a Text object, if the character to the left (right) of the new word is not in the left (right) table, an extra space is added on that side.

View Drawing Status

DECLARED IN appkit/View.h

SYNOPSIS NS_DEV_DOCFOR:global:NXDrawingStatus;;, short **NXDrawingStatus;**

DESCRIPTION Encodes the current drawing status for an application. It takes one of the three values listed under “Drawing Activity States,” above.

Workspace Name

DECLARED IN appkit/Listener.h

SYNOPSIS

```
NS_DEV_DOCFOR:global:NXWorkspaceName;;,   const char *NXWorkspaceName;
NS_DEV_DOCFOR:global:NXWorkspaceReplyName;;,   const char *const
NXWorkspaceReplyName;
```

DESCRIPTION Use the Workspace name constants (listed under “Symbolic Constants”) rather than these variables.