Application Kit Functions

Rectangle Functions

Modify a rectangle

void	**NXSetRect(**NXRect **aRect*, NXCoord *x*, NXCoord *y*, NXCoord *width*, NXCoord *height***)**
void	**NXOffsetRect(**NXRect **aRect*, NXCoord *dx*, NXCoord *dy***)**
void	**NXInsetRect(**NXRect **aRect*, NXCoord *dx*, NXCoord *dy***)**
void	**NXIntegralRect(**NXRect **aRect***)**
NXRect *	**NXDivideRect(**NXRect **aRect*, NXRect **bRect*, NXCoord *slice*, int *edge***)**

Test graphic relationships

BOOL	**NXMouseInRect(**const NXPoint **aPoint*, const NXRect **aRect*, BOOL *flipped***)**
BOOL	**NXPointInRect(**const NXPoint **aPoint*, const NXRect **aRect***)**
BOOL	**NXIntersectsRect(**const NXRect **aRect*, const NXRect **bRect***)**
BOOL	**NXContainsRect(**const NXRect **aRect*, const NXRect **bRect***)**
BOOL	**NXEqualRect(**const NXRect **aRect*, const NXRect **bRect***)**
BOOL	**NXEmptyRect(**const NXRect **aRect***)**

Compute third rectangle from two rectangles

NXRect *	**NXUnionRect(**const NXRect **aRect*, NXRect **bRect***)**
NXRect *	**NXIntersectionRect(**const NXRect **aRect*, NXRect **bRect***)**

Optimize drawing

void	**NXRectClip(**const NXRect **aRect***)**
void	**NXRectClipList(**const NXRect **rects*, int *count***)**
void	**NXRectFill(**const NXRect **aRect***)**
void	**NXRectFillList(**const NXRect **rects*, int *count***)**
void	**NXRectFillListWithGrays(**const NXRect **rects*, const float **grays*, int *count***)**
void	**NXEraseRect(**const NXRect **aRect***)**
void	**NXHighlightRect(**const NXRect **aRect***)**

Draw a bordered rectangle

void	**NXDrawButton(**const NXRect **aRect*, const NXRect **clipRect***)**
void	**NXDrawGrayBezel(**const NXRect **aRect*, const NXRect **clipRect***)**
void	**NXDrawGroove(**const NXRect **aRect*, const NXRect **clipRect***)**
void	**NXDrawWhiteBezel(**const NXRect **aRect*, const NXRect **clipRect***)**
NXRect *	**NXDrawTiledRects(**NXRect **boundsRect*, const NXRect **clipRect*, const int **sides*, const float **grays*, int *count***)**
void	**NXFrameRect(**const NXRect **aRect***)**
void	**NXFrameRectWithWidth(**const NXRect **aRect*, NXCoord *frameWidth***)**

Query an NXRect structure

NXCoord **NX_X(**NXRect **aRect***)*
NXCoord **NX_Y(**NXRect **aRect***)*
NXCoord **NX_WIDTH(**NXRect **aRect***)*
NXCoord **NX_HEIGHT(**NXRect **aRect***)*
NXCoord **NX_MAXX(**NXRect **aRect***)*
NXCoord **NX_MAXY(**NXRect **aRect***)*
NXCoord **NX_MIDX(**NXRect **aRect***)*
NXCoord **NX_MIDY(**NXRect **aRect***)*

Color Functions

Specify a color value

NXColor **NXConvertRGBAToColor(**float *red*, float *green*, float *blue*, float *alpha***)**
NXColor **NXConvertCMYKAToColor(**float *cyan*, float *magenta*, float *yellow*, float *black*, float *alpha***)**
NXColor **NXConvertHSBAToColor(**float *hue*, float *saturation*, float *brightness*, float *alpha***)**
NXColor **NXConvertGrayAlphaToColor(**float *gray*, float *alpha***)**
NXColor **NXConvertRGBToColor(**float *red*, float *green*, float *blue***)**
NXColor **NXConvertCMYKToColor(**float *cyan*, float *magenta*, float *yellow*, float *black***)**
NXColor **NXConvertHSBToColor(**float *hue*, float *saturation*, float *brightness***)**
NXColor **NXConvertGrayToColor(**float *gray***)**

Convert a color value to its standard components

void **NXConvertColorToRGBA(**NXColor *color*, float **red*, float **green*, float **blue*, float **alpha***)**
void **NXConvertColorToCMYKA(**NXColor *color*, float **cyan*, float **magenta*, float **yellow*, float **black*, float **alpha***)**
void **NXConvertColorToHSBA(**NXColor *color*, float **hue*, float **saturation*, float **brightness*, float **alpha***)**
void **NXConvertColorToGrayAlpha(**NXColor *color*, float **gray*, float **alpha***)**
void **NXConvertColorToRGB(**NXColor *color*, float **red*, float **green*, float **blue***)**
void **NXConvertColorToCMYK(**NXColor *color*, float **cyan*, float **magenta*, float **yellow*, float **black***)**
void **NXConvertColorToHSB(**NXColor *color*, float **hue*, float **saturation*, float **brightness***)**
void **NXConvertColorToGray(**NXColor *color*, float **gray***)**

Modify a color by changing one of its components

NXColor **NXChangeRedComponent(**NXColor *color*, float *red***)**
NXColor **NXChangeGreenComponent(**NXColor *color*, float *green***)**
NXColor **NXChangeBlueComponent(**NXColor *color*, float *blue***)**
NXColor **NXChangeCyanComponent(**NXColor *color*, float *cyan***)**
NXColor **NXChangeMagentaComponent(**NXColor *color*, float *magenta***)**
NXColor **NXChangeYellowComponent(**NXColor *color*, float *yellow***)**
NXColor **NXChangeBlackComponent(**NXColor *color*, float *black***)**
NXColor **NXChangeHueComponent(**NXColor *color*, float *hue***)**
NXColor **NXChangeSaturationComponent(**NXColor *color*, float *saturation***)**
NXColor **NXChangeBrightnessComponent(**NXColor *color*, float *brightness***)**
NXColor **NXChangeGrayComponent(**NXColor *color*, float *gray***)**
NXColor **NXChangeAlphaComponent(**NXColor *color*, float *alpha***)**

Isolate one component of a color

float **NXRedComponent(**NXColor *color***)**
float **NXGreenComponent(**NXColor *color***)**
float **NXBlueComponent(**NXColor *color***)**
float **NXCyanComponent(**NXColor *color***)**
float **NXMagentaComponent(**NXColor *color***)**
float **NXYellowComponent(**NXColor *color***)**
float **NXBlackComponent(**NXColor *color***)**
float **NXHueComponent(**NXColor *color***)**
float **NXSaturationComponent(**NXColor *color***)**
float **NXBrightnessComponent(**NXColor *color***)**
float **NXGrayComponent(**NXColor *color***)**
float **NXAlphaComponent(**NXColor *color***)**

Test whether two colors are the same

BOOL **NXEqualColor(**NXColor *oneColor*, NXColor *anotherColor***)**

Get information about color space and window depth

NXColorSpace **NXColorSpaceFromDepth(**NXWindowDepth *depth***)**
int **NXBPSFromDepth(**NXWindowDepth *depth***)**
int **NXNumberOfColorComponents(**NXColorSpace *space***)**
BOOL **NXGetBestDepth(**NXWindowDepth **depth*, int *numColors*, int *bps***)**

Read and write a color from a typed stream

NXColor **NXReadColor(**NXTypedStream **stream***)**
void **NXWriteColor(**NXTypedStream **stream*, NXColor *color***)**

Read and write a color from a pasteboard

NXColor **NXReadColorFromPasteboard**(id *pasteboard*)
void **NXWriteColorToPasteboard(**id *pasteboard*, NXColor *color***)**

Set the current color

void **NXSetColor(**NXColor *color***)**

Reading the color at a screen position

NXColor **NXReadPixel(**const NXPoint **location***)**

Associate named colors with their color lists

const char * **NXColorListName (**NXColor *color***)**
const char * **NXColorName (**NXColor *color***)**
BOOL **NXFindColorNamed (**const char **colorList*, const char **colorName*, NXColor **color***)**

Text Functions

Filter characters entered into Text object

unsigned short **NXFieldFilter(**unsigned short *theChar*, int *flags*, unsigned short *charSet***)**
unsigned short **NXEditorFilter(**unsigned short *theChar*, int *flags,* unsigned short *charSet***)**

Calculate or draw a line of text (in Text object)

int **NXScanALine(**id *self*, NXLayInfo **layInfo***)*
int **NXDrawALine(**id *self*, NXLayInfo **layInfo***)*

Calculate font ascender, descender, and line height (in Text object)

void **NXTextFontInfo(**id *fontId*, NXCoord **ascender*, NXCoord **descender*, NXCoord **lineHeight***)*

Access Text object's word tables

void **NXReadWordTable(**NXZone **zone*, NXStream **stream*, unsigned char ***preSelSmart*, unsigned char ***postSelSmart*, unsigned char ***charCategories*, NXFSM ***wrapBreaks*, int **wrapBreaksCount*, NXFSM ***clickBreaks*, int **clickBreaksCount*, BOOL **charWrap***)*
void **NXWriteWordTable(**NXStream **stream*, const unsigned char **preSelSmart*, const unsigned char **postSelSmart*, const unsigned char **charCategories*, const NXFSM **wrapBreaks*, int *wrapBreaksCount*, const NXFSM **clickBreaks*, int *clickBreaksCount*, BOOL *charWrap***)**

Provide table-driven string ordering service

int **NXOrderStrings(**const unsigned char **string1*, const unsigned char **string2*, BOOL *caseSensitive*, int *length*, NXStringOrderTable **table***)*
NXStringOrderTable * **NXDefaultStringOrderTable(**void**)**

Imaging Functions

Copy an image

void **NXCopyBits(**int *gstate*, const NXRect **aRect*, const NXPoint **aPoint***)*
void **NXCopyBitmapFromGstate(**int *gstate*, const NXRect **srcRect*, const NXRect **destRect***)*

Render and read bitmap images

void **NXDrawBitmap**(const NXRect **rect*,int *pixelsWide*,int *pixelsHigh*,
int *bitsPerSample*, int *samplesPerPixel*, int *bitsPerPixel*,
int *bytesPerRow*, BOOL *isPlanar*,BOOL *hasAlpha*, NXColorSpace *colorSpace*, const unsigned char *const *data*[5]
void **NXReadBitmap(**const NXRect **rect*, int *pixelsWide*, int *pixelsHigh*, int *bps*, int *spp*, int *config*, int *mask*, void **data1*, void **data2*, void **data3*, void **data4*, void **data5***)**
void **NXSizeBitmap(**const NXRect **rect*, int **size*, int **pixelsWide*, int **pixelsHigh*, int **bps*, int **spp*, int **config*, int **mask***)**

Object Management Functions

Refer to objects by name

id **NXGetNamedObject(**const char ******name*, id *owner***)**
const char * **NXGetObjectName(**id *theObject***)**
int **NXNameObject(**const char ******name*, id *theObject*, id *owner***)**
int **NXUnnameObject(**const char ******name*, id *owner***)**

Get information about an application's windows

void **NXCountWindows(**int ******count***)**
void **NXWindowList(**int *size*, int *list*[]**)**

Convert local and global window numbers

void **NXConvertWinNumToGlobal(**int *winNum*, unsigned int ******globalNum***)**
void **NXConvertGlobalToWinNum(**int *globalNum*, unsigned int ******winNum***)**

Set up a pop-up list

void **NXAttachPopUpList(**id *button*, PopUpList ******popUpList***)**
id **NXCreatePopUpListButton(**PopUpList ******popUpList***)**

Create or free an attention panel

int **NXRunAlertPanel(**const char ******title*, const char ******msg,* const char ******defaultButton,* const char ******alternateButton,* const char ******otherButton, ...***)**
int **NXRunLocalizedAlertPanel(**const char ******table*, const char ******title*, const char ******msg,* const char ******defaultButton,* const char ******alternateButton,* const char ******otherButton, ...***)**
id **NXGetAlertPanel(**const char ******title*, const char ******msg,* const char ******firstButton,* const char ******alternateButton,* const char ******otherButton, ...***)**
void **NXFreeAlertPanel(**id *alertPanel***)**

Error-Handling Functions

Set and return an error handler

void **NXDefaultTopLevelErrorHandler(**NXHandler ******errorState***)**
NXTopLevelErrorHandler * **NXSetTopLevelErrorHandler(**NXTopLevelErrorHandler ******proc***)**
/* a macro */

NXTopLevelErrorHandler * **NXTopLevelErrorHandler(**void**)** /* a macro */

Manage error reporting

void **NXRegisterErrorReporter(**int *min*, int *max*, NXErrorReporter ******proc***)**
void **NXRemoveErrorReporter(**int *code***)**
void **NXReportError(**NXHandler ******errorState***)**

Write a formatted error string

void **NXLogError(**const char *_format_, ...**)**

Typed Stream Functions

Read or write NeXT-defined data types from or to a typed stream

void **NXReadPoint(**NXTypedStream *_typedStream_, NXPoint *_aPoint_**)**
void **NXWritePoint(**NXTypedStream *_typedStream_, const NXPoint *_aPoint_**)**
void **NXReadSize(**NXTypedStream *_typedStream_, NXSize *_aSize_**)**
void **NXWriteSize(**NXTypedStream *_typedStream_, const NXSize *_aSize_)
void **NXReadRect(**NXTypedStream *_typedStream_, NXRect *_aRect_**)**
void **NXWriteRect(**NXTypedStream *_typedStream_, const NXRect *_aRect_**)**

Remote Messaging Functions

Save data received in a remote message

char * **NXCopyInputData(**int _parameter_**)**
char * **NXCopyOutputData(**int _parameter_**)**

Get send rights to an application port

port_t **NXPortFromName(**const char *_name_, const char *_host_**)**
port_t **NXPortNameLookup(**const char *_name_, const char *_host_**)**

Match an Objective-C method and a receiver to a remote message

NXRemoteMethod * **NXRemoteMethodFromSel(**SEL _aSelector_, NXRemoteMethod *_methods_**)**
id **NXResponsibleDelegate(**id _aListener_, SEL _aSelector_**)**

Services Menu Functions

Determine whether an item is included in Services menus

int **NXSetServicesMenuItemEnabled(**const char *_item_, BOOL _flag_**)**
BOOL **NXIsServicesMenuItemEnabled(**const char *_item_**)**

Programatically invoke a service

BOOL **NXPerformService(**const char *_item_, Pasteboard *_pboard_**)**

Force Services menu to update based on new services

void **NXUpdateDynamicServices(**void**)**

Event Function

Access event record in event queue

NXEvent * **NXGetOrPeekEvent(**DPSContext *context*, NXEvent **anEvent*, int *mask*, double *timeout*, int *threshold*, int *peek***)**

Memory Allocation Functions

Macros to allocate memory

type-name * **NX_MALLOC(***type-name* **var*, *type-name*, int *num***)**
type-name * **NX_REALLOC(***type-name* **var*, *type-name*, int *num***)**
void **NX_FREE(**void **pointer***)**

Allocate a variable-sized array

NXChunk * **NXChunkMalloc(**int *growBy*, int *initUsed***)**
NXChunk * **NXChunkRealloc(**NXChunk **pc***)**
NXChunk * **NXChunkGrow(**NXChunk **pc*, int *newUsed***)**
NXChunk * **NXChunkCopy(**NXChunk **pc*, NXChunk **dpc***)**
NXChunk * **NXChunkZoneMalloc(**int *growBy*, int *initUsed*, NXZone **zone***)**
NXChunk * **NXChunkZoneRealloc(**NXChunk **pc*, NXZone **zone***)**
NXChunk * **NXChunkZoneGrow(**NXChunk **pc*, int *newUsed*, NXZone **zone***)**
NXChunk * **NXChunkZoneCopy(**NXChunk **pc*, NXChunk **dpc*, NXZone **zone***)**

Macros to allocate zone memory

type-name * **NX_ZONEMALLOC(**NXZone **zone*, *type-name* **var*, *type-name*, int *num***)**
type-name * **NX_ZONEREALLOC(**NXZone **zone*, *type-name* **var*, *type-name*, int *num***)**

Other Application Kit Functions

Get user's home directory and name

const char * **NXHomeDirectory(**void**)**
const char * **NXUserName(**void**)**

Synchronize the application with the Window Server

void **NXPing(**void**)**

Find dimensions of specified paper type

const NXSize * **NXFindPaperSize(**const char **paperName***)**

Play the system beep

void **NXBeep(**void**)**

Set up timer events

NXTrackingTimer * **NXBeginTimer(**NXTrackingTimer **timer*, double *delay*, double *period***)**
void **NXEndTimer(**NXTrackingTimer **timer***)**

Allow journaling during direct mouse tracking

void **NXJournalMouse(**void**)**

Set or copy current graphics state object

void **NXSetGState(**int *gstate***)**
void **NXCopyCurrentGState(**int *gstate***)**

Report user's request to abort

BOOL **NXUserAborted(**void**)**
void **NXResetUserAbort(**void**)**

Return file-related pasteboard types

NXAtom **NXCreateFileContentsPboardType(**const char **fileType***)**
NXAtom **NXCreateFilenamePboardType(**const char **filename***)**
const char * **NXGetFileType(**const char **pboardType***)**
const char ** **NXGetFileTypes(**const char *const ***pboardTypes***)**

Find unique files from a path

int **NXCompleteFilename(**char **path*, int *maxPathSize***)**

Draw a distinctive outline around linked data

void **NXFrameLinkRect(**const NXRect **aRect*, BOOL *isDestination***)**
float **NXLinkFrameThickness(**void**)**

Get the amount of memory used by the Window Server

int **NXGetWindowServerMemory(**DPSContext *context*, int **virtualMemory*, int **windowBackingMemory*, NXStream **windowDumpStream***)**

Macro to write an error message

void **NX_ASSERT(**int *exp*, char **msg***)**

Macro for debugging Display PostScript

void **NX_PSDEBUG(**void**)**