Microwindows Nano-X API Reference Manual

Generated by Doxygen 1.3

Sun May 18 21:03:05 2003

Chapter 1

Microwindows Nano-X API Module Index

1.1 Microwindows Nano-X API Modules

Here is a list of all modules:

Chapter 2

Chapter 3

Microwindows Nano-X API Page Index

3.1 Microwindows Nano-X API Related Pages

Here is a list of all related documentation pages:

Chapter 4

Microwindows Nano-X API Module Documentation

4.1 Nano-X public API

This is the API which Nano-X applications use.

Modules

- Nano-X color/palette management API.

  Functions for querying and modifying the palette on palette-based Nano-X systems.

- Nano-X cursor API.

  Functions for controlling the appearance of the mouse pointer.

- Nano-X drawing API.

  Functions for actually drawing primitive shapes on the screen.

- Nano-X events API.

  The Nano-X event mechanism.

- Nano-X font API.

  Functions for handling fonts and drawing text.

- Nano-X basic API.

  Functions to initialise and close Nano-X.

- Nano-X image file API.

  Functions to draw images from standard image file formats.

- Nano-X miscellaneous APIs.

Functions that didn't fit anywhere else.

- Nano-X region API.

  Functions for handling clipping regions - these are used for clipping drawing, and for non-rectangular windows.

- Nano-X clipboard API.

  Functions for handling the current selection on the clipboard.

- Nano-X timer API.

  Functions for handling timers and delays.

- Nano-X window API.

  Functions for handling windows on the screen.

4.1.1 Detailed Description

This is the API which Nano-X applications use.

Returns:

The color found at the specified index.

4.2.2.3 void GrGetSystemPalette (GR_PALETTE *pal*)

Retrieves the system palette and places it in the specified palette structure.

Parameters:

pal pointer to a palette structure to fill in with the system palette

4.2.2.4 void GrSetSystemPalette (GR_COUNT *first*, GR_PALETTE *pal*)

Sets the system palette to the values stored in the specified palette structure.

The values before the specified first value are not set.

Parameters:

first the first palette value to set

pal pointer to a palette structure containing the new values

4.3 Nano-X cursor API.

Functions for controlling the appearance of the mouse pointer.

Functions

- void GrSetWindowCursor (GR_WINDOW_ID wid, GR_CURSOR_ID cid)

  Specify a cursor for a window.

- GR_CURSOR_ID GrNewCursor (GR_SIZE width, GR_SIZE height, GR_COORD hotx, GR_COORD hoty, GR_COLOR foreground, GR_COLOR background, GR_BITMAP *fgbitmap, GR_BITMAP *bgbitmap)

  Creates a server-based cursor (mouse graphic) resource.

- void GrMoveCursor (GR_COORD x, GR_COORD y)

Draws the boundary of ellipse at the specified position using the specified dimensions and graphics context on the specified drawable.

- void GrFillEllipse (GR_DRAW_ID id, GR_GC_ID gc, GR_COORD x, GR_COORD y, GR_SIZE rx, GR

Draws a point using the specified graphics context at the specified position on the specified drawable.

- void GrPoints (GR_DRAW_ID id, GR_GC_ID gc, GR_COUNT count, GR_POINT pointtable)

  Draws a set of points using the specified graphics context at the positions specified by the point table on the specified drawable.

- void GrPoly (GR_DRAW_ID id, GR_GC_ID gc, GR_COUNT count, GR_POINT pointtable)

  Draws an unfilled polygon on the specified drawable using the specified graphics context.

- void GrFillPoly (GR_DRAW_ID id, GR_GC_ID gc, GR_COUNT count, GR_POINT pointtable)

  Draws a filled polygon on the specified drawable using the specified graphics context.

- void GrStretchArea (GR_DRAW_ID dstid, GR_GC_ID gc, GR_COORD dx1, GR_COORD dy1, GR_COORD dx2, GR_COORD dy2, GR_DRAW_ID srcid, GR_COORD sx1, GR_COORD sy1, GR_COORD sx2, GR_COORD sy2, unsigned long op)

  Copies a region from one drawable to another.

4.4.1 Detailed Description

Functions for actually drawing primitive shapes on the screen.

4.4.2 Function Documentation

4.4.2.1 void GrArc (GR_DRAW_ID *id*, GR_

4.4.2.12 void GrFillPoly (GR_DRAW_ID *id*, GR_GC_ID *gc*, GR_COUNT *count*, GR_POINT *pointtable*)

Draws a filled polygon on the specified drawable using the specified graphics context.

The polygon is specified by an array of point structures. The polygon is automatically closed- the last point need not be the saTe as the first in order for the polygon to be closed.

ParaTeters:

Parameters:

id the ID of the drawable to draw the line on

gc

4.4.2.34 void GrSetGCTSOffset (GR_GC_ID *gc*, int *xoff*, int *yoff*)

FIXME.

PT 157Tameters:

gc FIXME

xoff FIXME

yoff FIXME

Todo

FIXME document this

4.4.2.35 void GrSetGCUseBackground (GR_GC_ID *gc*, GR_BOOL *flag*)

Sets the flag which chooses whether or not the background colour is used when drawing

Parameters:

ep Pointer to the GR_EVENT structure to return the event in.

timeout The number of milliseconds to wait before timing out, or 0 for forever.

4.5.2.4 int GrGetTypedEvent (GR_WINDOW_ID *wid*, GR_EVENT_MASK *mask*, GR_UPDATE_TYPE *update*, GR_EVENT *ep*, GR_BOOL *block*)

Fills in the specified event structure with a copy of the next event on the queue that

4.5.2.6 void GrMainLoop (GR_

Parameters:

maxfd

4.5.2.14 void GrUnregisterInput (int *fd*)

Stop monitoring a file descriptor previously registered with GrRegisterInput().

Parameters:

fd The file descriptor to stop monitoring.

4.6.2.8 void GrSetFontRotation (GR_FONT_ID *fontid*, int *tenthdegrees*)

Changes the rotation of the specified font to the specified angle.

P8D

4.7 Nano-X basic API.

Functions to initialise and close Nano-X.

Functions

- int GrOpen (void)

  Open a connection to the graphics server.

- void GrClose (void)

  Close the graphics device.

- void GrFlush (void)

Generates a human readable error message describing what error occurred and what function it occured in, then exits.

Parameters:

ep The error event structure.

4.7.2.3 void GrGetScreenInfo (GR_SCREEN_INFO *sip*)

Fills in the specified GR_SCREEN_INFO structure.

Parameters:

sip Pointer to a GR_SCREEN_INFO structure

4.7.2.4 int GrOpen (void)

Open a connection to the graphics server.

Returns:

the fd of the connection to the server or -1 on failure

4.7.2.5 GR_FNCALLBACKEVENT GrSetErrorHandler (GR_FNCALLBACKEVENT *fncb*)

4.8 Nano-X image file API.

Functions to draw images from standard image file formats.

Functions

- void GrDrawImageToFit (GR_DRAW_ID id, GR

4.9 Nano-X miscellaneous APIs.

Functions that didn't fit anywhere else.

Functions

- void GrReqShmCmds (long shmsize)

scancode The key scan code to inject.

pressed TRUE for a key press, FALSE for a key release.

4.9.2.3 void GrInjectPointerEvent (GR_COORD *x*, GR_COORD *y*, int *button*, int *visible*)

4.10 Nano-X region API.

Functions for handling clipping regions - these are used for clipping drawing, and for non-rectangu-ar windows.

Functions

- GR_REGION_ID GrNewRegion (void)

- GR_BOOL GrEmptyRegion (GR_REGION_ID region)

  Determines whether the specified region is empty.

- GR_BOOL GrEqualRegion (GR_REGION_ID rgn1, GR_REGION_ID rgn2)

  Determines whether the specified regions are identical, and returns GR_TRUE if it is, or GR_FALSE otherwise.

- void GrOffsetRegion (GR_REGION_ID region, GR_SIZE dx, GR_SIZE dy)

Returns:

GR_TRUE if the region is empty, or GR_FALSE if it is not.

4.10.2.3 GR_BOOL GrEqualRegion (GR_REGION_

Parameters:

region The ID of the region to offset

dx The distance to offset the region by in the X axis

dy The distance to offset the region by in the Y axis

4.10.2.10 GR_BOOL GrPointInRegion (GR_REGION_ID *region*, GR_COORD *x*, GR_COORD *y*)

Tests whether the specified point is within the specified region, and then returns either True or False depending on the result.

Parameters:

region the ID of the region to examine.

x

Parameters:

gc The ID of the graphics context to set the clip mask of.

region The ID of the region to use as the clip mask, or 0 for none.

4.10.2.13 void GrSubtractRegion (GR_REGION_ID

Parameters:

dst_rgn ***rgn***

4.11 Nano-X clipboard API.

Functions for handling the current selection on the clipboard.

Functions

- void GrSetSelectionOwner (GR_WINDOW_ID wid, GR_CHAR typelist)

  Sets the current selection (otherwise known as the clipboard) ownership to the specified window.

- GR_WINDOW_ID GrGetSelectionOwner (GR_CHAR typelist)

  Finds thehcm BT3 curr inds t6t selecinds-25ecinds50(curtur245(indsit45(indsID,cm BT3)rcm BT305(indsif5(indsnospeci-)]TJ

4.11.2.2 void GrRequestClientData (GR_WINDOW_ID *wid*, GR_WINDOW_ID

4.11.2.4 void GrSetSelectionOwner (GR_WINDOW_IDen3sF438 9.963 Tf13.8190 0 Td[woiden3sFF28 9.963 Tf140.3

4.12 Nano-X timer API.

Functions for handling timers and delays.

Functions

- void GrDelay (GR_TIMEOUT msecs)

  This function suspends execution of the program for the specified number of milliseconds.

- void GrResizeWindow (GR_WINDOW_ID wid, GR_SIZE width, GR_SIZE

4.13.1 Detailed Description

Functions for handling windows on the screen.

4.13.2.12 GR_WINDOW_ID GrNewWindow (GR_

4.13.2.15 void GrReparentWindow (GR_WINDOW_ID *wid*, GR_WINDOW_ID *pwid*, GR_

Parameters:

wid the ID of the window to unmap

Chapter 5

Microwindows Nano-X API Data Structure Documentation

5.1 GR_CAL_DATA Struct Reference

Calibration data passed to GrCalcTransform.

Data Fields

- int xres

  X resolution of the screen.

- int yres

  Y resolution of the screen.

- int minx

  min raw X value

- int miny

  min raw Y values

- int maxx

  max raw X value

- int maxy

  max raw Y value

- GR_BOOL xswap

  true if the x component should be swapped

- GR_BOOL yswap

true if the y component should be swapped

5.1.1 Detailed Description

Calibration data passed to GrCalcTransform.

The documentation for this struct was generated from the following file:

- nano-X.h

5.3 GR_EVENT_BUTTON Struct Reference

Event for a mouse button pressed down or released.

Data Fields

- GR_EVENT_TYPE type

  event type

- GR_WINDOW_ID wid

  window id event delivered to

- GR_WINDOW_ID subwid

5.5 GR_EVENT_

5.8 GR_EVENT_FDINPUT Struct Reference

GrRegisterInput() event.

Data Fields

- GR_EVENT_TYPE type

  event type

5.9 GR_EVENT_GENERAL Struct Reference

General events for focus in or focus out for a window, or mouse enter or mouse exit from a window, or window unmapping or mapping, etc.

Data Fields

- GR_EVENT_TYPE type

  event type

5.11 GR_EVENT_MOUSE Struct Reference

Events for mouse motion or mouse ponition.

Data Fields

- GR_EVENT_TYPE type

  event type

- GR_WINDOW_ID wid

  window id event delivered to

- GR_WINDOW_ID subwid

  sub-window id (pointer was in)

- GR_COORD rootx

  root window x coordinate

- GR_COORD rooty

  root window y coordinate

- GR_COORD x

  window x coordinate of mouse

- GR_COORD y

  window y coordinate of mouse

- GR_BUTTON buttons

  current state of buttons

  - GR_KEYMOD modifiers

    modifiers (MWKMOD_SHIFT, etc)

5.11.1 Detailed Description

_KEYMOD modifin.963 .hTd[(•)]TJ ET 1 0 0 1 k 1-87.7221.218 -10.959 cm 1 1 1 1465.25

Events for mouse motion or mouse ponition.

Thmouse24IG BT //F40 11.e38se24IG BT /-5(ents)-250(fw10(s RG BT /) 1 1T5065 0 1 61)68

Thmouse24IG BT //F40 11.e38se24IG BT /-5(ents)-250(fw10(s RG BT /) 1 1T5065 0 1 61)68

5.12 GR_EVENT_

5.15 GR_EVENT_UPDATE Struct Reference

GR_EVENT

5.16 GR_GC_ference

Graphics55(87)context55(87)properties55(49(returned55(87)by55(87)the)]TJ ET 1 0 0 1 82.442 -28.726 cm 0 0 1 rg 0 0 1 R

- int mode

  drawing mode

- GR_REGION_ID region

  user55(87)region

- int xoff

  x55(87)of878787egion

- int yoff

  y55(87)of878787egion

- GR_FONT_ID font

  font87

- GR_COLOR foreground

- GR_COLOR background

- GR_BOOL fgispixelval

  TRUE8787eground’8787xoff

5.16.1 Detailed Description

Graphics context properties returned by the GrGetGCInfo() call.

The documentation for this struct was generated from the following file:

- nano-X.h

background color

- GR_EVENT_MASK GR eventmask

Chapter 6

Global GrGetRegionBox(GR_REGION_ID region, GR_RECT rect) FIXME check Doxygen comments from this point down.

Global GrNewPixmap(GR_SIZE width, GR_SIZE height, void pixels) FIXME