

Drawing

COLLABORATORS

	<i>TITLE :</i> Drawing		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		October 9, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Drawing	1
1.1	New Drawing Library	1
1.2	nplot	2
1.3	nboxfill	2
1.4	nbackcolour	3
1.5	ncursx	3
1.6	ncursy	3
1.7	nfrontcolour	3
1.8	nlocate	3
1.9	ndprint	4
1.10	ntextstyle	4
1.11	ndrawingfont	4
1.12	ncircle	4
1.13	ncls	5
1.14	ncopybitmap	5
1.15	ndrawingoutput	5
1.16	ndrawingmode	5
1.17	nellipse	6
1.18	nline	6
1.19	npoint	6
1.20	ntextlength	6
1.21	nobtainbestpen	7
1.22	nreleasepen	7
1.23	ndrawingrastport	7

Chapter 1

Drawing

1.1 New Drawing Library

Drawing V1.15 General Information:

- * Blitz Basic II library number : #188
- * Library size when linked to executable: 320 bytes
- * Number of commands : 17
- * Ressources automatically freed at end : Yes

NDrawingOutput() must be put before any other drawing functions. Don't forget to turn the debugger ON when developping.

Commands summary:

NBackColour
Statement

NBoxFill
Statement

NCircle
Statement

NCl
Statement

NCopyBitmap
Statement

NCursX
Function (Word)

NCursY
Function (Word)

NDPrint
Statement

NDrawingFont

Statement

NDrawingMode
Statement

NDrawingOutput
Statement

NDrawingRastPort
Statement

NEllipse
Statement

NFrontColour
Statement

NLocate
Statement

NLine
Statement

NObtainBestPen
Function (Word)

NPlot
Statement

NPoint
Function (Word)

NReleasePen
Statement

NTextLength
Function (Word)

NTextStyle
Statement

1.2 nplot

SYNTAX

NPlot (x, y)

STATEMENT

Draw a plot of the active color (set by NWColour) on the used window.

1.3 nboxfill

SYNTAX

```
NBoxFill(x1, y1, x2, y2)
```

STATEMENT

Draw a filled box of the active color (set by `NFrontColour`) on the used output.

1.4 nbackcolour

SYNTAX

```
NBackColour(Colour)
```

STATEMENT

Set the default back colour for graphic function and text display.

1.5 ncursx

SYNTAX

```
Result.w = NCursX
```

FUNCTION

Return the actual text cursor X position in the used output.

1.6 ncursy

SYNTAX

```
Result.w = NCursY
```

FUNCTION

Return the actual text cursor Y position in the used output.

1.7 nfrontcolour

SYNTAX

```
NFrontColour(Colour)
```

STATEMENT

Set the default front colour for graphic function and text display.

1.8 nlocate

SYNTAX

```
NLocate(x, y)
```

FUNCTION

Set the text cursor to given position (for `NDPrint`).

1.9 ndprint

SYNTAX

NDPrint (String\$)

STATEMENT

Display the given string on the used output at position set by NLocate and with the colours set by NFrontColour. You can use NTextStyle to change the output style (bold, italic, underline) and NDrawingMode to change the way the text is render.

1.10 ntextstyle

SYNTAX

NTextStyle (Style)

STATEMENT

Set the style for futur text output.

The legal Style values are:

0 = Nothing
1 = Underline
2 = Bold
4 = Italic

You can mix it if needed (Bold+Italic will be 2+4 = 6)

1.11 ndrawingfont

SYNTAX

NDrawingFont (FontID)

STATEMENT

Set the drawing current font to given FontID. All next text render will be done with the new specified font.

FontID MUST be an legal IntuiFont Pointer. You can use the NFont library to load any font and pass the NFontID pointer to this function.

1.12 ncircle

SYNTAX

NCircle(x, y, Radius)

STATEMENT

Draw a outlined circle at the position x,y with size of Radius.

1.13 ncls

SYNTAX

NCls (Colour)

STATEMENT

Set the entire output to the given colour.

1.14 ncopybitmap

SYNTAX

NCopyBitMap (BitMapID, SourceX, SourceY, DestX, DestY, Width, Height)

STATEMENT

Copy the specified area situated in the given BitMapID, and determined by SourceX, SourceY (x,y position in the bitmap to start the copy) and Width, Height with give the size of the copied data. DestX and DestY allow to copy the data at this position in the current output.

This function is 100% OS Friendly and clipped.

1.15 ndrawingoutput

SYNTAX

NDrawingOutput (RastPort)

STATEMENT

Set the drawing output to the specified rastport. After setting this, all the Drawing commands are rendered on this rastport. You must specified a valid rastport pointer.

You can use the following functions to get easely rastport from objects:

- + NWindowRastPort
- + NScreenRastPort
- + NBitmapRastPort

Example:

NDrawingRastPort NWindowRastPort ; All drawing will be done on the window.

1.16 ndrawingmode

SYNTAX

NDrawingMode (Mode)

STATEMENT

Change the drawing mode for graphics output:

Here is a quick list of valid modes:

```
#JAM1      = 0 : For text output, leave the background transparent.
#JAM2      = 1 : For text output, print text with background colour
#COMPLEMENT = 2 : XORred graphics
#INVERSVID = 4 : Use in conjunction with JAM2, it invers background and
                foreground colour when printing text.
```

Note: These constants are found in the AmigaLibs.res file.

1.17 nellipse

SYNTAX

```
NEllipse(x, y, RadiusX, RadiusY)
```

STATEMENT

Draw a outlined ellipse at the position x,y with size of RadiusX and RadiusY.

1.18 nline

SYNTAX

```
NLine(x1, y1, x2, y2)
```

STATEMENT

Draw a line of the active color (set by NFrontColour) on the used output.

1.19 npoint

SYNTAX

```
Colour.w = NPoint(x, y)
```

STATEMENT

Return the colour number at the coordinates (x,y) in the used output.

1.20 ntextlength

SYNTAX

```
Length.w = NTextLength(x, y)
```

STATEMENT

Return the pixel length of the given string in the current output. The main advantage of this function is you can get the real length of any strings used with any fonts (even non-proportionnal ones).

1.21 nobtainbestpen

SYNTAX

```
Result.w = NObtainBestPen(r, g, b, precision)
```

STATEMENT

Return the colour number which match the most with the given parameter. On a public screen with free colours, it will allocate a new colour with the (r,g,b) value. The precision parameter tell the command how it should be detected:

Valid values of 'precision' (follow by system constante):

```
Exact = -1 (#PRECISION_EXACT)
Image = 0 (#PRECISION_IMAGE)
Icon = 16 (#PRECISION_ICON)
Gui = 32 (#PRECISION_GUI)
```

Note: You MUST use the NReleasePen(Result) for any colours you have allocated before quitting your program, else the colours will be never released to the free colour bank of the screen (very bad).

1.22 nreleasepen

SYNTAX

```
NReleasePen #Pen
```

STATEMENT

Tell the system than this pen is no more used (locked) by our program, so other program can use it. You must call this function if you have allocated any pens with NObtainBestPen().

1.23 ndrawingrastport

SYNTAX

```
*RastPort = NDrawingRastPort
```

STATEMENT

Return the value of th current used RastPort for output (for advanced programmers).
