

Appixware Graphics ELF Reference

COPYRIGHT NOTICE ON THE VERSION 6.0 SOFTWARE
©1990 - 2010 Vistasource, Inc. All Rights Reserved.

Vistasource, Inc. prepared the information contained in this document for use by Vistasource personnel, customers, and prospects. Vistasource reserves the right to change the information in this document without prior notice. The contents herein should not be construed as a representation or warranty by Vistasource. Vistasource assumes no responsibility for any errors that may appear in this document.

The Proximity Thesauri ®
©1985 Merriam-Webster Inc.
©1988 Williams Collins Sons & Co. Ltd.
©1989 Van Dale Lexicografie bv. ©1989 Nathan. ©1989 Kruger.
©1989 Zanichelli. ©1989 International Data Education a s.
©1989 C.A. Stromber A B. ©1989 Espasa-Calpe.
©1983-1996. Proximity Technology, Inc.
All Rights Reserved.

The Proximity Linguibase And Hyphenation Systems®
©1983 Merriam-Webster Inc.
©1984, 1985, 1986, 1988, 1990 Williams Collins Sons & Co. Ltd.
©1987, 1989 Van Dale Lexicografie bv. ©1988 Munksgaard International Publishers Ltd.
©1988, 1989 International Data Education a s.
©1983-1996 Proximity Technology, Inc.
All Rights Reserved

The Applixware Graphics Filter Pack contains elements of the Generator Metafile Development Libraries (MDL/G)
©1988-1996 Henderson Software, Inc.
All Rights Reserved

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraphs (c) (1) (ii) of SFARS 252.277-7013, or in FAR 52.227-19, as applicable.

Hardware and software products mentioned herein are used for identification purposes only and may be trademarks of their respective companies.

Applixware is a registered trademark of Vistasource, Inc. Applixware, Applixware Real Time, Applixware Data, and Applixware Builder are trademarks of Vistasource, Inc.

This manual was produced using Applixware.

Printed: June 2010

GP_APPLICATION_DLG@

Creates a new Graphics Presents window

Format taskID = GP_APPLICATION_DLG@([menubarID[, windowlessFlag]
[, hooklessFlag] [,gp_data])

Arguments menubarID The number of a menubar to be associated with this window. (This argument is optional.) This number should be a number between 300 and 399.

windowlessFlag

Boolean value where TRUE indicates that no window will be displayed. FALSE is the default.

hooklessFlag

A Boolean value where TRUE indicates that no hook macro is run at startup time. If the value is FALSE, Graphics runs the hook macro configured in the Graphics preferences.

gp_data

An array of format GP_DATA@.

Description Creates a new Graphics Presents window. This window can use the default menubar or it can use the menubar associated with menubarID. The task id for the newly created graphics presents task is returned.

The optional menubarID parameter lets you load a menubar according to the task that will be performed in the Graphics window. You could even display several versions at the same time by opening several windows, each with separate menubars. To display a custom menubar, you have to first load it into memory using [SET SELECTIONS@](#).

The windowlessFlag parameter lets you perform automated tasks "in the background" without displaying the Graphics application window. Using this option, two applications can work simultaneously without interrupting each other and without invoking two separate axmain processes. This windowless Graphics application becomes a "child" of the window from which it was invoked.

Performing non-interactive tasks without a window conserves computing resources as the window will never have to be displayed.

To perform a windowless task:

- You must suppress all messages that are displayed. Otherwise, the task hangs when it tries to display them.

To suppress information displayed by INFO_MESSAGE@, use:

SUPPRESS_INFO_MESSAGES@

Suppress error messages using:

SUPPRESS_ERROR_MESSAGES@

- You cannot include any prompts using **PROMPT@**.
- Be sure to explicitly exit any windowless Graphics applications after they complete. Otherwise, the tasks remain in memory until you log out.
- It is best to make a call to **SELECT_WINDOW@** when exiting the windowless application. In this way, you guarantee that the exit command is invoked against the right window.
- Before testing a newly written macro that invokes a windowless application, it is best to first test the macro with all windows displaying in the foreground.

The `hooklessFlag` parameter determines whether a hook macro is run when the Graphics application starts. A hook macro is configured through the Graphics preferences dialog, and runs whenever you start Graphics or open a graphics file.

If you configure a hook macro to run at startup time, and you set `hooklessFlag` to `NULL`, the hook macro runs. If you configure a hook macro to run at startup time, and you set `hooklessFlag` to `TRUE`, the hook macro does not run. This is an optional parameter.

The `gp_data` parameter is an array of format **GP_DATA@**. **GP_DATA@** is defined in the ELF include file `graphic_.am`. The diagram below shows the structure of the **GP_DATA@** array.

GP_DATA@

STARTUP_MODE	STARTUP MODE is one of two strings: "Create Presentation" or "Create Presentation with Temple."
MEDIA_TYPE	Media Type is one of four strings: "on screen", "slide", "color overhead", "mono overhead".
TITLE	Title and Subtitle are the first two lines on the first slide of the presentation.
SUBTITLE	
FILENAME	FILENAME is the name of the template file to use for the presentation.
OUTLINE	OUTLINE is one of the outline files in the Presents directories. The default outline files are named gp_olin1.txt, gp_olin2.txt, and so on.

GR/GFX_ABANDON@

Closes the current graphics window

Format GR_ABANDON@()
GFX_ABANDON@(gfx)

Method [this.abandon@](#)

Arguments gfx A graphics handle.

Description Closes the current graphics window. If the contents of the window are modified, these changes are abandoned; that is, you are not given any opportunity to save the information before the window is closed.

GR/GFX_ABANDON_EDITS@

Cancels edits

Format GR_ABANDON_EDITS@()
GFX_ABANDON_EDITS@(gfx)

Method [this.abandon_edits@](#)

Arguments gfx A graphics handle.

Description Cancels any edits made to arcs, sides, pixels, or objects while in edit mode in the pixel editor. If changes were made, it invokes a dialog box warning you that edits are about to be abandoned.

To exit the pixel editor without displaying a warning dialog box, use [GR_ABANDON_PIXELS@](#).

GR/GFX_ABANDON_PIXELS@

Cancels edits without warning

Format GR_ABANDON_PIXELS@()
GFX_ABANDON_PIXELS@(gfx)

Method [this.abandon_pixels@](#)

Arguments gfx A graphics handle.

Description Cancels any edits that have been made to arcs, sides, pixels, or objects while in edit mode in the pixel editor. When GR_ABANDON_PIXELS@ is executed, any changes made to the object since the last save are ignored and editing mode is ended.

GR_ABANDON_PIXELS@ abandons edits without displaying a dialog box warning you that edits are not saved. If you want to display a warning, use [GR_ABANDON_EDITS@](#).

GR/GFX_ADD_BITMAP@

Adds a bitmap to the bitmap table

Format GR_ADD_BITMAP@(bitmapName, path[, useColorFlag])
GFX_ADD_BITMAP@(gfx, bitmapName, path, useColorFlag)

Method [this.add_bitmap@](#)(bitmapName, path[, useColorFlag])

Arguments gfx A graphics handle.
bitmapName The bitmap pattern name.
path The file containing the bitmap pattern.
useColorFlag An optional Boolean value which if set to TRUE indicates that the bitmap will be treated as a color bitmap. The default is FALSE.

GR/GFX_ADD_CURVE@

Adds curvature to a polyline

Format GR_ADD_CURVE@()
GFX_ADD_CURVE@(gfx)

Method [this.add_curve@](#)

Arguments gfx A graphics handle.

Description Changes the vertices of a polyline object from angles to curves. An object must be selected when this macro executes.

This macro is bound to Transform ® Smooth.

To change the vertices of a polyline from curves back to angles, use [GR_REMOVE_CURVE@](#).

GR/GFX_ADD_PAGE@

Adds a new page

Format GR_ADD_PAGE@()
GFX_ADD_PAGE@(gfx)

Method `this.add_page@`

Arguments `gfx` A graphics handle.

Description Adds a new page. The page number of this page will be one greater than the previous "last page". That is, the page number of the page is set by the graphics editor and cannot be set using ELF.

GR/GFX_ADD_SELECTED_TO_LAYER@

Adds selected objects to layer

Format `GR_ADD_SELECTED_TO_LAYER@(layerName)`
`GFX_ADD_SELECTED_TO_LAYER@(gfx, layerName)`

Method `this.add_selected_to_layer@(layerName)`

Arguments `gfx` A graphics handle.
`layerName` The name of the layer to which the objects will be added. If this layer does not exist, it will be created.

Description Adds all currently selected objects to a layer. Because all objects exist on layers, this macro moves objects from one layer to another.

GR/GFX_ALIGN@

Aligns selected objects according to specifications

Format `GR_ALIGN@ (widthsEqIFlag, heightsEqIFlag, horizAlign, vertAlign, size)`
`GFX_ALIGN@ (gfx, widthsEqIFlag, heightsEqIFlag, horizAlign, vertAlign, size)`

Method `this.align@ (widthsEqIFlag, heightsEqIFlag, horizAlign, vertAlign, size)`

Arguments `gfx` A graphics handle.
`widthsEqIFlag` A Boolean value that indicates if the width of all selected objects should be equal to the width of the [backmost](#) object.
If TRUE, all objects will have the same width as the backmost object. If FALSE, object widths are not changed.

heightsEqIFlag	<p>A Boolean value that indicates if the height of all selected objects should be equal to the width of the backmost object.</p> <p>If TRUE, all objects will have the same height as the backmost object. If FALSE, object heights are not changed.</p>
horizAlign	<p>Moves selected objects horizontally. Possible values for horizAlign are:</p> <ol style="list-style-type: none"> 0 Align objects along their left sides. 1 Align objects along their centers. 2 Align objects along their right sides. 3 Align objects adjacently along left and right sides. The object with the leftmost center is the basis for alignment and is not moved. The object with the next leftmost center is moved so that its left side aligns vertically with the base object's right side. The remaining objects are aligned accordingly. 4 Center objects between the leftmost and rightmost objects. The two outer objects do not move. The inner objects are moved horizontally to align their centers midway between the outer objects. 5 Center object sides equally between the leftmost and rightmost objects. The two outer objects do not move. The inner objects are moved horizontally to align their sides equidistant from the two outside objects.
vertAlign	<p>Moves selected objects vertically to achieve the desired alignment. Possible values for vertAlign are:</p> <ol style="list-style-type: none"> 0 Align objects along their top sides. 1 Align objects along their middles. 2 Align objects along their bottom sides. 3 Align objects adjacently along top and bottom sides. The object with the topmost center is the basis for alignment and is not moved. The object with the next topmost center is moved so that its top side aligns horizontally with the base object's bottom side. The remaining objects are aligned accordingly. 4 Center objects between the topmost and bottommost objects. The two outer objects do not move. The inner objects are moved vertically to align their centers midway between the outer objects. 5 Center object sides equally between the topmost and bottommost objects. The two outer objects do not move. The inner objects are moved vertically to align their sides equidistant from the two outside objects.

size Indicates if the size of the selected objects should be adjusted. Possible values for size are:

- 0 Makes the width of all selected objects the same as their height.
- 1 Makes the height of all selected objects the same as their width.

Description Aligns selected items horizontally and/or vertically.

If an argument is set to NULL, the alignment specification for that argument is not applied to the selected objects.

GR_APPLICATION_DLG@

Creates a new Graphics window

Format taskID = GR_APPLICATION_DLG@([menubarID[, windowlessFlag]
[, hooklessFlag])

Arguments menubarID The number of a menubar to be associated with this window. (This argument is optional.) This number should be a number between 300 and 399.

windowlessFlag

Boolean value where TRUE indicates that no window will be displayed. FALSE is the default.

hooklessFlag

A Boolean value where TRUE indicates that no hook macro is run at startup time. If the value is FALSE, Graphics runs the hook macro configured in the Graphics preferences.

Description Creates a new Graphics window. This window can use the default menubar or it can use the menubar associated with menubarID. The task id for the newly created graphics task is returned.

The optional menubarID parameter lets you load a menubar according to the task that will be performed in the Graphics window. You could even display several versions at the same time by opening several windows, each with separate menubars. To display a custom menubar, you have to first load it into memory using **SET SELECTIONS@**.

The windowlessFlag parameter lets you perform automated tasks "in the background" without displaying the Graphics application window. Using this option, two applications can work simultaneously without interrupting each other and without invoking two separate axmain processes. This windowless Graphics application becomes a "child" of the window from which it was invoked.

Performing non-interactive tasks without a window conserves computing resources as the window will never have to be displayed.

To perform a windowless task:

- You must suppress all messages that are displayed. Otherwise, the task will hang when it tries to display them.

To suppress information displayed by `INFO_MESSAGE@`, use:

SUPPRESS INFO MESSAGES@

Suppress error messages using:

SUPPRESS ERROR MESSAGES@

- You cannot include any prompts using `PROMPT@`.
- Be sure to explicitly exit any windowless Graphics applications after they complete. Otherwise, the tasks remain in memory until you log out.
- It is best to make a call to **SELECT WINDOW@** when exiting the windowless application. In this way, you guarantee that the exit command is invoked against the right window.
- Before testing a newly written macro that invokes a windowless application, it is best to first test the macro with all windows displaying in the foreground.

The `hooklessFlag` parameter determines whether a hook macro is run when the Graphics application starts. A hook macro is configured through the Graphics preferences dialog, and runs whenever you start Graphics or open a graphics file.

If you configure a hook macro to run at startup time, and you set `hooklessFlag` to `NULL`, the hook macro runs. If you configure a hook macro to run at startup time, and you set `hooklessFlag` to `TRUE`, the hook macro does not run. This is an optional parameter.

`GR_APPLICATION_DLG@` is called by the * ® Graphics menu option.

GR/GFX_APPLY_ATTR@

Applies copied attributes

Format `GR_APPLY_ATTR@()`
`GFX_APPLY_ATTR@(gfx)`

Method `this.apply_attr@`

Arguments `gfx` A graphics handle.

Description Applies line, fill, and shadow attributes to an object.

Use **GR PICK UP ATTR@** to copy the following attributes from an object so they can be applied to another object:

- Line fill patterns and color

- Line width
- Type
- Joins
- Arrowheads and arrow tails
- The object fill patterns and colors
- Shadow type, offset and color

This is bound to Utilities ® Apply Attribute.

See also [GR GET ATTR@](#)

GR/GFX_ARRAY_DUP@

Transforms an object while making copies of it

Format GR_ARRAY_DUP@(num, xOffset, yOffset, xScale, yScale, angle)
 GFX_ARRAY_DUP@(gfx, num, xOffset, yOffset, xScale, yScale, angle)

Method [this.array_dup@](#)(num, xOffset, yOffset, xScale, yScale, angle)

Arguments	gfx	A graphics handle.
	num	The number of times the selection will be duplicated.
	xOffset	The horizontal distance from the previously drawn or selected object at which the new object will be drawn. A positive number indicates placement to the right of the original object. A negative number indicates placement to the left of the original object.
	yOffset	The vertical distance from the previously drawn or selected object at which the new object will be drawn. A positive number indicates placement below the original object. A negative number indicates placement above the original object.
	xScale	The percentage to scale the size of each successive copy of the original object in the horizontal direction.
	yScale	The percentage to scale the size of each successive copy of the original object in a vertical direction.
	angle	The number of degrees each object will be rotated from the original. A positive number of degrees rotates the object in a counter-clockwise direction. A negative number of degrees rotates the object in a clockwise direction.

Description Transforms an object while making copies of it.

GR/GFX_AUTO_GRID@

Sets grid snap on or off

Format GR_AUTO_GRID@()
GFX_AUTO_GRID@(gfx)

Method [this.auto_grid@](#)

Arguments gfx A graphics handle.

Description Toggles the grid snap feature on and off. When grid snap is on:

- Objects that are being drawn are moved to the intersection of grid lines.
- Objects that are being moved move in increments equal to the current grid setting.

See also [GR GRIDS@](#)
[GR GRID ?@](#)

GR/GFX_BACKSPACE_KEY@

Deletes the previous character

Format GR_BACKSPACE_KEY@()
GFX_BACKSPACE_KEY@(gfx)

Method [this.backspace_key@](#)

Arguments gfx A graphics handle.

Description If you are editing text, this macro deletes the character preceding the cursor. If text is selected, this text is removed.

See also [GR DELETE CHAR KEY@](#)
[GR DELETE PREV CHAR KEY@](#)

GR/GFX_BACK_RETURN_KEY@

Performs a RETURN

Format GR_BACK_RETURN_KEY@()
GFX_BACK_RETURN_KEY@(gfx)

Method [this.back_return_key@](#)

Arguments gfx A graphics handle.

Description Performs a RETURN within a text frame; within unbounded text, this macro moves the characters that follow the cursor above the current line.

GR/GFX_BOLD@

Makes selected text bold

Format GR_BOLD@()
GFX_BOLD@(gfx)

Method [this.bold@](#)

Arguments gfx A graphics handle.

Description If text is selected and it is not bold, sets the selected text to bold. If the selected text is bold, the bold attribute is removed from the text.

If no text is selected, this macro either sets or unsets the bold attribute text property.

GR_BOLD@ is called by the Text ® Bold menu option.

See also [GR ITALICS@](#)
[GR UNDERLINE@](#)

GR/GFX_BORDERS@

Toggles the text borders on and off

Format GR_BORDERS@()

Method this.borders@
GFX_BORDERS@(gfx)

Arguments gfx A graphics handle.

Description Displays (or removes from display) the borders around formatted text. These borders are only for display purposes and do not print.

This macro is designed primarily for interactive use and is bound to View ® Text Borders.

GR_BULLET_DEMOTE@

Demotes a bullet in Applixware Presents

Format GR_BULLET_DEMOTE@()

Description Demotes bulleted text by one level in Presents. Applixware Presents allows you to add bullet lists to your slides. These bullet lists can be displayed with different levels. For example:

- Level 1
 - Level 2
 - Level 2
 - Level 3

When you demote a bullet, you increase its indentation and decrease its text weight. This macro requires that your cursor is placed somewhere in the text that you want to demote.

See also GFX_BULLET_DEMOTE@

GR_BULLET_PROMOTE@

Promotes a bullet in Applixware Presents

Format GR_BULLET_PROMOTE@()

Description Promotes bulleted text by one level in Presents. Applixware Presents allows you to add bullet lists to your slides. These bullet lists can be displayed with different levels. For example:

- Level 1
 - Level 2
 - Level 2
 - Level 3

When you promote a bullet, you reduce its indentation and increase its text weight. This macro requires that your cursor is placed somewhere in the text that you want to promote.

See also GFX_BULLET_PROMOTE@

GE_IMPORT_?@

Imports a non-Appixware graphics file

Format GE_IMPORT_?@(filename)

Arguments filename The full path name of the graphics file being imported.

Description Every GE_IMPORT_?@ macro opens a new Graphics window into which it will inset the converted graphics file. For brevity, all the GE_IMPORT_?@ macros are listed below. The bold macros cannot be used without first obtaining a Graphics Filter Pack license from Appixware.

GE_IMPORT_CGM@	Computer Graphics Metafile
GE_IMPORT_DXF@	Document Interchange Format
GE_IMPORT_EPS@	Encapsulated Postscript
GE_IMPORT_FAX@	CCITT Group 3 Fax Format
GE_IMPORT_FAX4@	CCITT Group 4 Fax Format
GE_IMPORT_GEM@	GEM Image Format
GE_IMPORT_GIF@	CompuServe Graphics Interchange Format
GE_IMPORT_HPGL@	Hewlett-Packard Graphics Language
GE_IMPORT_ILBM@	Amiga IFF ILBM
GE_IMPORT_IM@	Appixware Bitmap Format
GE_IMPORT_JPEG@	JPEG Interchange Format

GE_IMPORT_MACPAINT@	MacPaint
GE_IMPORT_MSWINBM@	Microsoft Windows
GE_IMPORT_PBM@	X11 release's PBM Plus
GE_IMPORT_PCX@	PC Paintbrush
GE_IMPORT_PGM@	X11 PBM Plus
GE_IMPORT_PICT2@	MacDraw PICT2 file
GE_IMPORT_PICT@	MacDraw PICT file
GE_IMPORT_PPM@	X11 PBM Plus
GE_IMPORT_PPT@	wPowerpoint
GE_IMPORT_RAW@	Raw file
GE_IMPORT_SGI@	Silicon Graphics IRIS file
GE_IMPORT_SUN@	Sun Raster
GE_IMPORT_TGA@	True Vision Targa
GE_IMPORT_TIFF@	Tagged Image File Format
GE_IMPORT_WMF@	Windows Metafile
GE_IMPORT_WPG@	WordPerfect Graphics
GE_IMPORT_XBM@	Window Bitmap
GE_IMPORT_XPM@	X Window Pixmap file
GE_IMPORT_XWD@	X Window Dump

See also [FILTER ? TO GR@](#)
[FILTER GR TO ?@](#)
[GE PASTE ?@](#)
[GE PASTE TIF@](#)
[GR IMPORT ?@](#)

GE_PASTE_?@

Inserts a bitmap image

Format GE_PASTE_?@(filename)

Arguments filename The full path name of the file containing the bitmap image.

Description Pastes a bitmap image into the current Graphics document. The image is selected after it is pasted. For brevity, all the GE_PASTE_?@ macros are listed below. The bold macros cannot be used without first obtaining a Graphics Filter Packs license from Ap-
plixware.

Notice that GR_PASTE_EPS@, GR_PASTE_FAX@, and GR_PASTE_XWD@ do not adhere to the naming conventions of the rest of these macros.

GE_PASTE_CGM@	Computer Graphics Metafile
GE_PASTE_DXF@	Document Interchange Format
GR_PASTE_EPS@	Encapsulated Postscript
GR_PASTE_FAX@	CCITT Group 3 FAX Format
GE_PASTE_GEM@	GEM Image Format
GE_PASTE_GIF@	Compuserve Graphics Interchange Format
GE_PASTE_GP4@	CCITT Group 4 FAX Format
GE_PASTE_HPGL@	Hewlett-Packard Graphics Language
GE_PASTE_ILBM@	Amiga IFF ILBM
GE_PASTE_IM@	Applixware Bitmap Format
GE_PASTE_IRIS@	Silicon Graphics IRIS
GE_PASTE_JPEG@	JPEG format
GE_PASTE_MACPAINT@	MacPaint
	MacPaint
GE_PASTE_MSWINBM@	Microsoft Windows bitmap file
GE_PASTE_PBM@	X11 release's PBM Plus
GE_PASTE_PCX@	PC Paintbrush
GE_PASTE_PGM@	X11 Portable Graymap
GE_PASTE_PICT@	MacDraw
GE_PASTE_PICT2@	MacDraw PICT2 file
GE_PASTE_PPM@	X11 release's PPM Plus
GE_PASTE_PPT@	Powerpoint file
GE_PASTE_RAW@	Raw file
GE_PASTE_SUN@	Sun Raster
GE_PASTE_TGA@	True Vision Targa
GE_PASTE_TIF@	Tagged Image File Format
GE_PASTE_WMF@	Windows metafile
GE_PASTE_WPG@	WordPerfect Graphics
GE_PASTE_XBM@	Window Bitmap
GE_PASTE_XPM@	X Window Pixmap
GR_PASTE_XWD@	X Window Dump File

GE_PASTE_TIF@ includes an optional brightness argument. See that separately documented macro for details.

Because all of these macros create temporary raster files before actually pasting an image, your axTempDir directory must have enough space to accommodate the temporary files. You can change your **Alternate /tmp File Directory Pathname** option using the * ® Applixware Preferences menu option.

See also **GE_PASTE_TIF@**
FILTER GR TO ?@
FILTER ? TO GR@

[GE_IMPORT_?@](#)

[GR_IMPORT_HPGL@](#)

GE_PASTE_TIF@

Inserts a TIFF bitmap image

Format GE_PASTE_TIF@(filename[, brightness])

Arguments

filename	The full path name of the file containing the bitmap image.
brightness	The level of brightness, or density, desired for the TIFF image. (This applies only to gray-scale bitmaps being displayed on monochrome terminals.) Enter a number from 1 to 100, where 100 is the brightest value. brightness is optional.

Description GE_PASTE_TIF@ pastes a TIFF bitmap image into the current Graphics document. The image is selected after it is pasted.

All other GE_PASTE_?@ macros omit the brightness argument.

Because GE_PASTE_TIF@ creates a temporary raster file before actually pasting an image, your axTempDir directory must have enough space to accommodate the temporary file. You can change your axTempDir Preference Editor option using the * ® Ap-
plixware Preferences menu option.

See also [GE_PASTE_?@](#)

[FILTER_GR_TO_?@](#)

[FILTER_?_TO_GR@](#)

[GE_IMPORT_?@](#)

[GR_IMPORT_HPGL@](#)

GFX/GR_?DPI@

Sets the magnification from 75 DPI to 1050 DPI

Format GFX_?DPI@

GR_?DPI@()

Description GFX_?DPI@ and GR_?DPI@ macros set an image's magnification size in terms of dots (pixels) per inch (DPI). These macros are listed below.

GFX/GR_7DPI@ 1/10 ´ magnification; called by View ® Zoom ® 10%.

GFX/GR_19DPI@ ¼ ´ magnification; called by View ® Zoom ® 25%.
 GFX/GR_25DPI@ 1/3 ´ magnification.
 GFX/GR_37DPI@ ½ ´ magnification; called by View ® Zoom ® 50%.
 GFX/GR_56DPI@ ¾ ´ magnification; called by View ® Zoom ® 75%.
 GFX/GR_75DPI@ 1 ´ magnification; called by View ® Zoom ® 100%.
 GFX/GR_150DPI@ 2 ´ magnification; called by View ® Zoom ® 200%.
 GFX/GR_300DPI@ 4 ´ magnification; called by View ® Zoom ® 400%.
 GFX/GR_450DPI@ 6 ´ magnification; called by View ® Zoom ® 600%.
 GFX/GR_600DPI@ 8 ´ magnification; called by View ® Zoom ® 800%.
 GFX/GR_750DPI@ 10 ´ magnification; called by View ® Zoom ® 1000%.
 GFX/GR_900DPI@ 12 ´ magnification; called by View ® Zoom ® 1200%.
 GFX/GR_1050DPI@ 14 ´ magnification.

The different zoom factors allow you to manipulate objects (and individual pixels using the pixel editor) with as much detail as necessary.

GR/GFX_CENTER@

Centers text

Format GR_CENTER@()
 GFX_CENTER@(gfx)

Method [this.center@](#)

Arguments gfx A graphics handle.

Description If text is selected and it is not centered, sets the alignment to centered. If the selected text is already centered, the centered attribute is removed and the text will be left-justified.

If no text is selected, this macro either sets or unsets the 'centered' property.

GR_CENTER@ is called by the Keys ® Center menu option.

GR/GFX_CHANGE_CALLBACK@

Changes the callback associated with an index

Format GR_CHANGE_CALLBACK@(index, format gr_callback_info@ macroName)
GFX_CHANGE_CALLBACK@(gfx, index, format gr_callback_info@ macroName)

Method [this.change_callback@](#)(index, format gr_callback_info@ macroName)

Arguments

gfx	A graphics handle.
index	A number indicating which entry in the array of callback macros is being changed.
macroName	The name of the macro to be associated with a callback index.

Description Replaces the existing macro associated with index with macroName. When you need to the complete array of callback macros, use [GR_GET_CALLBACKS@](#). To return information on a specific index, use [GR_GET_CALLBACK@](#).

The definition of the gr_callback_info@ format is as follows:

```
format gr_callback_info@  
    macro_name,  
    trigger_on_drag,  
    trigger_on_single_click,  
    trigger_on_double_click,  
    argument_list
```

GR/GFX_CHANGE_PART@

Changes a part's name and sizing attribute

Format GR_CHANGE_PART@(name, fixedSizeFlag)
GFX_CHANGE_PART@(gfx, name, fixedSizeFlag)

Method [this.change_part@](#)(name, fixedSizeFlag)

Arguments

gfx	A graphics handle.
name	The name, a string, to assign to the selected part. If you want to change the size attribute but not the name, enter the current name as name.
fixedSizeFlag	A Boolean value that indicates whether to make the part a fixed-size part.

If TRUE, the part becomes a fixed-size part. When a fixed-size part is placed, it is drawn the same size as when it was created. If FALSE, the part is not a fixed-size part.

Description Changes the name and size attribute for the currently selected part in the parts area.

See also [GR_CREATE_PART@](#)

GR_CHART_CHANGE_CHART_TYPE@

Changes a chart's format to 'type'

Format GR_CHART_CHANGE_CHART_TYPE@(chartName, chartType)

Method [this.chart_change_chart_type@\(chartName, chartType\)](#)

Arguments

chartName	The name of a chart within the current graphics document.
chartType	The new 'chart type'. This type is one of the following strings: cat-val histogram spider strata surface val-val

See also [GR_CHART_SET_TYPE@](#)

GR/GFX_CHART_CREATE@

Creates a chart

Format GR_CHART_CREATE@(chartName[, x1[, y1[, x2[, y2]]]])

GFX_CHART_CREATE@(gfx, chartName)

Method [this.chart_create@\(chartName\[, x1\[, y1\[, x2\[, y2 \] \] \] \] \)](#)

Arguments

gfx	A graphics handle.
chartName	The name by which the chart will be known internally.
x1	The upper left horizontal position
y1	The upper left vertical position.
x2	The bottom right horizontal position.

y2 The bottom right vertical position.

Description Creates and names an internal memory structure that will contain chart data used to display a chart. The following list identifies many of the elements that are initialized. As the value to which an element is set may vary from release to release, use one of the CHART_GET macros to obtain the current values.

```
type
orientation

/* Decorations */
maj_hrz_format
min_hrz_format
maj_vrt_format
min_vrt_format
close_top
close_bot
close_lft
close_rt
maj_hrz_grids
min_hrz_grids
maj_vrt_grids
min_vrt_grids

/* Chart size and position */
extent x1
extent y1
extent x2
extent y2

/* Margins inside the extent */
margin x1
margin y1
margin x2
margin y2

/* Axis offsets */
def_x_offset
def_y_offset

/* Legend attributes */
legend disabled
legend x_space_eating
legend y_space_eating
legend text_before_sample
legend arranged_by_row
```

```

legend max_per_row_or_column
legend h_align
legend v_align
legend h_off
legend v_off
legend t_margin
legend b_margin
legend l_margin
legend r_margin
legend row_margin
legend column_margin
legend auto_margins
legend title_align
legend box_attr
legend label_attr
legend title_attr
legend title_str

/* Title, subtitle, and footers */
title
subtitle
footer
title_align
subtitle_align
footer_align
title_attr
subtitle_attr
footer_attr

/* strata info */
stacked_strata

/* bar info */
/* 0 to 500% */
bar_margin
/* -100% to 100% */
bar_overlap

/* pie info */
info start_angle

/* chart axes */
n_x_axes
n_y_axes
x_axis[0] pos
x_axis[0] pos_val

```


y_axis[0] label_attr.field_angle

See also [GR CHART DESTROY@](#)

GR/GFX_CHART_CREATE_AXIS@

Creates an axis

Format axisName = GR_CHART_CREATE_AXIS@(chartName, axisType, format
chart_axis_info@ axisInfo)

axisName = GFX_CHART_CREATE_AXIS@(gfx, chartName, axisType, format
chart_axis_info@ axisInfo)

Method axisName = this.chart_create_axis@(chartName, axisType, format chart_axis_info@
axisInfo)

Arguments

- gfx A graphics handle.
- chartName The name of a chart within the current document.
- axisType One of the following strings:
 - x Sets the X-axis as the primary axis.
 - y Sets the Y-axis as the primary axis.
- axisInfo A structure containing axis information, as described below.

Description Creates a chart's axis based on the information passed to it in the following structure. This includes the major and minor ticks, x and y label margins, and the tick margin.

format chart_axis_info@

value_type,	percent	0
	cat-val	1
	val-val	2
	histo	3
	spider	4
	strata	5
log10,	Boolean (only for value types 1 & 2)	
auto_max,	Boolean	
max_value,	Number,	
auto_min,	Boolean	
min_value,	Number,	
auto_bas,	Boolean	
bas_value,	Number	
position,	string:	"fixed"
		"fixed, primary"

		"fixed, "secondary" "floating"
value,	left	0
	right	1
	bottom	2
	top	3
	v_val	4
	h_val	5
minor_tik_type,	none	0
	inside	1
	outside	2
	across	3
minor_tik_size,	Number (length)	
auto_n_minor_tiks,	Boolean	
n_minor_tiks,	Number (increment)	
major_tik_type,	none	0
	inside	1
	outside	2
	across	3
major_tik_size,	Number (length)	
auto_n_major_tiks,	Boolean	
n_major_tiks,	Number (increment)	
format chart_number_format@	tik_format,	
label_x_margin,	Number	
label_y_margin,	Number	
arrayof tik_labels,	Category labels	
label,	String	
bar_margin,	0 to 500%, use a number	
bar_overlap,	-100% to 100%, use a number	
label_alignment,	left	0
	center	1
	right	2
tik_alignment,	left	0
	center	1
	right	2
tik_margin	Number: Margin between tick and tick label	

format chart_number_format@
 style, One of the following strings:
 Unstyled Currency
 Comma Fixed
 General Scientific

	Percentage	Date
	Boolean	Graph
	Time	Custom
aux_style_info,	date formats:	
	1	Mmmm d, yyy
	2	Mm d, yyyy
	3	d Mm yyy
	4	mm/dd/yy
	5	dd.mm.yy
	6	yyyy-mm-dd
	7	yy-mm-dd
	8	yyyy mm dd
	9	yy mm dd
	10	yyymmdd
	11	yymmdd
	12	dd/mm/yy
	13	dd.mm.yyyy
	14	Mm dd, yyyy
	15	Mmmm yyyy
	16	Mm yyyy
	17	Mm yy
	18	mm/dd
	19	yy mm
	20	yyyy mm
	time formats:	
	0	hh:mm:ss AM/PM
	1	hh:mm AM/PM
	2	hh:mm:ss
	3	hh:mm
	4	hh:mm:ss.tss AM/PM
	5	hh:mm:ss.tss
	money format:	
	0	Leading Symbol
	1	Trailing Symbol
units,	Number: the display units of tick label values. The actual values of the data are divided by the unit number to produce the displayed tick label values	
radix_places,	Number: maximum value is 9	
prefix,	String	
suffix,	String	
am_str,	String	
pm_str,	String	

true_str,	String
false_str,	String
thousands_seperator,	String
decimal_seperator,	String
display_format_errors,	Boolean
trim_setting	Boolean indicating if trailing zeros are eliminated

This macro returns the name of the created axis. The name is always of the form:

axis <x_or_y><number>

See also [GR CHART DESTROY AXIS@](#)

GR/GFX_CHART_CREATE_GROUP@

Adds a group to a chart

Format groupName = GR_CHART_CREATE_GROUP@ (chartName, [groupName], format chart_group@ inGrp)
 groupName = GFX_CHART_CREATE_GROUP@ (gfx, chartName, [groupName], format chart_group@ inGrp)

Method groupName = this.chart_create_group@ (chartName, [groupName], format chart_group@ inGrp)

Arguments

gfx	A graphics handle.
chartName	The name of a chart in the current document.
groupName	The name of the group being created. If this argument is NULL or is not a string, a name is created for you. A group name created by Graphics is of the following form: data 0 data 1 data 2 ... However, the value following data can be any user-defined string.
inGrp	The data for the group.

Description Adds a new group to a chart. The chart information is passed within a chart_group@ format. See [GR CHART SET GROUP@](#) for more information.

See also [GR_CHART_DESTROY_GROUP@](#)

GR/GFX_CHART_DESTROY@

Removes a chart

Format GR_CHART_DESTROY@(chartName)
GFX_CHART_DESTROY@(gfx, chartName)

Method [this.chart_destroy@](#)(chartName)

Arguments gfx A graphics handle.
chartName The name of the chart being deleted.

Description Removes a chart and frees the memory associated with the chart. The name associated with the chart (that is, chartName) is also deleted.

See also [GR_CHART_CREATE@](#)

GR/GFX_CHART_DESTROY_AXIS@

Deletes an axis from a chart

Format remainingAxesArray = GR_CHART_DESTROY_AXIS@ (chartName, axisName)
remainingAxesArray = GFX_CHART_DESTROY_AXIS@ (gfx, chartName, axisName)

Method [remainingAxesArray = this.chart_destroy_axis@](#) (chartName, axisName)

Arguments gfx A graphics handle.
chartName The name of a chart.
axisName One of the following strings:
 axis x<number>
 axis y<number>

Description Deletes one of a chart's axis from the displayed chart. The information associated with this axis is also deleted from memory.

When you used [GR_CHART_CREATE_AXIS@](#) to create an axis, it returned the name of the newly created axis.

GR/GFX_CHART_DESTROY_GROUP@

Removes a group

Format groupArray = GR_CHART_DESTROY_GROUP@ (chartName, groupName)
groupArray = GFX_CHART_DESTROY_GROUP@ (gfx, chartName, groupName)

Method groupArray = this.chart_destroy_group@ (chartName, groupName)

Arguments

gfx	A graphics handle.
chartName	The name of a chart within the current document.
groupName	The name of the group being created. If this argument is NULL or is not a string, a name is created for you. A groupName created by Graphics is in the following form: data 0 data 1 data 2 ... However, the value following data can be any user-defined string.

Description Deletes the group from memory and from the displayed chart. If a legend is being displayed, information is deleted from the legend.

A list containing the names of the remaining groups is returned.

See also [GR_CHART_CREATE_GROUP@](#)

GR/GFX_CHART_DRAW@

Draws, or redraws, a chart

Format GR_CHART_DRAW@(chartName[, x1, y1, x2, y2])
GFX_CHART_DRAW@(gfx, chartName[, x1, y1, x2, y2])

Method this.chart_draw@(chartName[, x1, y1, x2, y2])

Arguments

gfx	A graphics handle.
chartName	The name of the chart being drawn (or redrawn).
x1	One of the two x-coordinates of the chart.

y2 One of the two y-coordinates of the chart.
 x2 The second of the two x-coordinates of the chart.
 y2 The second of the two y-coordinates of the chart.

Description Draws (or redraws) a chart within the rectangle defined by the four x and y coordinates. These coordinates should be the upper left (x1 and y1) and lower right corners (x2 and y2) of the chart's bounding rectangle.

If you are redrawing an existing chart, the X and Y coordinates are optional. If you omit them, the chart is redrawn within the existing extent.

GR/GFX_CHART_FORMAT_NUMBER@

Displays a formatted number

Format formattedString = GR_CHART_FORMAT_NUMBER@ (chart_number_format@ format, value)

formattedString = GFX_CHART_FORMAT_NUMBER@ (gfx, chart_number_format@ format, value)

Method formattedString = this.chart_format_number@ (chart_number_format@ format, value)

Arguments gfx A graphics handle.
 format The format into which value will be converted. The structure of this data is discussed within the description.
 value The number that will be displayed in a format.

Description Displays a number (that is, value) after it is converted into a format. For example, a numeric value could be formatted as a date. For general information, see [Spreadsheets Number Formats](#).

The definition of chart_number_format@ is as follows:

```
format chart_number_format@
  style,            One of the following strings:
                   Unstyled    Currency
                   Comma      Fixed
                   General    Scientific
                   Percentage Date
                   Boolean    Graph
                   Time        Custom
  aux_style_info,  date formats:
                   1        Mmmm d, yyy
```

- 2 Mm d, yyyy
- 3 d Mm yyy
- 4 mm/dd/yy
- 5 dd.mm.yy
- 6 yyyy-mm-dd
- 7 yy-mm-dd
- 8 yyyy mm dd
- 9 yy mm dd
- 10 yyymmdd
- 11 yymmdd
- 12 dd/mm/yy
- 13 dd.mm.yyyy
- 14 Mm dd, yyyy
- 15 Mmmm yyyy
- 16 Mm yyyy
- 17 Mm yy
- 18 mm/dd
- 19 yy mm
- 20 yyyy mm

time formats:

- 0 hh:mm:ss AM/PM
- 1 hh:mm AM/PM
- 2 hh:mm:ss
- 3 hh:mm
- 4 hh:mm:ss.tss AM/PM
- 5 hh:mm:ss.tss

money format:

- 0 Leading Symbol
- 1 Trailing Symbol

units, Number: the display units of tick label values. The actual values of the data are divided by the unit number to produce the displayed tick label values

radix_places, Number: maximum value is 9

prefix, String

suffix, String

am_str, String

pm_str, String

true_str, String

false_str, String

thousands_seperator, String

decimal_seperator, String

display_format_errors, Boolean
trim_setting Boolean indicating if trailing zeros are eliminated

GR/GFX_CHART_GET_3D@

Returns general 3D information: yaw, pitch, projection, and depth

Format format chart_3D_effect@ 3D = GR_CHART_GET_3D@(chartName)
format chart_3D_effect@ 3D = GFX_CHART_GET_3D@(gfx, chartName)

Method format chart_3D_effect@ 3D = this.chart_get_3d@(chartName)

Arguments gfx A graphics handle.
chartName The name of a chart within the current document.

Description Returns a chart_3D_effect@ format containing general 3D information. The definition of this format is as follows:

format chart_3D_effect@
enabled,
yaw,
pitch,
projection,
depth

See also [GR_CHART_SET_3D@](#)

GR/GFX_CHART_GET_ATTR@

Returns graphic attributes of chart

Format format gr_attribute@ info = GR_CHART_GET_ATTR@(chartName, element)
format gr_attribute@ info = GFX_CHART_GET_ATTR@(gfx, chartName, element)

Method format gr_attribute@ info = this.chart_get_attr@(chartName, element)

Arguments gfx A graphics handle.
chartName The name of the chart.
element One of the following chart elements:

default
 minor horizontal grid
 major horizontal grid
 major vertical grid
 minor vertical grid
 line, axis x<number>
 line, axis y<number>
 label, axis x<number>
 label, axis y<number>
 tik label, axis x<number>
 tik label, axis y<number>
 data
 label
 data
 title
 subtitle
 footer
 legend box
 legend title
 legend labels

Description Returns the current fill, shadow, line, text field, and character attributes for a chart component.

The definition of `gr_attribute@` is as follows:

format `gr_attribute@`

format `gr_fill_attr_type@` backfill,
 format `gr_fill_attr_type@` linefill,
 format `gr_shadow_attr_type@` shadow,
 format `gr_line_style_type@` line,
 format `gr_text_field_attr_type@` field,
 format `gr_text_char_attr_type@` char,

format `gr_file_attr_type@`

type, 'string; built-in <#>, <filename>, linear gradient
 fg_color, 'string: name of color in colormap
 bg_color, 'string: name of color in colormap
 angle, 'int: in degrees
 offset 'int: in mills (1000 mills = 1 inch)

format `gr_shadow_attr_type@`

type, 'string: none, background drop shadow, local drop shadow
 color, 'string: name of color in colormap
 'int: in mils
 horizontal_offset

vertical_offset
'int: in mils

format gr_line_style_type@
style,
weight,
first_symbol,
join_symbol,
final_symbol

format gr_text_char_attr_type@
face,
size,
ruling,
strike_thru,
bold,
italic,
horizontal_scale,
vertical_offset,
horizontal_sub_sup,
vertical_sub_sup

format gr_text_field_attr_type@
horizontal_alignment,
vertical_alignment,
line_space,
horizontal_scale,
vertical_scale,
shear,
angle,
left_margin,
right_margin,
top_margin,
bottom_margin

In this format, the following elements are always NULL:

- field.line_space
- field.left_margin
- field.right_margin
- field.top_margin
- field.bottom_margin

See also [GR CHART SET ATTR@](#)

GR/GFX_CHART_GET_AXIS_LABELS@

Places label information into 'info'

Format format chart_axis_labels@ info = GR_CHART_GET_AXIS_LABELS@(chartName, axisName)

format chart_axis_labels@ info = GFX_CHART_GET_AXIS_LABELS@ (gfx, chartName, axisName)

Method format chart_axis_labels@ info = this.chart_get_axis_labels@(chartName, axisName)

Arguments

gfx	A graphics handle.
chartName	The name of a chart within the current document.
axisName	The name of one of the chart's axes. This name takes the following form: axis x1 axis x2 ... axis y1 axis y2 ...

Description Returns a data structure containing information defining a chart's labels. The definition of this structure is:

```
format chart_axis_labels@
    type,          'category, value, log10
    tight_label,
    label,
    label_x_margin,
    label_y_margin,
    label_alignment,
    format chart_number_format@ tik_format,
    tight_tiks,
    tik_labels,
    tik_margin
    tik_alignment,
    tik_filter,
    tik_flags,
    hidden
```

The definition of chart_number_format@ is as follows:

```
format chart_number_format@
```

style,
aux_style_info,
units,
radix_places,
prefix,
suffix,
am_str,
prm_str,
true_str,
false_str,
thousands_seperator,
decimal_separator,
display_format_errors,
trim_string

See also [GR_CHART_SET_AXIS_LABELS@](#)

GR/GFX_CHART_GET_AXIS_LINE@

Returns an axis' line attributes

Format format chart_axis@ info = GR_CHART_GET_AXIS_LINE@(chartName, axisName)
format chart_axis@ info = GFX_CHART_GET_AXIS_LINE@(gfx, chartName,
axisName)

Method format chart_axis@ info = this.chart_get_axis_line@(chartName, axisName)

Arguments

gfx	A graphics handle.
chartName	The name of a chart within the current document.
axisName	The name of one of the chart's axes. This name takes the following form: axis x1 axis x2 ... axis y1 axis y2 ...

Description Returns a data structure containing information defining one a chart's axis lines. The definition of this structure is:

format chart_axis@
type, 'category, value, log10
auto_max,

max_value,
auto_min,
min_value,
auto_bas,
bas_val,
position,
floating_value,
minor_tik_type,
minor_tik_size,
auto_n_minor_tiks,
n_minor_tiks,
major_tik_type,
major_tik_size,
auto_n_major_tiks,
n_major_tiks,
bar_margin,
bar_overlap,
use_tik_attriutes,
behind_data,
hidden

See also [GR CHART SET AXIS LINE@](#)

GR/GFX_CHART_GET_AXES@

Returns axes names

Format axesNameArray = GR_CHART_GET_AXES@(chartName)
axesNameArray = GFX_CHART_GET_AXES@(gfx, chartName)

Method axesNameArray = this.chart_get_axes@(chartName)

Arguments gfx A graphics handle.
chartName The name of a chart within the current document.

Description Returns the names of axes. These names take the following form:

axis x1
axis x2
...
axis y1

axis y2

...

GR/GFX_CHART_GET_AXIS@

Returns an axis' definition

Format format chart_axis_info@ info = GR_CHART_GET_AXIS@(chartName, axisName)
format chart_axis_info@ info = GFX_CHART_GET_AXIS@(gfx, chartName, axisName)

Method format chart_axis_info@ info = this.chart_get_axis@(chartName, axisName)

Arguments

gfx	A graphics handle.
chartName	The name of a chart within the current document.
axisName	The name of one of the chart's axes. This name takes the following form: axis x1 axis x2 ... axis y1 axis y2 ...

Description Returns a data structure containing information defining one of a chart's axes. For information on this structure and its contents, see [GR_CHART_CREATE_AXIS@](#).

See also [GR_CHART_GET_AXES@](#)
[GR_CHART_GET_CHARTS@](#)
[GR_CHART_SET_AXIS@](#)

GR/GFX_CHART_GET_CHARTS@

Returns all chart names

Format chartArray = GR_CHART_GET_CHARTS@()
chartArray = GFX_CHART_GET_CHARTS@(gfx)

Method chartArray = this.chart_get_charts@

Arguments gfx A graphics handle.

Description Returns an array that contains the names of all charts.

GR/GFX_CHART_GET_DATUM@

Returns a group's data

Format format chart_datum@ datum = GR_CHART_GET_DATUM@ (chartName, groupName, dataIX)

format chart_datum@ datum = GFX_CHART_GET_DATUM@ (gfx, chartName, groupName, dataIX)

Method format chart_datum@ datum = this.chart_get_datum@ (chartName, groupName, dataIX)

Arguments

gfx	A graphics handle.
chartName	The name of a chart within the current document.
groupName	The name of the group being created. If this argument is NULL or is not a string, a name is created for you. A group name created by Graphics is of the following form: data 0 data 1 data 2 ... However, the value following data can be any user-defined string.
dataIX	The data element index; that is, this number indicates the data element within the group identified by groupName.

Description Returns the data associated with a group and attributes that are unique to this data series. The definition of the returned data is as follows:

format chart_datum@

label_x_offset	' The label's X offset from the data point
label_y_offset,	' The label's Y offset from the data point
label_string,	' The text of the label
display_type,	' How to display the point
plot	display it, or if the point is NULL, use the default NULL format
substitute	
	if null, substitute a zero
span	if outlier, ignore
gap	Place gap in line
x,	' The x value associated with a point
y,	' The y value associated with a point

Description Returns the coordinates of a chart's bounding box as the following four element arrays:

- x0
- y0
- x1
- y1

These coordinates represent the upper left and lower right coordinates.

See also [GR_CHART_SET_EXTENT@](#)

GR/GFX_CHART_GET_GROUP@

Returns group information

Format format chart_group@ group = GR_CHART_GET_GROUP@(chartName, groupName)
format chart_group@ group = GFX_CHART_GET_GROUP@(gfx, chartName, groupName)

Method format chart_group@ group = this.chart_get_group@(chartName, groupName)

Arguments

gfx	A graphics handle.
chartName	The name of a chart within the document.
groupName	The name of the group being created. A groupName created by Graphics is in the following form: data 0 data 1 data 2 ... However, the value following data can be any user-defined string.

Description Returns the information associated with a group. For the definition of chart_group@, see [GR_CHART_SET_GROUP@](#).

GR/GFX_CHART_GET_GROUPS@

Returns names of all groups in a chart

Format groupArray = GR_CHART_GET_GROUPS@ (chartName)
groupArray = GFX_CHART_GET_GROUPS@ (gfx, chartName)

Method groupArray = this.chart_get_groups@ (chartName)

Arguments gfx A graphics handle.
chartName The name of a chart.

Description Returns an array that contains the names of all groups contained within a chart.

GR/GFX_CHART_GET_LEGEND@

Returns legend information

Format format chart_legend@ legend = GR_CHART_GET_LEGEND@(chartName)
format chart_legend@ legend = GFX_CHART_GET_LEGEND@(gfx, chartName)

Method format chart_legend@ legend = this.chart_get_legend@(chartName)

Arguments gfx A graphics handle.
chartName The name of a chart.

Description Returns the information associated with a chart's legend. For the definition of chart_legend@, see [GR_CHART_SET_LEGEND@](#).

GR/GFX_CHART_GET_MARGIN@

Returns a chart's margins

Format marginArray = GR_CHART_GET_MARGIN@(chartName)
marginArray = GFX_CHART_GET_MARGIN@(gfx, chartName)

Method marginArray = this.chart_get_margin@(chartName)

Arguments gfx A graphics handle.
chartName The name of a chart within a document.

Description Returns the margin within the chart's extent. That is, these margins are within the area used by Graphics to draw the chart. The returned array has four numeric elements representing, in order, the:

- Right margin
- Left margin

- Bottom margin
- Right margin

Note that the title, subtitle, footer, and legend ignore these margins and are placed on the extent.

See also [GR_CHART_SET_MARGIN@](#)

GR/GFX_CHART_GET_NULL_FORMAT@

Indicates how NULL items are displayed

Format formatString = GR_CHART_GET_NULL_FORMAT@ (chartName)
 formatString = GFX_CHART_GET_NULL_FORMAT@ (gfx, chartName)

Method formatString = this.chart_get_null_format@ (chartName)

Arguments gfx A graphics handle.
 chartName The name of a chart within the current document.

Description Returns a string indicating how NULL data points are displayed. The returned value is one of the following strings:

- zero, indicating that null points are displayed as zeroes.
- span, indicating that the point is ignored (that is the graph *spans* this point).
- gap, indicating that if a line is being drawn, a gap is displayed where the point should be.

GR/GFX_CHART_GET_ORIENTATION@

Returns a chart's orientation

Format list = GR_CHART_GET_ORIENTATION@ (chartName)
 list = GFX_CHART_GET_ORIENTATION@(gfx, chartName)

Method list = this.chart_get_orientation@ (chartName)

Arguments gfx A graphics handle.
 chartName The name of a chart within the current document.

Description Returns a chart's orientation (that is, the direction of the chart's major axis) as one of the following strings:

- vertical
- horizontal

See also [GR CHART SET ORIENTATION@](#)

GR/GFX_CHART_GET_TITLE@

Returns all of a chart's titles

Format format chart_titles@ info = GR_CHART_GET_TITLE@(chartName)
format chart_titles@ info = GFX_CHART_GET_TITLE@(gfx, chartName)

Method format chart_titles@ info = this.chart_get_title@(chartName)

Arguments gfx A graphics handle.
chartName The name of a chart within the current document.

Description Returns the text and alignment for a chart's title, subtitle, and footer, as follows:

```
format chart_title@
  title,
  subtitle,
  footer,
  title_alignment,
    0    left
    1    center
    2    right
  subtitle_alignment,
    0    left
    1    center
    2    right
  footer_alignment
    0    left
    1    center
    2    right
```

See also [GR CHART SET TITLE@](#)

GR/GFX_CHART_GET_TYPE_INFO@

Returns the bar margin, bar overlay, and starting position

Format `format chart_type_info@ info = GR_CHART_GET_TYPE_INFO@(chartName)`
`format chart_type_info@ info = GFX_CHART_GET_TYPE_INFO@(gfx, chartName)`

Method `format chart_type_info@ info = this.chart_get_type_info@(chartName)`

Arguments `gfx` A graphics handle.
`chartName` The name of a chart within the current document.

Description Returns bar chart information. This information is either what is used to display the current chart or what would be used if the chart were being displayed as a bar chart.

The definition of `chart_type_info@` is as follows:

```
format chart_type_info@
    bar_margin,
    bar_overlap,
    starting_pos
```

GR/GFX_CHART_GET_TYPE_NAME@

Returns a chart's type

Format `typeString = GR_CHART_GET_TYPE_NAME@ (chartName)`
`typeString = GFX_CHART_GET_TYPE_NAME@ (gfx, chartName)`

Method `typeString = this.chart_get_type_name@ (chartName)`

Arguments `gfx` A graphics handle.
`chartName` The name of a chart within the current document.

Description Returns a string indicating a chart's type, as follows:

```
cat-val
histogram
spider
strata
surface
val-val
```

See also [GR_CHART_SET_TYPE@](#)

GR/GFX_CHART_RENAME@

Renames a chart

Format GR_CHART_RENAME@(oldName, newName)
GFX_CHART_RENAME@(gfx, oldName, newName)

Method [this.chart_rename@](#)(oldName, newName)

Arguments gfx A graphics handle.
oldName The old name of the chart.
newName The new name of the chart.

Description Changes a chart's name from oldName to newName.

GR/GFX_CHART_RESTORE_HOST@

Makes the host task the chart's parent task

Format GR_CHART_RESTORE_HOST@(
GFX_CHART_RESTORE_HOST@(gfx)

Method [this.chart_restore_host@](#)

Arguments gfx A graphics handle.

Description Names the host task as the chart's parent task. This allows you to close the graphics task and transfer control to the parent.

GR/GFX_CHART_SAVE@

Saves the current chart

Format GR_CHART_SAVE@(
GFX_CHART_SAVE@(gfx)

Method [this.chart_save@](#)

Arguments gfx A graphics handle.

Description Saves the current chart. The editing session with the chart is not ended. If you are editing a bitmap, the chart is saved in *.im* format. Otherwise, it is saved in the Applixware graphics format.

GR/GFX_CHART_SET_3D@

Defines general 3D information: pitch, yaw, projection, and depth

Format GR_CHART_SET_3D@(chartName, format chart_3D_effect@ threeD)
GFX_CHART_SET_3D@(gfx, chartName, format chart_3D_effect@ threeD)

Method [this.chart_set_3d@](#)(chartName, format chart_3D_effect@ threeD)

Arguments gfx A graphics handle.
chartName The name of a chart within the current document.
threeD The 3D information being set.

Description Defines the general 3D information contained within a chart_3D_effect@ format. The definition of this format is as follows:

```
format chart_3D_effect@
    enabled,
    yaw,
    pitch,
    projection,
    depth
```

See also [GR_CHART_GET_3D@](#)

GR/GFX_CHART_SET_ATTR@

Sets graphic attributes for a chart

Format GR_CHART_SET_ATTR@(chartName, element, format gr_attribute@ attr[, noDrawFlag])
GFX_CHART_SET_ATTR@(gfx, chartName, element, format gr_attribute@ attr[, noDrawFlag])

Method [this.chart_set_attr@](#)(chartName, element, format gr_attribute@ attr[, noDrawFlag])

Arguments	gfx	A graphics handle.
	chartName	The name of the chart.
	element	One of the following chart elements: default minor horizontal grid major horizontal grid major vertical grid minor vertical grid line, axis x<number> line, axis y<number> label, axis x<number> label, axis y<number> tik label, axis x<number> tik label, axis y<number> data label, data title subtitle footer legend box legend title legend labels
	attr	The attributes being set for element.
	noDrawFlag	A Boolean value where TRUE indicates that the chart should not be re-drawn when the new attributes are applied.

Description Sets the graphic display attributes for an element in the chart. For more information, see [**GR_CHART_GET_ATTR@**](#).

If you are not setting (or using) an attribute within the format, set the attribute's value to NULL.

GR/GFX_CHART_SET_AXIS@

Sets axis information

Format GR_CHART_SET_AXIS@(chartName, axisName,
 format chart_axis_info@ info)

GFX_CHART_SET_AXIS@(gfx, chartName, axisName,
 format chart_axis_info@ info)

Method `this.chart_set_axis@(chartName, axisName, format chart_axis_info@ info)`

Arguments

<code>gfx</code>	A graphics handle.
<code>chartName</code>	The name of the chart.
<code>axisName</code>	The name of one of the chart's axes. This name takes the following form: axis x1 axis x2 ... axis y1 axis y2 ...
<code>info</code>	The axis information being set.

Description Changes the information associated with an axis to `info`.
For information on the `chart_axis_info@` format, see [GR CHART CREATE AXIS@](#).

See also [GR CHART GET AXIS@](#)

GR/GFX_CHART_SET_AXIS_LABELS@

Sets axis label information

Format `GR_CHART_SET_AXIS_LABELS@(chartName, axisName, format chart_axis_labels@ info)`
`GFX_CHART_SET_AXIS_LABELS@(gfx, chartName, axisName, format chart_axis_labels@ info)`

Method `this.chart_set_axis_labels@(chartName, axisName, format chart_axis_labels@ info)`

Arguments

<code>gfx</code>	A graphics handle.
<code>chartName</code>	The name of a chart within the current document.
<code>axisName</code>	The name of one of the chart's axes. This name takes the following form: axis x1 axis x2 ... axis y1 axis y2 ...

info The axis label information.

Description Defines the properties and attributes of an axis label. This information is defined as a `chart_axis_labels@` format whose definition is as follows:

```
format chart_axis_labels@
    type,            'category, value, log10
    tight_label,
    label,
    label_x_margin,
    label_y_margin,
    label_alignment,
    format chart_number_format@ tik_format,
    tight_tiks,
    tik_labels,
    tik_margin
    tik_alignment,
    tik_filter,
    tik_flags,
    hidden
```

The definition of `chart_number_format@` is as follows:

```
format chart_number_format@
    style,
    aux_style_info,
    units,
    radix_places,
    prefix,
    suffix,
    am_str,
    prm_str,
    true_str,
    false_str,
    thousands_seperator,
    decimal_separator,
    display_format_errors,
    trim_string
```

See also [GR CHART GET AXIS LABELS@](#)

GR/GFX_CHART_SET_AXIS_LINE@

Sets axis line information

Format GR_CHART_SET_AXIS_LINE@(chartName, axisName, format chart_axis@ info)
GFX_CHART_SET_AXIS_LINE@(gfx, chartName, axisName, format chart_axis@ info)

Method [this.chart_set_axis_line@](#)(chartName, axisName, format chart_axis@ info)

Arguments

gfx	A graphics handle.
chartName	The name of a chart within the current document.
axisName	The name of one of the chart's axes. This name takes the following form: axis x1 axis x2 ... axis y1 axis y2 ...
info	Information describing one of a chart's axis.

Description Sets the properties and attributes of one of a chart's axis lines. The definition of the format used when invoking this macro is as follows:

format chart_axis@
type, 'category, value, log10
auto_max,
max_value,
auto_min,
min_value,
auto_bas,
bas_val,
position,
floating_value,
minor_tik_type,
minor_tik_size,
auto_n_minor_tiks,
n_minor_tiks,
major_tik_type,
major_tik_size,
auto_n_major_tiks,
n_major_tiks,

bar_margin,
bar_overlap,
use_tik_attriutes,
behind_data,
hidden

See also [GR_CHART_GET_AXIS_LINE@](#)

GR/GFX_CHART_SET_DATUM@

Sets the chart's data points

Format GR_CHART_SET_DATUM@(chartName, groupName, dataIX, format chart_datum@ data)

GFX_CHART_SET_DATUM@(gfx, chartName, groupName, dataIX, format chart_datum@ data)

Method [this.chart_set_datum@](#)(chartName, groupName, dataIX, format chart_datum@ data)

Arguments

gfx	A graphics handle.
chartName	The name of a chart within the current document.
groupName	The name of the group being created. If this argument is NULL or is not a string, a name is created for you. A groupName created by Graphics is in the following form: data 0 data 1 data 2 ... However, the value following data can be any user-defined string.
dataIX	A data element index; that is, this number identifies which data point is being set.
data	The data being set.

Description Sets a data point (and information associated with this data point). This information is passed as the following format:

format chart_datum@	
label_x_offset,	' the label's X offset from the data point
label_y_offset,	' the label's Y offset from the data point
label_string,	' the text of the label
display_type,	'How to display the point

	plot	display it, or if the point is NULL, use the default NULL format
	substitute	if null, substitute a zero
	span	if outlier, ignore
	gap	Place gap in line
x,		'The x value associated with a point
y,		'The y value associated with a point
z		'The z value associated with a point

Note that an x/y bubble chart needs 1 X value and 2 Y values.

For more information, see [GR_CHART_GET_DATUM@](#).

GR/GFX_CHART_SET_DECORATIONS@

Sets a chart's borders and grids

Format GR_CHART_SET_DECORATIONS@(chartName, format chart_decorations@ layout)
 GFX_CHART_SET_DECORATIONS@(gfx, chartName, format chart_decorations@ layout)

Method [this.chart_set_decorations@](#)(chartName, format chart_decorations@ layout)

Arguments

gfx	A graphics handle.
chartName	The name of a chart within the current document.
layout	A set of TRUE/FALSE values indicating if one of the eight kinds of chart <i>decoration</i> is drawn.

Description Sets the following format, which contains 8 Boolean chart properties:

```
format chart_decorations@
  close_top,
  close_bottom,
  close_left,
  close_right,
  major_h_grids,    ' horizontal
  minor_h_grids,
  major_v_grids,    ' vertical
  minor_v_grids
```

GR/GFX_CHART_SET_EXTENT@

Sets a chart's size

Format GR_CHART_SET_EXTENT@(chartName, x1, y1, x2, y2)
GFX_CHART_SET_EXTENT@(gfx, chartName, x1, y1, x2, y2)

Method [this.chart_set_extent@](#)(chartName, x1, y1, x2, y2)

Arguments

gfx	A graphics handle.
chartName	A chart's name.
x1	One of a chart's x-coordinates.
y1	One of a chart's y-coordinates.
x2	The second of a chart's x-coordinates.
y2	The second of a chart's y-coordinates.

Description Sets the size at which a chart will be drawn. The coordinates represent the upper left (x1 and y1) and lower right (x2 and y2) of the bounding box surrounding the chart.

GR/GFX_CHART_SET_GROUP@

Sets a data group

Format GR_CHART_SET_GROUP@(chartName, groupName, format chart_group@ inGrp)

Method [this.chart_set_group@](#)(chartName, groupName, format chart_group@ inGrp)
[GFX_CHART_SET_GROUP@](#)(gfx, chartName, groupName, format chart_group@ inGrp)

Arguments

gfx	A graphics handle.
chartName	A chart's name.
groupName	The name of the group being created. If this argument is NULL or is not a string, a name is created for you. A groupName created by Graphics is in the following form: data 0 data 1 data 2 ...

However, the value following data can be any user-defined string.

inGrp The data for the group.

Description Defines how a group is drawn. The definition of how it is drawn is contained within in-Grp.

```
format chart_group@
    type,
        0    line
        1    curve
        2    bar
        3    league bar
        4    custom bar
        5    bubble
        6    hilo1
        7    strata
        8    pie
        9    hilo2
        10   stacked bar
        11   stacked custom bar
        12   stacked strata

    stack_id,
    custom_part_name,
    x_axis,          parent x axis
    y_axis,          parent y axis
    z_axis,          parent z axis
    label_x_offset, label's x offset
    label_y_offset, label's y offset
    label_type,
        0    none
        1    value
        2    percent
        3    string

    format chart_number_format@
        label_format,
    format arrayof chart_datum@
        data,
    legend_string,  An array or a string
    label_alignment,
        left
        center
        right
```

See also [GR CHART GET GROUP@](#)

GR/GFX_CHART_SET_LEGEND@

Creates the chart's legend

Format GR_CHART_SET_LEGEND@(chartName, format chart_legend@ legend)
GFX_CHART_SET_LEGEND@(gfx, chartName, format chart_legend@ legend)

Method [this.chart_set_legend@](#)(chartName, format chart_legend@ legend)

Arguments gfx A graphics handle.
chartName A chart's name.
legend The data for the legend.

Description Sets a chart's legend to the values within legend.

format chart_legend@
disabled, Boolean
uses_x_space, Boolean
uses_y_space, Boolean
text_before_sample, Boolean
arrange_by_row, Boolean
max_per_row_or_column,
 Number
title, String
horizontal_alignment, left
 center
 right
horizontal_offset, Number
vertical_alignment, top
 middle
 bottom

vertical_offset,
top_margin,
bottom_margin,
row_margin,
left_margin,
right_margin,
column_margin,
proportional_margins,
title_alignment

See also [GR_CHART_GET_LEGEND@](#)

GR/GFX_CHART_SET_MARGIN@

Sets the margin within the chart area

Format GR_CHART_SET_MARGIN@(chartName, x1, y1, x2, y2)
GFX_CHART_SET_MARGIN@(gfx, chartName, x1, y1, x2, y2)

Method [this.chart_set_margin@](#)(chartName, x1, y1, x2, y2)

Arguments

gfx	A graphics handle.
chartName	The chart's name within the current document.
x1	The left margin.
y1	The chart's top margin.
x2	The chart's right margin.
y2	The chart's bottom margin

Description Sets the margin within the chart's extent. That is, these margins are within the area used by Graphics to draw the chart. The default value is 500 mils. (1000 mils = 1 inch)
Note that the title, subtitle, footer, and legend ignore these margins and are placed on the extent.

See also [GR_CHART_GET_MARGIN@](#)

GR/GFX_CHART_SET_NULL_FORMAT@

Sets the format when a point is NULL

Format GR_CHART_SET_NULL_FORMAT@(chartName, formatString)
GFX_CHART_SET_NULL_FORMAT@(gfx, chartName, formatString)

Method [this.chart_set_null_format@](#)(chartName, formatString)

Arguments

gfx	A graphics handle.
chartName	The name of chart within the current document.
formatString	One of the following values: zero span gap

Description Tells Graphics how it should chart NULL points. For more information, see [GR_CHART_GET_NULL_FORMAT@](#).

GR/GFX_CHART_SET_ORIENTATION@

Sets the chart's orientation to horizontal or vertical

Format GR_CHART_SET_ORIENTATION@(chartName, type)
GFX_CHART_SET_ORIENTATION@(gfx, chartName, type)

Method [this.chart_set_orientation@](#)(chartName, type)

Arguments

gfx	A graphics handle.
chartName	The name of the chart within the current document.
type	The chart's orientation. To set the orientation to vertical, set this parameter's value to "vertical". Any other value sets the orientation to horizontal.

Description Sets the chart's orientation to horizontal or vertical.

See also [GR_CHART_GET_ORIENTATION@](#)

GR/GFX_CHART_SET_TITLE@

Sets a chart's titles

Format GR_CHART_SET_TITLE@(chartName, format chart_titles@ titles)
GFX_CHART_SET_TITLE@(gfx, chartName, format chart_titles@ titles)

Method [this.chart_set_title@](#)(chartName, format chart_titles@ titles)

Arguments

gfx	A graphics handle.
chartName	The name of the chart.
titles	The title information being set.

Description Sets the title, subtitle, and footer as well as these element's alignment.

format chart_title@
title,
subtitle,
footer,

title_alignment,
subtitle_alignment,
footer_alignment

See also [GR CHART GET TITLE@](#)

GR/GFX_CHART_SET_TYPE@

Sets a chart's type

Format GR_CHART_SET_TYPE@(chartName, type
[, format chart_type_info@ data])
GFX_CHART_SET_TYPE@(gfx, chartName, type
[, format chart_type_info@ data])

Method [this.chart_set_type@](#)(chartName, type
[, format chart_type_info@ data])

Arguments

gfx	A graphics handle.
chartName	The name of the chart.
type	The new 'chart type'. This type is one of the following strings: cat-val histogram spider strata surface val-val
data	The following general chart information format chart_type_info@ bar_margin, ' Number bar_overlap, ' Number starting_pos ' Number

Description Sets the type of chart to be displayed and passes to it information it needs when deciding how to display groups in the chart.

See also [GR CHART GET TYPE NAME@](#)
[GR CHART CHANGE CHART TYPE@](#)

GR/GFX_COMBINE@

Combines objects and alternately applies attributes between object edges

Format GR_COMBINE@ ()
GFX_COMBINE@(gfx)

Method [this.combine@](#)

Arguments gfx A graphics handle.

Description This macro alternately applies attributes and no attributes to the areas between the edges of objects.

When you combine objects, the objects are treated as one object. Based on an odd/even number of edges within the combined object, enclosed areas alternately display and do not display. The areas which display possess the attributes of the object which was backmost (usually the object created first) . This macro is called by Arrange ® Combine.

See also [GR_UNCOMBINE@](#)

GR/GFX_COORDINATES@

Sets the coordinate display

Format GR_COORDINATES@()
GFX_COORDINATES@(gfx)

Method [this.coordinates@](#)

Arguments gfx A graphics handle.

Description Toggles the display of coordinates on and off. If coordinates are not currently displayed when objects are drawn or moved, GR_COORDINATES@ displays the coordinates. If coordinates are currently displayed, GR_COORDINATES@ turns off the display of coordinates.

This macro is called by View ® Coordinates.

GR/GFX_COPY@

Copies selected objects

Format GR_COPY@()
GFX_COPY@(gfx)

Method [this.copy@](#)

Arguments gfx A graphics handle.

Description Copies selected objects from the current Graphics document and places the copy in the clipboard.

See also [GR CUT@](#)
[GR PASTE@](#)

GR/GFX_CREATE_CALLBACK@

Adds a new callback macro to the graphics document

Format index = GR_CREATE_CALLBACK@(format gr_callback_info@ macroName)
index = GFX_CREATE_CALLBACK@(gfx, format gr_callback_info@ macroName)

Method [index = this.create_callback@\(format gr_callback_info@ macroName\)](#)

Arguments gfx A graphics handle.
macroName The name of the macro being added to the graphics document.

Description Adds a new macro to the graphics document. The index position of the macro within the array of callback macros is returned. As the callback macro is added to the end of the callback macro array, this number also indicates the last position in the array.

The definition of the gr_callback_info@ format is as follows:

```
format gr_callback_info@
    macro_name,
    trigger_on_drag,
    trigger_on_single_click,
    trigger_on_double_click,
    argument_list
```

See also [GR CHANGE CALLBACK@](#)
[GR GET CALLBACK@](#)
[GR GET CALLBACKS@](#)

GR/GFX_CREATE_COLOR@

Creates a new color

Format GR_CREATE_COLOR@(cyan, magenta, yellow, black, maxVal, transparent, type, name)

GFX_CREATE_COLOR@(gfx, cyan, magenta, yellow, black, maxVal, transparent, type, name)

Method this.create_color@(cyan, magenta, yellow, black, maxVal, transparent, type, name)

Arguments

gfx	A graphics handle.
cyan	The color's cyan component.
magenta	The color's magenta component.
yellow	The color's yellow component.
black	The color's black component.
maxVal	A number from 0 to 255.
transparent	A Boolean value where TRUE indicates that the color is transparent.
type	Not used.
name	An arbitrary name assigned to the color

Description Adds a new color to the default color map. The color palette and the color selection dialog box (if displayed) will be updated.

GR_CREATE_DBASE_OBJECT@

Creates a new Applixware Data inset

Format GR_CREATE_DBASE_OBJECT@()

Description Embeds a new Data query inset in Applixware Graphics, and opens an Applixware Data window. Refer to Chapter 13, "[Working with Objects](#)", in the Applixware Words manual for more information on Applixware data insets.

GR_CREATE_EQN_OBJECT@

Creates a new Equation Editor inset

Format GR_CREATE_EQN_OBJECT@()

Description Embeds a new equation editor inset in Applixware Graphics, and opens an Equation Editor window. Refer to Chapter 13, "[Working with Objects](#)", in the Applixware Words manual for more information on Applixware insets.

GR_CREATE_GR_OBJECT@

Creates a new Applixware Graphics inset

Format GR_CREATE_GR_OBJECT@()

Description Embeds a new Graphics Editor inset in Applixware Graphics, and opens an empty Graphics window. Refer to Chapter 13, "[Working with Objects](#)", in the Applixware Words manual for more information on Applixware insets.

GR/GFX_CREATE_LAYER@

Creates a new layer

Format GR_CREATE_LAYER@(layerName)
GFX_CREATE_LAYER@(gfx, layerName)

Method [this.create_layer@\(layerName\)](#)

Arguments gfx A graphics handle.
layerName The name of the layer being created.

Description Creates a new layer named layerName. No error occurs if this layer already exists.

See also [GR_DELETE_LAYER@](#)

GR_CREATE_SS_OBJECT@

Creates a new Applixware Spreadsheet inset

Format GR_CREATE_SS_OBJECT@()

Description Embeds a new Spreadsheets inset in Applixware Graphics, and opens an empty Spreadsheets window. Refer to Chapter 13, "[Working with Objects](#)", in the Applixware Words manual for more information on Applixware insets.

GR_CREATE_WP_OBJECT@

Creates a new Applixware Words inset

Format GR_CREATE_WP_OBJECT@()

Description Embeds a new Words inset in Applixware Graphics, and opens an empty Words window. Refer to Chapter 13, "[Working with Objects](#)", in the Applixware Words manual for more information on Applixware insets.

GR/GFX_CREATE_PART@

Creates a custom part

Format GR_CREATE_PART@(name, fixedSizeFlag)
GFX_CREATE_PART@(gfx, name, fixedSizeFlag)

Method [this.create_part](#)@(name, fixedSizeFlag)

Arguments

gfx	A graphics handle.
name	A string indicating the name for the part to be created. name cannot exceed 30 characters. If name is an empty string, the system assigns a name to the part (part1, part2, and so on). If name is an existing part, the existing part is overwritten.
fixedSizeFlag	A Boolean value that indicates whether the part is a fixed-size part. If TRUE, the part becomes a fixed size part. When a fixed size part is placed, it is drawn the same size as when it was created. If fixedSizeFlag

is FALSE, the part is not a fixed-size part and the user can specify the size when placing the part.

Description All objects selected at the time GR_CREATE_PART@ is called are included in the custom part.

See also [GR_CHANGE_PART@](#)

GR/GFX_CURRENT_DOC_PTR_STRING@

Returns the document's full path name

Format pathname = GR_CURRENT_DOC_PTR_STRING@()
pathname = GFX_CURRENT_DOC_PTR_STRING@ (gfx)

Method pathname = this.current_doc_ptr_string@

Arguments gfx A graphics handle.

Description Returns the pathname of the current Graphics document.

GR/GFX_CUT@

Cuts selected objects

Format GR_CUT@()

Method this.cut@
GFX_CUT@(gfx)

Arguments gfx A graphics handle.

Description Cuts selected objects from the current Graphics document and places the objects in the clipboard.

See also [GR_COPY@](#)
[GR_PASTE@](#)

GR/GFX_DEFAULT_LAYER@

Sets the default layer

Format GR_DEFAULT_LAYER@(layerName)
GFX_DEFAULT_LAYER@(gfx, layerName)

Method [this.default_layer@](#)(layerName)

Arguments gfx A graphics handle.
layerName The name of a layer. If this layer does not exist, it is created.

Description Sets the default layer. After a layer is made the default layer, it cannot be deleted. In addition, it is the layer upon which objects are drawn if they are not explicitly tagged with a layer name.

See also [GR GET N LAYERS@](#)

GR/GFX_DEL_CMAP_ENTRY@

Deletes a color from the colormap

Format GR_DEL_CMAP_ENTRY@(index)
GFX_DEL_CMAP_ENTRY@(gfx, index)

Method [this.del_cmap_entry@](#)(index)

Arguments gfx A graphics handle.
index A number indicating the position of the color map entry being deleted.

Description Deletes a color from the current color map. After the color is deleted, all color ribbons are updated.

GR/GFX_DELETE@

Deletes selected objects

Format GR_DELETE@()
GFX_DELETE@(gfx)

Method `this.delete@`

Arguments `gfx` A graphics handle.

Description Deletes all selected objects in the current Graphics document. Deleted objects are not saved in the clipboard. If no objects are selected, an error is thrown.

GR/GFX_DELETE_CHAR_KEY@

Deletes the character following the cursor

Format `GR_DELETE_CHAR_KEY@()`
`GFX_DELETE_CHAR_KEY@(gfx)`

Method `this.delete_char_key@`

Arguments `gfx` A graphics handle.

Description Deletes the character following the cursor or deletes all selected characters.

See also [GR_BACKSPACE_KEY@](#)

GR/GFX_DELETE_DOC@

Deletes the current Graphics document

Format `GR_DELETE_DOC@()`
`GFX_DELETE_DOC@(gfx)`

Method `this.delete_doc@`

Arguments `gfx` A graphics handle.

Description Deletes the current Graphics document and closes the Graphics window.

GR/GFX_DELETE_FILL@

Removes a fill pattern from the fill pattern list (display)

Format `GR_DELETE_FILL@(fillIndex)`
`GFX_DELETE_FILL@(gfx, fillIndex)`

Method `this.delete_fill@(fillIndex)`

Arguments `gfx` A graphics handle.
`fillIndex` A number between 0 and 28 representing the fill pattern shown in the illustration section of this macro.

Description Deletes a fill pattern from the list of fill patterns. The default fill pattern list is as follows:

0	1	13	2	11	3	22	4	14	5
6	7	8	9	10	21	12	19	20	23
24	15	16	18	17	25	26	27	28	

See also [GR SET FILL@](#)

GR/GFX_DELETE_LAYER@

Deletes a layer

Format `GR_DELETE_LAYER@(layerName)`
`GFX_DELETE_LAYER@(gfx, layerName)`

Method `this.delete_layer@(layerName)`

Arguments `gfx` A graphics handle.
`layerName` The name of an already existing layer. If the layer does not exist, no deletions will occur (of course). However, deleting a layer that does not exist does not throw an error.

Description Deletes a layer and all the objects in the layer. The default layer cannot be deleted.

See also [GR CREATE LAYER@](#)

GR/GFX_DELETE_LINE@

Deletes a line in a Graphics document

Format GR_DELETE_LINE@()
GFX_DELETE_LINE@(gfx)

Method [this.delete_line@](#)

Arguments gfx A graphics handle.

Description Deletes all text in the line in which the cursor resides in the current Graphics document. GR_DELETE_LINE@ is relevant only if a line of text is in editing mode in the Graphics document.

GR/GFX_DELETE_PAGE@

Deletes a page

Format GR_DELETE_PAGE@([pageNumber[, forceFlag]])
GFX_DELETE_PAGE@(gfx[, pageNumber[, forceFlag]])

Method [this.delete_page@](#)([pageNumber[, forceFlag]])

Arguments gfx A graphics handle.
pageNumber The page number of the page being deleted. If this argument is omitted, the current page is deleted.
forceFlag A Boolean value which if set to TRUE allows you to delete non-empty pages. If this flag is omitted, the default is FALSE.

Description Deletes a graphic page. If the graphic only contains one page, the page is re-initialized to an empty state.
After a page is deleted, all pages following the deleted page are renumbered so that no page numbering gaps exist.

GR/GFX_DELETE_PREV_CHAR_KEY@

Deletes the character before the cursor

Format GR_DELETE_PREV_CHAR_KEY@()
GFX_DELETE_PREV_CHAR_KEY@(gfx)

Method [this.delete_prev_char_key@](#)

Arguments gfx A graphics handle.

Description Deletes the character preceding the cursor or deletes all selected characters.

See also [GR BACKSPACE KEY@](#)

GR/GFX_DELETE_SELECTED_PART@

Deletes selected custom part

Format GR_DELETE_SELECTED_PART@()
GFX_DELETE_SELECTED_PART@(gfx)

Method [this.delete_selected_part@](#)

Arguments gfx A graphics handle.

Description GR_DELETE_SELECTED_PART@ deletes the part that is currently selected from the custom parts list. If no parts are selected, an error is thrown. Deleted parts are not recoverable.

GR/GFX_DELETE_TO_EOL@

Deletes characters to the end of the line

Format GR_DELETE_TO_EOL@()
GFX_DELETE_TO_EOL@(gfx)

Method [this.delete_to_eol@](#)

Arguments gfx A graphics handle.

Description Deletes all text from the current cursor position to the end of the current line.
GR_DELETE_TO_EOL@ is relevant only if a line of text is in editing mode in the Graphics document.

GR/GFX_DELETE_WORD@

Deletes the word at the cursor position

Format GR_DELETE_WORD@()
GFX_DELETE_WORD@(gfx)

Method [this.delete_word@](#)

Arguments gfx A graphics handle.

Description GR_DELETE_WORD@ is relevant only if a line of text is in editing mode in the Graphics document. If the cursor is in a space, the next word after the space is deleted.

GR/GFX_DESTROY_CALLBACK@

Removes a callback macro

Format GR_DESTROY_CALLBACK@(index)
GFX_DESTROY_CALLBACK@(gfx, index)

Method [this.destroy_callback@\(index\)](#)

Arguments gfx A graphics handle.
index A number indicating which callback macro will be removed from the callback macros list.

Description Deletes a callback macro and its associated information from the list of graphics callback macros. After this macro is deleted, the list of macros is renumbered.

See also [GR_CREATE_CALLBACK@](#)
[GR_GET_CALLBACK@](#)
[GR_GET_CALLBACKS@](#)

GR/GFX_DISABLE_HANDLES@

Disables the display of handles on the current selection

Format GR_DISABLE_HANDLES@(
GFX_DISABLE_HANDLES@(gfx)

Method [this.disable_handles@](#)

Arguments gfx A graphics handle.

Description Disables the display of handles in the current Graphics window on selected. The graphics cursor will continue to change its appearance as it moves around a selection.

See also [GR_ENABLE_HANDLES@](#)

GR/GFX_DISABLE_PRINT_LAYER@

Makes objects on this layer unprintable

Format GR_DISABLE_PRINT_LAYER@(layerName)
GFX_DISABLE_PRINT_LAYER@(gfx, layerName)

Method [this.disable_print_layer@\(layerName\)](#)

Arguments gfx A graphics handle.
layerName The name of a layer. If this layer does not exist, it is created.

Description Makes objects on this layer unprintable. Making objects *unprintable* does not affect the way they are displayed. For additional information, see [GR_ENABLE_PRINT_LAYER@](#).

GR/GFX_DISPLAY_COLOR_PALETTE@

Toggles the display of the color ribbon

Format GR_DISPLAY_COLOR_PALETTE@()
GFX_DISPLAY_COLOR_PALETTE@(gfx)

Method [this.display_color_palette@](#)

Arguments gfx A graphics handle.

Description Turns on and off the display of the color ribbon. (The *color ribbon* is also called the *color palette*.) If it is displayed, it is displayed beneath the drawing area within the Graphics window.

GR/GFX_DISPLAY_TOOL_PALETTE@

Toggles the display of the palette

Format GR_DISPLAY_TOOL_PALETTE@()
GFX_DISPLAY_TOOL_PALETTE@(gfx)

Method [this.display_tool_palette@](#)

Arguments gfx A graphics handle.

Description Turns on and off the display of the tool palette.

GR/GFX_DOWN_ARROW_KEY@

Moves down

Format GR_DOWN_ARROW_KEY@()
GFX_DOWN_ARROW_KEY@(gfx)

Method [this.down_arrow_key@](#)

Arguments gfx A graphics handle.

Description If the cursor is within text, this macro moves the cursor down one line.

If an object or a group is selected, this macro moves the object down by 10 mils. (1000 mils = 1 inch)

See also [GR UP ARROW KEY@](#)

GR/GFX_DRAW_BY_HANDLE@

Redraws objects in the list

Format GR_DRAW_BY_HANDLE@(handleList)
GFX_DRAW_BY_HANDLE@(gfx, handleList)

Method [this.draw_by_handle@](#)(handleList)

Arguments gfx A graphics handle.
handleList A list of object handles. Typically, this list is created using [GR_QUERY@](#).

Description Draws (or redraws) the graphic objects in the handleList.

GR/GFX_DRAW_PROPORTIONAL@

Allows for proportional drawing/resizing of an object

Format GR_DRAW_PROPORTIONAL@()
GFX_DRAW_PROPORTIONAL@(gfx)

Method [this.draw_proportional@](#)

Arguments gfx A graphics handle.

Description Any object drawn immediately after GR_DRAW_PROPORTIONAL@ is executed will have an equal height and width. When an object is resized using a corner handle after GR_DRAW_PROPORTIONAL@ is executed, the object is resized equally in both the horizontal and vertical direction.

GR_DRAW_PROPORTIONAL@ is only effective for a single drawing or resizing. If you perform an action other than drawing or resizing an object after executing GR_DRAW_PROPORTIONAL@, the proportional drawing attribute is ended. If a side handle rather than a corner handle is used for resizing, no proportional resizing is performed.

GR/GFX_DUPLICATE_SELECTED@

Creates a duplicate of selected objects

Format GR_DUPLICATE_SELECTED@()
GFX_DUPLICATE_SELECTED@(gfx)

Method [this.duplicate_selected@](#)

Arguments gfx A graphics handle.

Description Creates a duplicate of any selected objects and draws it in the current Graphics document. The duplicate is placed slightly down and to the right of the original object.

GR/GFX_EDIT_ARC@

Places the selected ellipse or arc in edit mode

Format GR_EDIT_ARC@()
GFX_EDIT_ARC@(gfx)

Method [this.edit_arc@](#)

Arguments gfx A graphics handle.

Description Lets you alter the extent of the arc of an ellipse or an arc. While in edit mode, two arc handles are displayed on an ellipse or arc. You can move the arc handles to create or change the shape of an arc.

If no ellipses or arcs are selected, an error is thrown.

GR/GFX_EDIT_COLOR@

Changes a color's definition

Format GR_EDIT_COLOR@(index, cyan, magenta, yellow, black, maxVal, transparent, type, name)
GFX_EDIT_COLOR@(gfx, index, cyan, magenta, yellow, black, maxVal, transparent, type, name)

Method `this.edit_color@(index, cyan, magenta, yellow, black, maxVal, transparent, type, name)`

Arguments

<code>gfx</code>	A graphics handle.
<code>index</code>	A color's index within the current color map.
<code>cyan</code>	The color's cyan component.
<code>magenta</code>	The color's magenta component.
<code>yellow</code>	The color's yellow component.
<code>black</code>	The color's black component.
<code>maxVal</code>	A number from 0 to 255.
<code>transparent</code>	A Boolean value where TRUE indicates that the color is transparent.
<code>type</code>	Not used.
<code>name</code>	An arbitrary named assigned to the color

Description Changes a color within the default color map. The color palette and the color selection dialog box (if displayed) will be updated.

GR_EDIT_HANDOUT_MASTER@

Places the Presents window into Handout Master editing mode

Format `GR_EDIT_HANDOUT_MASTER@()`

Method `this.edit_handout_master@`

Description Allows you to edit the handout master for your presentation. In Presents, the Handout Master defines text and objects that you want to include with each page of paper handouts. Handouts show 2, 3 or 6 slides per page.

See also `GFX_EDIT_HANDOUT_MASTER@`

GR_EDIT_OUTLINE@

Places the Presents window into Outline editing mode

Format `GR_EDIT_OUTLINE@()`

Method `this.edit_outline@`

Description Allows you to edit the outline text for your presentation. In Graphics presents, the outliner is used to organize information for your presentation.

See also GFX_EDIT_OUTLINE@

GR_EDIT_OUTLINE_MASTER@

Places the Presents window into Outline Master editing mode

Format GR_EDIT_OUTLINE_MASTER@()

Method [this.edit_outline_master@](#)

Description Allows you to edit the outline master for your presentation. In Graphics presents, the outliner is used to organize information for your presentation. The outline master provides the structure for the outline document of your presentation.

See also GFX_EDIT_OUTLINE_MASTER@

GR/GFX_EDIT_PIXELS@

Places the Graphics window into pixel editing mode

Format GR_EDIT_PIXELS@()
GFX_EDIT_PIXELS@(gfx)

Method [this.edit_pixels@](#)

Arguments gfx A graphics handle.

Description If an image is selected when GR_EDIT_PIXELS@ is executed, the image is displayed in pixel editing mode. If no images are selected, an error is thrown.

GR/GFX_EDIT_SIDES@

Places a rectangle, polyline, line, or free form shape in edit mode

Format GR_EDIT_SIDES@()
GFX_EDIT_SIDES@(gfx)

Method `this.edit_sides@`

Arguments `gfx` A graphics handle.

Description While in edit mode, editing handles appear on an object. You can move, delete, or add handles to reshape the object. If no rectangles, polylines, lines, or free form shapes are selected, an error is thrown.

GR_EDIT_SLIDE@

Places the Presents window into slide editing mode

Format `GR_EDIT_SLIDE@()`

Method `this.edit_slide@`

Description Allows you to edit a slide in your presentation. In Presents, a slide is a single page of a presentation. There can be many slides in a presentation.

See also `GFX_EDIT_SLIDE@`

GR_EDIT_SLIDE_MASTER@

Places the Presents window into slide master editing mode

Format `GR_EDIT_SLIDE_MASTER@()`

Method `this.edit_slide_master@`

Description Allows you to edit a slide in your presentation. In Presents, a slide is a single page of a presentation. The slide master provides the default text format for all slides in a presentation.

See also `GFX_EDIT_SLIDE_MASTER@`

GR_EDIT_SPEAKER_NOTES@

Allows you to edit Speaker Notes in Presents

Format GR_EDIT_SPEAKER_NOTES@()

Method [this.edit_speaker_notes@](#)

Description In Applixware Presents, speaker notes are used to structure your spoken remarks on each slide in your presentation. GR_EDIT_SPEAKER_NOTES@ allows you to edit the speaker notes for the current slide.

See also GFX_EDIT_SPEAKER_NOTES@

GR_EDIT_SPEAKER_NOTES_MASTER@

Allows you to edit the Speaker Notes Master in Presents

Format GR_EDIT_SPEAKER_NOTES_MASTER@()

Method [this.edit_speaker_notes_master@](#)

Description Allows you to edit the master document for the speaker notes in Applixware Presents. Speaker notes allow you to record verbal comments to accompany each slide in your presentation. The speaker notes master provides the format for the speaker notes in a presentation.

See also GFX_EDIT_SPEAKER_NOTES_MASTER@

GR/GFX_EDIT_TEXT@

Allows you to edit text

Format GR_EDIT_TEXT@()
GFX_EDIT_TEXT@(gfx)

Method [this.edit_text@](#)

Arguments gfx A graphics handle.

Description If a text block is selected, displays a cursor within the text block, allowing you to alter the text within the block.

GR_EMBED_INSET@

Embeds an inset in a graphics document

Format GR_EMBED_INSET@ (Url, Exec_macro, Filt_macro, As_icon, Source)

Arguments

Url	The absolute pathname or URL of the file to import as an inset.
Exec_macro	The macro to execute when this inset is double-clicked in a Graphics document.
Filt_macro	The macro to run to convert a foreign file into an Applixware document.
As_icon	A Boolean. If TRUE, the inset appears in the graphics document as an icon. If FALSE, a portion of the inset itself is displayed.
Source	An array. This argument can be set to NULL.

Description Embeds a file in a Graphics document.

GR/GFX_ENABLE_HANDLES@

Enables the display of handles on the current selection

Format GR_ENABLE_HANDLES@(
GFX_ENABLE_HANDLES@(gfx)

Method [this.enable_handles@](#)

Arguments gfx A graphics handle.

Description Enables the display of handles in the current Graphics window on selected.

See also [GR_DISABLE_HANDLES@](#)

GR/GFX_ENABLE_PRINT_LAYER@

Allows objects on a layer to be printed

Format GR_ENABLE_PRINT_LAYER@(layerName)

GFX_ENABLE_PRINT_LAYER@(gfx, layerName)

Method [this.enable_print_layer@\(layerName\)](#)

Arguments gfx A graphics handle.
layerName The name of a layer. If this layer does not exist, it is created.

Description Allows all the objects on a layer to be printed. For additional information, see [GR_DISABLE_PRINT_LAYER@](#).

GR/GFX_ENTER_HELP_MODE@

Enters 'help on context' mode

Format GR_ENTER_HELP_MODE@()
GFX_ENTER_HELP_MODE@(gfx)

Method [this.enter_help_mode@](#)

Arguments gfx A graphics handle.

Description Places Graphics into a state where clicking on an icon, tool, or a menu item invokes help rather than invoking the command associated with the icon, tool, or menu item.

GR/GFX_ESCAPE_KEY@

Cancels the current operation or exits from a dialog box

Format GR_ESCAPE_KEY@()
GFX_ESCAPE_KEY@(gfx)

Method [this.escape_key@](#)

Arguments gfx A graphics handle.

GR/GFX_EXIT@

Ends the current graphics interaction

Format GR_EXIT@()

GFX_EXIT@(gfx)

Method this.exit@

Arguments gfx A graphics handle.

Description Closes the current graphics window. If the contents of the window are modified, these changes are abandoned; that is, you are not given any opportunity to save the information before the window is closed.

GR_EXPORT_?@

Exports an Applixware Graphics file

Format GR_EXPORT_?@(dstName, depth, grayFlag, res)

Method this.export_?@(dstName, depth, grayFlag, res)

Arguments dstName The name of the file to which the exported file will be written.
depth The number of bits used to express color. The default is 8 bits. Other valid values are 1 (black and white) and 24 (which is often called true color).
grayFlag A Boolean value where TRUE means convert the objects into an 8-bit gray scale.
res The resolution of the exported image in dots per inch (DPI). The default value is 75 DPI.

Description Exports an Applixware Graphics file to another format. The bold macros cannot be used without first obtaining a Graphics Pack license from Applixware. This family of macros is as follows:

GR_EXPORT_CGM@	Exports to CGM format
GR_EXPORT_EPS@	Exports to encapsulated postscript
GR_EXPORT_FAXI@	Exports to Group 3 FAX (lsb) format
GR_EXPORT_FAXM@	Exports to Group 3 FAX (msb) format
GR_EXPORT_GIF@	Exports to GIF format. This macro is obsolete and will be removed in a future release. Please use GR_EXPORT_GIF87@ or GR_EXPORT_GIF89@ for all new programming.
GR_EXPORT_GIF87@	Exports to GIF 87 format
GR_EXPORT_GR31@	Exports to Applixware 3.1 Graphics format
GR_EXPORT_GR400@	Exports to Applixware 4.0 Graphics format
GR_EXPORT_HPGL@	Exports to HPGL format
GR_EXPORT_IM@	Exports to Applixware Bitmap format

GR_EXPORT_IRIS@	Exports to Silicon Graphics IRIS format
GR_EXPORT_JPEG@	Exports to JPEG format
GR_EXPORT_JPEG50@	Exports to JPEG 50 format
GR_EXPORT_JPEG75@	Exports to JPEG 75 format
GR_EXPORT_JPEG90@	Exports to JPEG 90 format
GR_EXPORT_MSWINBM@	Exports to Microsoft Windows Bitmap format
GR_EXPORT_PBM@	Exports to X11 Portable Bitmap format
GR_EXPORT_PGM@	Exports to X11 Portable Greymap format
GR_EXPORT_PPM@	Exports to X11 Portable Pixmap format
GR_EXPORT_PPT@	Exports to Microsoft PowerPoint 97 format
GR_EXPORT_RS@	Exports to Sun Raster format
GR_EXPORT_TIFFI@	Exports to TIFFI format
GR_EXPORT_TIFFM@	Exports to TIFFM format
GR_EXPORT_WMF@	Exports to Windows MetaFile
GR_EXPORT_XBM@	Exports to X Windows Bitmap format
GR_EXPORT_XWD@	Exports to X Windows Dump format

GR/GFX_EXPORT_IMAGE@

Saves a Graphics document as a bitmap image

Format GR_EXPORT_IMAGE@(graphics, bitmap)
 GFX_EXPORT_IMAGE@(gfx, graphics, bitmap)

Method [this.export_image@](#)(graphics, bitmap)

Arguments

gfx	A graphics handle.
graphics	The full path name, a string, of the file you want to save as a bitmap image.
bitmap	The name, a string, of the file in which you want to save the bitmap image. The file name must include the .im extension. If you do not provide a path name, bitmap is saved in the current directory.

Description Saves a copy of the Graphics document you specify as a bitmap image. The original Graphics document is not affected in any way. The resulting bitmap image can be edited using the Bitmap Editor.

GR/GFX_EXTRUDE@

Creates a 3D solid from a 2D object.

Format GR_EXTRUDE@(xOffset, yOffset)
GFX_EXTRUDE@(gfx, xOffset, yOffset)

Method [this.extrude@](#)(xOffset, yOffset)

Arguments

gfx	A graphics handle.
xOffset	The offset of the added plane, in the default units of the window. A negative number places the plane to the left of the original, a positive number places it to the right of the original.
yOffset	The offset of the added plane, in the default units of the window. A negative number places the plane above the original, a positive number places it below the original.

Description Transforms a 2D object into a 3D object by taking a planar object and adding another plane to it.

GR/GFX_FIRST_PAGE@

Moves to the first Graphics page

Format GR_FIRST_PAGE@(
GFX_FIRST_PAGE@(gfx)

Method [this.first_page@](#)

Arguments gfx A graphics handle.

Description Changes the display so that the first Graphics page is displayed.

GR/GFX_FIRST_SCREEN_KEY@

Displays the upper left part of the drawing area

Format GR_FIRST_SCREEN_KEY@()
GFX_FIRST_SCREEN_KEY@(gfx)

Method `this.first_screen_key@`

Arguments `gfx` A graphics handle.

Description Displays the upper left corner. This macro is bound to Key ® Beginning of Document, which has the ^A accelerator key.

GR/GFX_FIT_PAGE_IN_WINDOW@

Redisplays to fit window

Format `GR_FIT_PAGE_IN_WINDOW@()`
`GFX_FIT_PAGE_IN_WINDOW@(gfx)`

Method `this.fit_page_in_window@`

Arguments `gfx` A graphics handle.

Description Redraws the graphics page so that an entire page and its contents can be displayed within the window.

GR/GFX_FIT_TEXT_IN_PATH@

Fits the text within the boundary

Format `GR_FIT_TEXT_IN_PATH@()`
`GFX_FIT_TEXT_IN_PATH@(gfx)`

Method `this.fit_text_in_path@`

Arguments `gfx` A graphics handle.

Description Fits text inside the contours of an object. Both the text and the object must be selected before you execute this macro.

After this macro executes, the text displays inside of the selected object and wraps to fit within it.

This macro is bound to Text ® Fit In Path.

To fit text around the outside of the contours of an object, see [**GR FIT TEXT TO PATH@**](#).

GR/GFX_FIT_TEXT_TO_PATH@

Fits the text along the curved boundary of a path

Format GR_FIT_TEXT_TO_PATH@()

Method this.fit_text_to_path@
GFX_FIT_TEXT_TO_PATH@(gfx)

Arguments gfx A graphics handle.

Description Fits text around the outside of the contours of an object. Both the text and object must be selected before you execute this macro.

After this macro executes, the text displays in the shape of the selected object outside of the edges of the object.

This macro is bound to Text ® Fit To Path.

To fit text inside the contours of an object, see [GR FIT TEXT IN PATH@](#).

GR/GFX_GET_AREA_BY_HANDLE@

Returns the area around objects in the 'handleList'

Format format gr_area@ area = GR_GET_AREA_BY_HANDLE@ (handleList)
format gr_area@ area = GFX_GET_AREA_BY_HANDLE@ (gfx, handleList)

Method format gr_area@ area = this.get_area_by_handle@ (handleList)

Arguments gfx A graphics handle.
handleList A list of object handles. This list is usually created using the [GR QUERY@](#) macro.

Description Returns a rectangular region that defines the smallest bounding box that can surround the objects in handleList.

The definition of the returned gr_area@ format is as follows:

format gr_area@
 x,
 y,

width,
height

GR/GFX_GET_ATTR@

Gets the current attributes

Format format gr_attribute@ info = GR_GET_ATTR@()
format gr_attribute@ info = GFX_GET_ATTR@(gfx)

Method format gr_attribute@ info = this.get_attr@

Arguments gfx A graphics handle.

Description Returns a format containing backfill, line fill, line, and text attributes.
For information on the gr_attribute@ format, see [GR_CHART_GET_ATTR@](#).

See also [GR_SET_ATTR@](#)

GR/GFX_GET_BACKFILL@

Returns the backfill attribute

Format format gr_fill_attr_type@ backFillAttr = GR_GET_BACKFILL@(
format gr_fill_attr_type@ backFillAttr = GFX_GET_BACKFILL@(gfx)

Method format gr_fill_attr_type@ backFillAttr = this.get_backfill@

Arguments gfx A graphics handle.

Description Returns the backfill attribute. The definition of the gr_fill_attr_type@ format is as follows:

format gr_fill_attr_type@ backfill
type, 'String: built-in number, file name, linear gradient
fg_color, 'color name in colormap
bg_color, 'color name in colormap
angle, 'in degrees
offset 'in mils (1000 mils = 1 inch)

GR/GFX_GET_BOUNDER@

Return the dimensions of a bounding box surrounding the selection

Format format gr_area@ info = GR_GET_BOUNDER@ ()
format gr_area@ info = GFX_GET_BOUNDER@(gfx)

Method format gr_area@ info = this.get_bounder@

Arguments gfx A graphics handle.

Description Returns a gr_area@ formatted variable that defines a rectangle that is just large enough to enclose the current selection.

The definition of the gr_area@ format is as follows:

```
format gr_area@
    x,
    y,
    width,
    height
```

GR/GFX_GET_CALLBACK@

Identifies the callback associated with a selection

Format index = GR_GET_CALLBACK@()
index = GFX_GET_CALLBACK@(gfx)

Method index = this.get_callback@

Arguments gfx A graphics handle.

Description Returns an index indicating which callback macro is associated with the current selection. This index represents the position within the array of gr_callback_info@ information returned by **GR GET CALLBACKS@**.

If the selection has more than one callback macro associated with it, a NULL value is returned.

GR/GFX_GET_CALLBACKS@

Returns information defining stored callback macros

Format format arrayof gr_callback_info@ info = GR_GET_CALLBACKS@()
format arrayof gr_callback_info@ info = GFX_GET_CALLBACKS@(gfx)

Method format arrayof gr_callback_info@ info = this.get_callbacks@

Arguments gfx A graphics handle.

Description Returns an array of gr_callback_info@ structures. Each member of this array contains information describing a callback macro that has been associated with a Graphics document.

The definition of the gr_callback_info@ format is as follows:

```
format gr_callback_info@
    macro_name,
    trigger_on_drag,
    trigger_on_single_click,
    trigger_on_double_click,
    argument_list
```

GR/GFX_GET_CHAR_ATTS@

Returns the currently defined character attributes

Format format gr_text_char_attr_type@ info = GR_GET_CHAR_ATTS@()
format gr_text_char_attr_type@ info = GFX_GET_CHAR_ATTS@(gfx)

Method format gr_text_char_attr_type@ info = this.get_char_atts@

Arguments gfx A graphics handle.

Description Returns a gr_text_char_attr_type@ format variable that defines the default character attributes. The definition of this format is as follows:

```
format gr_text_char_attr_type@
    face,
    size,
    ruling,
    strike_thru,
```

bold,
italic,
horizontal_scale,
shear,
angle,
horizontal_offset,
vertical_offset,
horizontal_sub_sup,
vertical_sub_sup

See also [GR SET CHAR ATTS@](#)

GR/GFX_GET_COLOR_INFO@

Returns information defining a color in the current color map

Format format gr_color@ = GR_GET_COLOR_INFO@(colorIndex)
format gr_color@ = GFX_GET_COLOR_INFO@(gfx, colorIndex)

Method format gr_color@ = this.get_color_info@(colorIndex)

Arguments gfx A graphics handle.
colorIndex A number indicating a color's position in the current color map.

Description Returns a gr_color@ format variable that defines the color pointed to by colorIndex in the current color map. The definition of gr_color@ is as follows:

format gr_color@

c,	'Cyan
m,	'Magenta
y,	'Yellow
k,	'Black
see_thru,	'A Boolean value where TRUE indicates that the color is transparent
ink_type,	'Not used
name	'An arbitrary name assigned to the color

GR/GFX_GET_CURRENT_CHART@

Returns the current chart's name

Format name = GR_GET_CURRENT_CHART@()

name = GFX_GET_CURRENT_CHART@(gfx)

Method name = this.get_current_chart@

Arguments gfx A graphics handle.

GR/GFX_GET_DOC_INFO@

Returns an array of information on a Graphics document

Format format doc_format_info = GR_GET_DOC_INFO@()
format doc_format_info = GFX_GET_DOC_INFO@(gfx)

Method format doc_format_info = this.get_doc_info@

Arguments gfx A graphics handle.

Description Returns an array giving the following information on the current Graphics document. The FORMAT template for the array information is named doc_format_. The header file containing this template is fileinf_.am.

name	The full path name of the current document.
docid	The document's unique ID number.
on_disk	Indicates whether the document has been saved or read using the file name specified by name. Returns TRUE if the document has been saved or read. Returns FALSE if it hasn't.
save_mode	The mode in which the document has been saved. save_mode can be one of the following: 0 binary 1 normal 2 TIFF I 3 TIFF M
grp_access	The permission setting for the file for members of the same group. grp_access can be one of the following: 0 none 1 read 2 read/write
writeable	Indicates whether the Graphics file is writeable or read-only. Returns TRUE if the user has write access to the file. Returns FALSE if the file is read-only.

GR/GFX_GET_DPI@

Returns the dots per inch x 1000

Format 1000 = GR_GET_DPI@()
1000 = GFX_GET_DPI@(gfx)

Method 1000 = this.get_dpi@

Arguments gfx A graphics handle.

Description Returns the number of dots per inch. At this time, this macro always returns 1000.

See also [GR SET DPI@](#)

GR/GFX_GET_ELL@

Returns starting and ending angles of an ellipse

Format format gr_ellipse_info@ info = GR_GET_ELL@()
format gr_ellipse_info@ info = GFX_GET_ELL@(gfx)

Method format gr_ellipse_info@ info = this.get_ell@

Arguments gfx A graphics handle.

Description Returns a gr_ellipse_info@ variable whose contents indicate the starting and ending angles of the selected ellipse. The definition of the gr_ellipse_info@ format is as follows:

```
format gr_ellipse_info@
    angle1,      'The starting angle
    angle2      'The ending angle
```

GR/GFX_GET_FIELD_ATTS@

Returns information defining the way space is managed around text

Format format gr_text_field_attr_type@ info = GR_GET_FIELD_ATTS@()
format gr_text_field_attr_type@ info = GFX_GET_FIELD_ATTS@(gfx)

Method format gr_text_field_attr_type@ info = this.get_field_atts@

Arguments gfx A graphics handle.

Description Returns a `gr_text_field_attr_type@` variable whose contents define the way Graphics uses the space within a text area. The definition of the `gr_text_field_attr_type@` format is as follows:

```
format gr_text_field_attr_type@
    horizontal_alignment,
    vertical_alignment,
    line_space,
    horizontal_scale, 'A number between -1 and 1'
    vertical_scale,   'A number between -1 and 1'
    shear,           'A number between -1 and 1'
    angle,           'A rotation angle'
    left_margin,
    right_margin,
    top_margin,
    bottom_margin
```

For more information, on field settings, see the [Whole Field Settings](#) on-line help.

GR/GFX_GET_FILL_NAME@

Returns the name of a fill pattern

Format name = GR_GET_FILL_NAME@(index)
name = GFX_GET_FILL_NAME@(gfx, index)

Method name = this.get_fill_name@(index)

Arguments gfx A graphics handle.
index A number indicating the fill pattern.

GR/GFX_GET_FILL_STYLE@

Returns the fill style

Format styleType = GR_GET_FILL_STYLE@(index)
styleType = GFX_GET_FILL_STYLE@(gfx, index)

Method styleType = this.get_fill_style@(index)

Arguments gfx A graphics handle.
 index A number indicating the desired fill pattern.

Description Returns a string indicating the fill style. This string is one of the following:

- pattern
- texture

GR/GFX_GET_FONT@

Returns the current font name

Format fontName = GR_GET_FONT@()
 fontName = GFX_GET_FONT@(gfx)

Method fontName = this.get_font@

Arguments gfx A graphics handle.

Description Returns a string that specifies the current font. Use [GR_LIST_FONT_FAMILIES@](#) to obtain a list of all Graphics fonts.

GR/GFX_GET_GRID_FACTOR@

Returns the grid setting in the current unit (centimeter or inches)

Format factor = GR_GET_GRID_FACTOR@()
 factor = GFX_GET_GRID_FACTOR@(gfx)

Method factor = this.get_grid_factor@

Arguments gfx A graphics handle.

Description Returns a numeric value that indicates the grid spacing setting; that is, this number indicates the distance from one grid line to another.

GR/GFX_GET_GUIDES@

Returns grid, ruler, and other guide attributes

Format format gr_guides_ info = GR_GET_GUIDES@()

Method format gr_guides_ info = this.get_guides@
format gr_guides_ info = GFX_GET_GUIDES@(gfx)

Arguments gfx A graphics handle.

Description Returns the following information about guides in the current Graphics document. (The FORMAT template for GR_GET_GUIDES@ is named gr_guides_ . The header file containing the FORMAT template is graphics_.am.)

rulers	Indicates whether horizontal and vertical rulers are displayed on the screen. Returns 1 if rulers are displayed. Returns 0 if rulers are not displayed.
grids	Indicates whether horizontal and vertical grid lines are displayed on the screen. Returns 1 if grid lines are displayed. Returns 0 if grid lines are not displayed.
coordinates	Indicates whether coordinates are displayed when an object is drawn or moved. Returns 1 if coordinates are displayed. Returns 0 if coordinates are not displayed.
auto_gridding	Indicates whether grid snap is enabled. Returns 1 if grid snap is enabled, returns 0 if it is not enabled.
sticky_points	Indicates whether sticky points are enabled. Returns 1 if sticky points are enabled, returns 0 if not enabled.
page_breaks	Returns 1 if page breaks are displayed in the document. Returns 0 if page breaks are not displayed.
target_viewmode	Not used. Always returns 0.
grid_size	Indicates the following scales used for grids and rulers: 0 1/16 inch 1 1/10 inch 2 1/8 inch 3 1/4 inch

- 4 1/2 inch
- 5 centimeter scale

device_setup

Not used. Always returns "Printer".

See also [GR GUIDES@](#)

GR/GFX_GET_HEADERS_AND_FOOTERS@

Returns format and contents of the page header and footer

Format format hdrftr_info info = GR_GET_HEADERS_AND_FOOTERS@()
 format hdrftr_info info = GFX_GET_HEADERS_AND_FOOTERS@(gfx)

Method format hdrftr_info info = this.get_headers_and_footers@

Arguments gfx A graphics handle.

Description Returns a hdrftr_info format that contains the format and contents of the page header and footer. The definition of this format is:

```
format hdrftr_info
  special_even_odd,
  special_first,
  special_final,
  format hdrftr_pg_info base,
  format hdrftr_pg_info even,
  format hdrftr_pg_info odd,
  format hdrftr_pg_info first,
  format hdrftr_pg_info final
format hdrftr_pg_info
  hdr_offset,
  ftr_offset,
  format hdrftr_line_info hdr,
  format hdrftr_line_info ftr,
format hdrftr_line_info
  format hdrftr_group_inner,
  format hdrftr_group_cent,
  format hdrftr_group_outer
```

format	hdrftr_group_	
	text,	'text of the header/footer segment
	face,	'a string
	size,	'a string
	color	'a string
	bold	'true/false
	italic,	'true/false
	strike,	'true/false
	underline,	'0-none, 1-single, 2-double
	word_underline	'true/false, word underline only

GR/GFX_GET_HOOK@

Returns the macro that is run when Graphics is started

Format macroName = GR_GET_HOOK@ ()
 macroName = GFX_GET_HOOK@(gfx)

Method macroName = this.get_hook@

Arguments gfx A graphics handle.

Description Returns the name of the hook macro defined for the current Graphics document. This name was *registered* with Graphics using **GR SET HOOK@**.

GR/GFX_GET_LAYER_INFO@

Returns information describing the current layer

Format format gr_layer@ info = GR_GET_LAYER_INFO@(id)
 format gr_layer@ info = GFX_GET_LAYER_INFO@(gfx, id)

Method format gr_layer@ info = this.get_layer_info@(id)

Arguments gfx A graphics handle.
 id A number indicating the layer for which you want information.

Description Returns a gr_layer@ format containing information on the current layer. The definition of gr_layer@ is as follows:

format gr_layer@
 name,

locked,
hidden,
n_parts,
printable

GR/GFX_GET_LINEFILL@

Returns the line fill attribute

Format format gr_fill_attr_type@ linefillAtr = GR_GET_LINEFILL@ ()
format gr_fill_attr_type@ linefillAtr = GFX_GET_LINEFILL@(gfx)

Method format gr_fill_attr_type@ linefillAtr = this.get_linefill@

Arguments gfx A graphics handle.

Description Returns the linefill attribute. The definition of the gr_fill_attr_type@ format is as follows:

format gr_fill_attr_type@ linefill
type, 'String: built-in number, file name, linear gradient
fg_color, 'color name in colormap
bg_color, 'color name in colormap
angle, 'in degrees
offset 'in mils (1000 mils = 1 inch)

See also [GR SET LINE FILL@](#)

GR/GFX_GET_MARKED_POSITION@

Returns the current selection's upper left X and Y coordinates

Format format gr_point@ coord = GR_GET_MARKED_POSITION@()
format gr_point@ coord = GFX_GET_MARKED_POSITION@(gfx)

Method format gr_point@ coord = this.get_marked_position@

Arguments gfx A graphics handle.

Description Returns a gr_point@ variable whose contents indicate the upper left X and Y coordinates of the current selection. The definition of the gr_point@ format is as follows:

format gr_point@
x,
y

GR/GFX_GET_MOUSE_POSITION@

Returns the X and Y coordinates of the mouse

Format posArray = GR_GET_MOUSE_POSITION@ ()
posArray = GFX_GET_MOUSE_POSITION@(gfx)

Method posArray = this.get_mouse_position@

Arguments gfx A graphics handle.

Description Returns a two-element array containing the X- and Y-coordinates of the mouse's cursor, as follows:
posArray[0] The X-coordinate in screen units.
posArray[1] The Y-coordinate in screen units.

GR/GFX_GET_N_COLORS_IN_CMAP@

Returns the number of entries in the color map

Format num = GR_GET_N_COLORS_IN_CMAP@ ()
num = GFX_GET_N_COLORS_IN_CMAP@(gfx)

Method num = this.get_n_colors_in_cmap@

Arguments gfx A graphics handle.

Description Returns a number specifying the number of entries in the color map. That is, this is the size of the color map.

See also [GR DEL CMAP ENTRY@](#)
[GR LOAD COLORMAP FILE@](#)
[GR SET COLORMAP@](#)
[GR SET COLOR BY CMAP INDEX@](#)

GR/GFX_GET_N_FILLS@

Returns the number of user-defined fill patterns

Format num = GR_GET_N_FILLS@()
num = GFX_GET_N_FILLS@(gfx)

Method num = this.get_n_fills@

Arguments gfx A graphics handle.

GR/GFX_GET_N_LAYERS@

Returns the number of layers and the default layer

Format listArray = GR_GET_N_LAYERS@()
listArray = GFX_GET_N_LAYERS@(gfx)

Method listArray = this.get_n_layers@

Arguments gfx A graphics handle.

Description Returns a two-element array, as follows:

listArray[0] The number of layers
listArray[1] The default layer

See also [GR_DEFAULT_LAYER@](#)

GR/GFX_GET_N_PAGES@

Returns the number of pages

Format num = GR_GET_N_PAGES@()
num = GFX_GET_N_PAGES@(gfx)

Method num = this.get_n_pages@

Arguments gfx A graphics handle.

GR/GFX_GET_N_PAL_PARTS@

Returns the number of user-defined palette parts

Format num = GR_GET_N_PAL_PARTS@()
num = GFX_GET_N_PAL_PARTS@(gfx)

Method num = this.get_n_pal_parts@

Arguments gfx A graphics handle.

GR/GFX_GET_NEWEST_PART@

Returns the ID of the most recently created object

Format numID = GR_GET_NEWEST_PART@()
numID = GFX_GET_NEWEST_PART@(gfx)

Method numID = this.get_newest_part@

Arguments gfx A graphics handle.

Description Returns an integer that represents the ID of the most recently created object within Graphics. This ID will be associated with this object only during the current session.

GR/GFX_GET_PAGE@

Returns the current page number

Format pageNum = GR_GET_PAGE@()
pageNum = GFX_GET_PAGE@(gfx)

Method pageNum = this.get_page@

Arguments gfx A graphics handle.

GR/GFX_GET_PAGE_SETUP@

Returns print setup attributes for a Graphics document

Format format page_setup_info = GR_GET_PAGE_SETUP@()
format page_setup_info = GFX_GET_PAGE_SETUP@(gfx)

Method format page_setup_info = this.get_page_setup@

Arguments gfx A graphics handle.

Description Returns the following print setup information for the current spreadsheet. (The FORMAT template for GR_GET_PAGE_SETUP@ is named page_setup_. The header file containing this template is hdrftr_.am.)

height The height of the document.

width The width of the document.

landscape Indicates the orientation of the document. Returns 0 for portrait orientation, 1 for landscape orientation.

See also [GR SET PAGE SETUP@](#)

GR/GFX_GET_PALETTE_LINE@

Returns the width of a palette's line for settings 1, 2, 3

Format width = GR_GET_PALETTE_LINE@(lineIndex)
width = GFX_GET_PALETTE_LINE@(gfx, lineIndex)

Method width = this.get_palette_line@(lineIndex)

Arguments gfx A graphics handle.
lineIndex A number (values 1, 2, or 3) indicating one of the three displayed lines in the palette. See [Line Width](#) for more information.

GR/GFX_GET_PAL_PART_NAME@

Returns the name of a user-defined item on the palette

Format name = GR_GET_PAL_PART_NAME@(index)

name = GFX_GET_PAL_PART_NAME@(gfx, index)

Method name = this.get_pal_part_name@(index)

Arguments gfx A graphics handle.
index A number indicating a user-defined item on the palette.

GR_GET_PRINT_INFO@

Retrieves print information for a graphic file

Format format print_info@ setup = GR_GET_PRINT_INFO@()
format print_info@ setup = GFX_GET_PRINT_INFO@(gfx)

Method this.info@ setup = GR_GET_PRINT_INFO@()

Arguments gfx A graphics handle.

Description Returns an array of format print_info@, which describes how the current Graphics document is printed. The definition of this format is:

```
format print_info@  
  height,  
  width,  
  upper_margin,  
  lower_margin,  
  inner_margin,  
  outer_margin,  
  bleed,  
  landscape,  
  crop_marks,  
  registration_marks,  
  content_border,  
  printer,  
  printable_x_pos,  
  printable_y_pos,  
  printable_width,  
  printable_height
```

See also [**GR_SET_PRINT_INFO@**](#)

GR/GFX_GET_RPOLY@

Returns the starting angle and number of sides for a regular polygon

Format format gr_regular_poly_info@ info = GR_GET_RPOLY@()
format gr_regular_poly_info@ info = GFX_GET_RPOLY@(gfx)

Method format gr_regular_poly_info@ info = this.get_rpoly@

Arguments gfx A graphics handle.

Description Returns a gr_regular_poly_info@ variable whose contents describe the format of the regular polygon. The definition of gr_regular_info@ is as follows:

```
format gr_regular_info@
    start_angle,
    n_sides
```

GR/GFX_GET_RRECT@

Returns information about a rounded rectangle

Format format gr_rect_info@ info = GR_GET_RRECT@()
format gr_rect_info@ info = GFX_GET_RRECT@(gfx)

Method format gr_rect_info@ info = this.get_rrect@

Arguments gfx A graphics handle.

Description Returns a format that describes the scale and radius of rounded rectangles. The definition of gr_rect_info@ is as follows:

```
format gr_rect_info@
    scale_corners,
    x_corner_radius,
    y_corner_radius
```

See also [GR SET RRECT@](#)

GR/GFX_GET_SESSION_HOST@

Returns the session task id

Format taskID = GR_GET_SESSION_HOST@()
taskID = GFX_GET_SESSION_HOST@(gfx)

Method taskID = this.get_session_host@

Arguments gfx A graphics handle.

Description Returns the host task's program ID (PID) and converts it to a unique task id.
Use the task id returned by this macro as an argument within **SET MACRO PARENT TASK@**. In this way you can set the macro parent task to be different tasks at different times.

GR/GFX_GET_SHADOW@

Returns the shadow attribute

Format format gr_shadow_attr_type@ shadow = GR_GET_SHADOW@()
format gr_shadow_attr_type@ shadow = GFX_GET_SHADOW@(gfx)

Method format gr_shadow_attr_type@ shadow = this.get_shadow@

Arguments gfx A graphics handle.

Description Returns a format that describes the attributes of an object's shadow. The definition of gr_shadow_attr_type@ is as follows:

```
format gr_shadow_attr_type@
type,          ' string: none, background drop shadow, local drop shadow
color,         ' string: name of color in colormap
               ' int: in mils (1000 mils = 1 inch)
horizontal_offset,
vertical_offset
               'int: in mils
```

See also **GR SET SHADOW@**

GR_GET_SLIDE_BG@

Returns information on the background of the current slide

Format format gr_background@ infoArray = GR_GET_SLIDE_BG@()
 format gr_background@ infoArray = GFX_GET_SLIDE_BG@(gfx)

Method format gr_background@ infoArray = this.get_slide_bg@

Arguments gfx A graphics handle as returned by CREATE_GRAPHIC@

Description Returns a gr_background@ format containing information about the background of the current slide.

format gr_background@

format gr_fill_attr_type@

hide ' a boolean indicating whether or not the
 ' background of the slide is hidden. You can hide
 ' the background of a slide using the macro
 ' GR_SET_TEMPLATE_BG@.

format gr_fill_attr_type@

type, ' a string: "built-in" followed by a number.
 ' Indicates the fill used in the background of
 ' the slide
fg_color, ' a string: the name of a color in the colormap
bg_color, ' a string: the name of a color in the colormap
angle, ' Not returned by GR_GET_SLIDE_BG@
offset ' Not returned by GR_GET_SLIDE_BG@

See also [GR SET SLIDE BG@](#)

GR_GET_SLIDE_COLOR_SCHEME@

Returns an array describing the color scheme of the current slide

Format format gr_color_scheme@ info = GR_GET_SLIDE_COLOR_SCHEME@()

format gr_color_scheme@ info = GFX_GET_SLIDE_COLOR_SCHEME@(gfx)

Method format gr_color_scheme@ info = this.get_slide_color_scheme@

Arguments gfx A graphics handle as returned by CREATE_GRAPHIC@

Description Returns a gr_color_scheme@ array containing information about the color scheme of the current slide. The gr_color_scheme@ format contains the following fields:

format gr_color_scheme@

title_fg,	' Foreground color for title text in each slide
title_bg,	' Background color for title text in each slide
text_fg,	' Foreground color for regular text
text_bg,	' Background color for regular text
line_fg,	' Foreground color for lines
line_bg,	' Background color for lines
fill_fg,	' Foreground color for fills
fill_bg,	' Background color for fills
shadow_color,	' Color for shadows
accent_color`	' An array of 15 accent colors used by charts and other objects

Color schemes are established for each presentation through the Utilities ® Presentations ® Color Scheme dialog in Presents.

See also [GR SET SLIDE COLOR SCHEME@](#),
[Utilities --> Presentations --> Color Scheme](#)

GR_GET_SLIDE_LAYOUT@

Returns information about the layout of a slide

Format format gr_layout@ layout = GR_GET_SLIDE_LAYOUT@()
format gr_layout@ layout = GFX_GET_SLIDE_LAYOUT@(gfx)

Method format gr_layout@ layout = this.get_slide_layout@

Arguments gfx A graphics handle as returned by CREATE_GRAPHIC@

Description Returns an array of information about the layout of the slide, and the objects on the slide. The information is an array of format `gr_layout@`. The `gr_layout@` format contains the following fields:

format `gr_layout@`

TYPE,	' A string containing the slide type. These are ' as follows:	
' Title Slide	List and Chart	List and ClipArt
' Bulleted List	Chart and List	ClipArt and List
' Chart	List and Object	Object
' 2 Column List	List and 2 objects	Object and list
' Object over List	2 Objects and List	2 Objects over list
' List over Object	4 Objects	Title Only
' Blank	Custom	
OBJECT1,	' Strings containing the Object types	
OBJECT2,	' These strings can be can be Text, Bulleted List,	
OBJECT3,	' Clipart, or Object. If there are fewer than four	
OBJECT4	' objects on the list, some of the object string may ' contain "none".	

See also [GR SET SLIDE LAYOUT@](#)

GR_GET_SLIDE_SHOW_OPTIONS@

Returns an array describing the Slide Show Options of the current presentation

Format `format gr_slide_show_options@ options = GR_GET_SLIDE_SHOW_OPTIONS@()`
`format gr_slide_show_options@ options =`
`GFX_GET_SLIDE_SHOW_OPTIONS@(gfx)`

Arguments `gfx` A graphics handle as returned by `CREATE_GRAPHIC@`

Description Returns an array with information about the slide show options established for this presentation. The `gr_slide_show_options@` format contains the following fields:

format gr_slide_show_options@

full_screen, ' -1 (full screen) or 0 (not full screen)
all_pages, ' -1 means use all pages in the presentation
' for the slide show. 0 indicates use pages
' between page1 and pageN.
page1, ' First page of the slide show
pageN, ' Last Page of the slide show
at_end ' ' a string = "restart", "exit", or "wait" (default)

See also GFX_GET_SLIDE_SHOW_OPTIONS@, [GR SET SLIDE SHOW OPTIONS@](#)

GR_GET_SLIDE_TRANSITION@

Returns an array of Slide transitions for the current presentation

Format format gr_slide_show@ options = GR_GET_SLIDE_TRANSITION@(page)
format gr_slide_show@ options = GFX_GET_SLIDE_TRANSITION@(gfx, page)

Method format gr_slide_show@ options = this.get_slide_transition@(page)

Arguments gfx A graphics handle as returned by CREATE_GRAPHIC@
page a number indicating the page whose transitions you want returned. This is a zero-based number. Therefore, the first slide in your presentation is slide 0.

Description Returns an array of format gr_slide_show@. The gr_slide_show@ format contains the following fields:

format gr_slide_show@

transition, ' a string containing The type of transition
' for the page, and the direction.
delay, ' The number of seconds to delay before
' advancing to the next slide. -1 indicates
' that you want to advance on mouse click.
enable_builds, ' Reserved for future use
dim_inactive_bullets, ' Reserved for future use
dim_color, ' Reserved for future use
build_transition ' Reserved for future use

See also [GR SET SLIDE TRANSITION@](#)

GR/GFX_GET_STRUCTURED_COMMENT@

Returns the value of an embedded Graphics comment

Format value = GR_GET_STRUCTURED_COMMENT@(name)
value = GFX_GET_STRUCTURED_COMMENT@(gfx, name)

Method value = this.get_structured_comment@(name)

Arguments gfx A graphics handle.
name The name of an existing Graphics comment.

Description Returns the value previously assigned to a Graphics structured comment using the following macro:

[GR SET STRUCTURED COMMENT@](#)

Structured comment variables (and their values) are stored within the document and are retained between Applixware sessions.

GR/GFX_GET_STRUCTURED_COMMENTS@

Returns a list containing the names of all structured comments

Format nameArray = GR_GET_STRUCTURED_COMMENTS@()
nameArray = GFX_GET_STRUCTURED_COMMENTS@(gfx)

Method nameArray = this.get_structured_comments@

Arguments gfx A graphics handle.

Description Returns a list of the names of all structured comment variables stored within a Graphics document.

GR/GFX_GET_TAG@

If a selected object is tagged, returns the tag

Format tagName = GR_GET_TAG@()
tagName = GFX_GET_TAG@(gfx)

Method tagName = this.get_tag@

Arguments gfx A graphics handle.

Description If a selected object is tagged, this macro returns the tagName associated with this object.

If no object is tagged, NULL is returned.

See also [GR SET TAG@](#)

GR_GET_TEMPLATE_BG@

Returns the background
of slides in the current presentation

Format array = GR_GET_TEMPLATE_BG@()

Method this.get_template_bg@

Description Gets the display characteristics of the background of template you used to create the current presentation. This macro returns a gr_background@ format containing the following fields:

format gr_background@

format gr_fill_attr_type@

hide ' a boolean indicating whether or not the
 ' background of the slide is hidden. If hide is -1,
 ' the background is hidden.

format gr_fill_attr_type@

type, ' The name of fill used for the background. This
 ' name is the string "built-in " and the number of
 ' the fill. The fills are available from the Slide

' Background option on the Utilities menu in
 ' Applixware Presents.
 fg_color, ' a string: the name of a color in the colormap
 bg_color, ' a string: the name of a color in the colormap
 angle, ' Not used by GR_SET_SLIDE_BG@
 offset ' Not used by GR_SET_SLIDE_BG@

See also GFX_GET_TEMPLATE_BG@, [GR_GET_SLIDE_BG@](#)

GR/GFX_GET_TOOL_INFO@

Returns the ID number of the currently selected Graphics tool

Format id = GR_GET_TOOL_INFO@()
 id = GFX_GET_TOOL_INFO@(gfx)

Method id = this.get_tool_info@

Arguments gfx A graphics handle.

Description Returns the ID number of the currently selected drawing tool or Pixel Editor tool. The ID numbers are:

Drawing tools:

- 0 Selection tool
- 1 Text
- 2 Line
- 3 Polyline
- 4 Rectangle
- 5 Ellipse
- 6 Free form
- 7 Digitizer
- 8 Square
- 9 Circle
- 10 Curve
- 11 Color digitize
- 12 Zoom
- 13 Textbox

Pixel editor tools:

- 20000 Paint brush
- 20001 Wash (color only)

20002 Spray
20003 Blend (color only)
20004 Erase
20005 Erase foreground (color only)
20006 Line
20007 Rectangle
20008 Ellipse
20009 Fill
20010 Image Filter
20011 Select region

See also [GR_TOOL_PICK@](#)

GR/GFX_GET_UDO_INFO@

Returns name and isResizable flag for a user-defined object

Format infoArray = GR_GET_UDO_INFO@()
infoArray = GFX_GET_UDO_INFO@(gfx)

Method infoArray = this.get_udo_info@

Arguments gfx A graphics handle.

Description Returns a two element array whose contents indicate some properties of the current User-Defined Object (UDO), as follows:

infoArray[0] The name of the object.

infoArray[1] A Boolean value which if set to TRUE indicates that the size of the object cannot be changed.

GR/GFX_GOTO_BEGIN_LINE@

Moves cursor to the beginning of the line

Format GR_GOTO_BEGIN_LINE@()
GFX_GOTO_BEGIN_LINE@(gfx)

Method this.goto_begin_line@

Arguments gfx A graphics handle.

Description Moves the cursor to the beginning of the current line. The cursor is placed before the first character in the line. GR_GOTO_BEGIN_LINE@ is relevant only if a line of text is in editing mode in the Graphics document.

GR_GOTO_BEGIN_LINE@ is called by the Keys ® Beginning of line menu option in Graphics.

GR/GFX_GOTO_END_LINE@

Moves cursor to the end of the line

Format GR_GOTO_END_LINE@()
GFX_GOTO_END_LINE@(gfx)

Method [this.goto_end_line@](#)

Arguments gfx A graphics handle.

Description Moves the cursor to the end of the current line. The cursor is placed after the last character in the line. GR_GOTO_END_LINE@ is relevant only if a line of text is in editing mode in the Graphics document.

GR_GOTO_END_LINE@ is called by the Keys ® End of line menu option.

GR/GFX_GRIDS@

Sets the grid line display

Format GR_GRIDS@()
GFX_GRIDS@(gfx)

Method [this.grids@](#)

Arguments gfx A graphics handle.

Description Toggles the display of grid lines in the Graphics document. If grid lines are not currently displayed, GR_GRIDS@ displays grid lines. If grid lines are currently displayed, GR_GRIDS@ turns off the display of grid lines.

GR_GRIDS@ is called by the View ® Grid Lines menu option.

See also [GR_AUTO_GRID@](#)
[GR_GRID ?@](#)

GR_GRID_?@

Sets the grid scale

Format GR_GRID_?@()

Method [this.grid_?@](#)

Description Sets the grid scale for the current Graphics document. One of the following macros exists for each setting:

GR_GRID_ONE_HALF_INCH@
Sets the grid scale to ½ inch

GR_GRID_ONE_QUARTER_INCH@
Sets the grid scale to ¼ inch

GR_GRID_ONE_EIGHTH_INCH@
Sets the grid scale to 1/8 inch

GR_GRID_ONE_TENTH_INCH@
Sets the grid scale to 1/10 inch

GR_GRID_ONE_SIXTEENTH_INCH@
Sets the grid scale to 1/16 inch

GR_GRID_CENTIMETER@
Sets the grid scale to 1 centimeter.

All GR_GRID_?@ macros are called by the View ® Grid/Ruler Scale menu option.

See also [GR_AUTO_GRID@](#)
[GR_GRIDS@](#)

GR/GFX_GROUP@

Groups selected objects

Format GR_GROUP@()

GFX_GROUP@(gfx)

Method [this.group@](#)

Arguments gfx A graphics handle.

Description All objects selected at the time GR_GROUP@ is executed become part of a single group. GR_GROUP@ is called by the Arrange ® Group menu option.

See also [GR_UNGROUP@](#)

GR/GFX_GUIDES@

Sets guide attributes for a Graphics document

Format GR_GUIDES@ (notUsed, reserved, rulersFlag, gridsFlag, coordinatesFlag, gridSnapFlag, stickyPointsFlag, pageBreaksFlag)

Method this.guides@ (notUsed, reserved, rulersFlag, gridsFlag, coordinatesFlag, gridSnapFlag, stickyPointsFlag, pageBreaksFlag)

GFX_GUIDES@ (gfx, notUsed, reserved, rulersFlag, gridsFlag, coordinatesFlag, gridSnapFlag, stickyPointsFlag, pageBreaksFlag)

Arguments

gfx	A graphics handle.
printer	The name of the printer, a string, on which the document will be printed. Pagination of a document is based on the printer specification.
reserved	Not used. Set to 0.
rulersFlag	Enables and disables the display of rulers. If TRUE, horizontal and vertical rulers are displayed on the screen. If FALSE, rulers are not displayed.
gridsFlag	Enables and disables the display of grid lines. If TRUE, horizontal and vertical grid lines are displayed on the screen. If FALSE, no grid lines are displayed.
coordinatesFlag	Enables and disables the display of coordinates when an object is drawn or moved. If TRUE, location and size coordinates are displayed when an object is drawn or moved. If FALSE, no coordinates are displayed.
gridSnapFlag	Enables and disables grid snap. If TRUE, then grid snap is enabled. If FALSE, then grid snap is disabled.
stickyPointsFlag	Enables and disables sticky points. When enabled, an object's lines and handles attract other objects when the object is placed near them. If TRUE, sticky points are enabled. If FALSE, sticky points are disabled.
pageBreaksFlag	Enables and disables the display of page breaks. If TRUE, page breaks are displayed. If FALSE, page breaks are not displayed.

Description If an argument is set to NULL, then no change to that guide attribute is made.

See also [GR_GET_GUIDES@](#)

GR/GFX_HIDE_LAYER@

Hides objects on layer

Format GR_HIDE_LAYER@(layerName)
GFX_HIDE_LAYER@(gfx, layerName)

Method [this.hide_layer@](#)(layerName)

Arguments

gfx	A graphics handle.
layerName	A name of a layer. If the layer does not exist, it will be created. In the layer needs to be created, however, no objects will be hidden (as no objects will be on the layer).

Description Hides all objects that are on a layer. Conceptually, this makes all of the layer's objects invisible.

See also [GR_REVEAL_LAYER@](#)

GR/GFX_HIDE_SELECTED@

Hides selected objects from view

Format GR_HIDE_SELECTED@()
GFX_HIDE_SELECTED@(gfx)

Method [this.hide_selected@](#)

Arguments

gfx	A graphics handle.
-----	--------------------

Description Hidden objects cannot be edited and do not appear in insets or in the printed document. GR_HIDE_SELECTED@ is called by the View ® Hide Selected menu option.

See also [GR_HIDE_UNSELECTED@](#)
[GR_REVEAL@](#)

GR/GFX_HIDE_UNSELECTED@

Hides all unselected objects from view

Format GR_HIDE_UNSELECTED@()
GFX_HIDE_UNSELECTED@(gfx)

Method [this.hide_unselected@](#)

Arguments gfx A graphics handle.

Description Hides any objects that are not selected in the Graphics document. Hidden objects cannot be edited and do not appear in insets or in the printed document. GR_HIDE_UNSELECTED@ is called by the View ® Hide Selected menu option.

See also [GR_HIDE_SELECTED@](#)
[GR_REVEAL@](#)

GR_HORIZ_GUIDE@

Limits movement to horizontal direction

Format GR_HORIZ_GUIDE@()

Method [this.horiz_guide@](#)

Description Any objects moved immediately after GR_HORIZ_GUIDE@ is executed can only be moved horizontally. GR_HORIZ_GUIDE@ is only effective for a single movement. If you perform an action other than moving an object after executing GR_HORIZ_GUIDE@, the horizontal movement limitation is ended.

See also [GR_VERTICAL_GUIDE@](#)

GR_IMPORT_?@

Imports a file into the current Graphics window

Format GR_IMPORT_?@(filename)

Method [this.import_?@\(filename\)](#)

Arguments filename The name of the file being imported.

Description Converts a file into Applixware format and inserts it into the current Graphics window. Each file format that can be converted has a macro that does the conversion. The bold macros cannot be used without first obtaining a Graphics Pack license from Applixware. These macros are as follows:

GR_IMPORT_CGM@	CGM (Common Graphics Metafile) file
GR_IMPORT_DXF@	DXF file
GR_IMPORT_EPS@	Encapsulated PostScript
GR_IMPORT_FAX@	CCITT Group 3 FAX file
GR_IMPORT_GEM@	GEM file
GR_IMPORT_GIF@	GIF file
GR_IMPORT_GP4@	CCITT Group 4 FAX file
GR_IMPORT_HPGL@	HPGL file
GR_IMPORT_ILBM@	ILBM file
GR_IMPORT_IM@	Applixware Bitmap file
GR_IMPORT_IRIS@	Silicon Graphics IRIS file
GR_IMPORT_JPEG@	JPEG Interchange Format
GR_IMPORT_MACPAINT@	MacPaint file
GR_IMPORT_MSWINBM@	Microsoft Window Bitmap file
GR_IMPORT_PBM@	X11 Portable Bitmap file
GR_IMPORT_PCX@	PCX file
GR_IMPORT_PGM@	X11 Portable Graymap file
GR_IMPORT_PICT2@	MacDraw PICT2 file
GR_IMPORT_PICT@	MacDraw PICT file
GR_IMPORT_PPM@	X11 Portable Pixmap file
GR_IMPORT_PPT@	Powerpoint file
GR_IMPORT_RAW@	Raw file
GR_IMPORT_SUN@	Sun Raster file
GR_IMPORT_TGA@	TGA File
GR_IMPORT_TIF@	TIFF file
GR_IMPORT_WMF@	Windows Metafile
GR_IMPORT_WPG@	WordPerfect Graphics file
GR_IMPORT_XBM@	X Windows Bitmap file
GR_IMPORT_XPM@	X Windows Pixmap file
GR_IMPORT_XWD@	X Windows Dump file

See also [**GE PASTE ?@**](#)

GR_IMPORT_FILE@

Imports a file into the current graphics window

Format GR_IMPORT_FILE@(filename, type)

Method [this.import_file@](#)(filename, type)

Arguments

filename	The name of the file being imported.
type	One of the following values: 0 Applixware bitmap file 1 CGM file 2 CGM CALS file 3 DXF 4 Encapsulated PostScript 5 CCITT Group 3 FAX 6 CCITT Group 4 FAX 7 GEM 8 GIF 9 HPGL 10 IGES 11 ILBM 12 Interleaf LEAF 13 MacPaint 14 Microsoft Windows Bitmap 15 X11 Portable Bitmap file 16 X11 Portable Graymap file 17 X11 Portable Pixmap file 18 PCX file 19 MacDraw PICT file 20 MacDraw PICT2 file 21 Raw file 22 SGI IRIS file 23 Sun Raster 24 TGA 25 TIFF 26 Windows Metafile 27 WordPerfect Graphics 28 X Windows Bitmap file 29 X windows Pixmap file 30 X windows Dump file

Description Converts a file into Applixware format and inserts it into the current Graphics window.

GR_INSERT_PAGE@

Adds a page to the current graphics document

Format GR_INSERT_PAGE@()

Description Adds a page to the current graphics document. The page is added immediately after the current page of the document. If you are using Applixware Presents, an empty slide is added to your presentation immediately after the current slide.

See also GFX_INSERT_PAGE@

GR/GFX_ITALICS@

Makes selected text italic

Format GR_ITALICS@()
GFX_ITALICS@(gfx)

Method [this.italics@](#)

Arguments gfx A graphics handle.

Description If no text is selected, italics becomes a default attribute for the document. If the selection is already italicized, GR_ITALICS@ removes the italic attribute from the selection. GR_ITALICS@ is called by the Text ® Italic menu option.

See also [GR_BOLD@](#)
[GR_UNDERLINE@](#)

GR/GFX_LARGER@

Increases the size of selected text, objects, or pixels

Format GR_LARGER@()
GFX_LARGER@(gfx)

Method [this.larger@](#)

Arguments gfx A graphics handle.

Description If the current Graphics window is not in pixel editing mode, GR_LARGER@ increases the size of selected text and objects as follows:

- Selected text is increased to the next larger point size.
- Selected objects are increased in size by 10 percent in both the horizontal and vertical direction.

If the current Graphics window is in pixel editing mode, GR_LARGER@ increases the size of pixels in the pixel editing area. GR_LARGER@ is called by the Transform ® Scale up menu option.

See also [GR_SMALLER@](#)

GR/GFX_LAST_PAGE@

Moves to and displays last page

Format GR_LAST_PAGE@()
GFX_LAST_PAGE@(gfx)

Method [this.last_page@](#)

Arguments gfx A graphics handle.

Description Makes the last graphics page current and displays it within the Graphics window.

GR/GFX_LAST_SCREEN_KEY@

Displays the bottom right area of the current page

Format GR_LAST_SCREEN_KEY@()
GFX_LAST_SCREEN_KEY@(gfx)

Method [this.last_screen_key@](#)

Arguments gfx A graphics handle.

GR/GFX_LEFT_ARROW_KEY@

Moves object or cursor to the left

Format GR_LEFT_ARROW_KEY@()
GFX_LEFT_ARROW_KEY@(gfx)

Method [this.left_arrow_key@](#)

Arguments gfx A graphics handle.

Description If objects are selected, moves the selected objects left. If the cursor is within text, the cursor is moved to the left one character.

See also [GR_RIGHT_ARROW_KEY@](#)

GR/GFX_LEFT_SCREEN_KEY@

Move to the left one screen

Format GR_LEFT_SCREEN_KEY@()
GFX_LEFT_SCREEN_KEY@(gfx)

Method [this.left_screen_key@](#)

Arguments gfx A graphics handle.

Description Displays the Graphics objects that are to the left of the currently displayed objects.

GR/GFX_LIST_FONT_FAMILIES@

Returns a list of available fonts

Format fontArray = GR_LIST_FONT_FAMILIES@()
fontArray = GFX_LIST_FONT_FAMILIES@(gfx)

Method fontArray = [this.list_font_families@](#)

Arguments gfx A graphics handle.

Description Returns a string array whose elements are the names of all Applixware fonts associated with a Graphics document.

GR/GFX_LOAD_COLORMAP_FILE@

Loads a colormap from an external file

Format GR_LOAD_COLORMAP_FILE@(pathname)
GFX_LOAD_COLORMAP_FILE@(gfx, pathname)

Method [this.load_colormap_file@](#)(pathname)

Arguments gfx A graphics handle.
pathname The name of the file containing the color map.

Description Replaces the current colormap with the colormap contained within pathname. Note that the placement of entries in a colormap determines use. That is, references are not by a name but by the position within the colormap.

For example, assume "red" was in position 2 in the current map and "green" was in position 2 in pathname's colormap. In this case, loading the new colormap would change the color in which "red" items were displayed.

GR/GFX_LOAD_CUSTOM_FILLS@

Loads the fill patterns contained within a file

Format GR_LOAD_CUSTOM_FILLS@(pathname)
GFX_LOAD_CUSTOM_FILLS@(gfx, pathname)

Method [this.load_custom_fills@](#)(pathname)

Arguments gfx A graphics handle.
pathname The name of the file containing the fill patterns.

Description Adds the fill patterns contained in path name to the list of currently available fill patterns.

See also [Custom Fill Patterns](#)

GR/GFX_LOAD_FILE@

Loads a file into the current Graphics window

Format GR_LOAD_FILE@(filename)
GFX_LOAD_FILE@(gfx, filename)

Method this.load_file@(filename)

Arguments gfx A graphics handle.
filename The name of the Graphics file you want to load. If the file is not in the current directory, filename must be a full pathname.

Description Loads the specified file into the current Graphics document. If you have not saved changes to the current document, the changes are lost.
If the document specified by filename does not exist, a blank drawing area is displayed in the current window and the window is given the name specified by filename.

GR/GFX_LOAD_PARTS_FILE@

Loads a custom part file

Format GR_LOAD_PARTS_FILE@(filename)
GFX_LOAD_PARTS_FILE@(gfx, filename)

Method this.load_parts_file@(filename)

Arguments gfx A graphics handle.
filename The name of the Graphics file containing the parts you want to load. If the file is not in the current directory, filename must be a full path name.

Description Adds the custom parts from the file you specify to the custom parts area in the current Graphics document. This allows you to use all the defined parts from the other file in your current document.

GR/GFX_LOCK_LAYER@

Locks a layer

Format GR_LOCK_LAYER@(layerName)
GFX_LOCK_LAYER@(gfx, layerName)

Method [this.lock_layer@](#)(layerName)

Arguments gfx A graphics handle.
layerName The name of the layer to lock.

Description Locks the passed layerName, making objects in layerName unselectable and uneditable.

See also [GR_UNLOCK_LAYER@](#)

GR/GFX_LOCK_SELECTED@

Locks selected objects

Format GR_LOCK_SELECTED@(
GFX_LOCK_SELECTED@(gfx)

Method [this.lock_selected@](#)

Arguments gfx A graphics handle.

Description Locks the selected objects, making the objects unselectable and uneditable.

See also [GR_UNLOCK_ALL@](#)
[GR_UNLOCK_SELECTED@](#)

GR/GFX_LOCK_TOOL@

Selects and locks a drawing tool

Format GR_LOCK_TOOL@(tool)

Method [this.lock_tool@](#)(tool)

GFX_LOCK_TOOL@(gfx, tool)

Arguments gfx A graphics handle.
 tool The ID number of the tool you want to select. tool can be one of the following:

Drawing Tools:

selection_tool
text_tool
line_tool
polyline_tool
rectangle_tool
ellipse_tool
freeform_tool
mono_digitizer_tool
rounded_rectangle_tool
polygon_tool
curve_tool
color_digitizer_tool
zoom_tool
formatted_text_tool

Pixel Editor Tools:

paint_tool
wash_tool
pen_tool
blend_tool
erase_tool
color_erase_tool
line_tool
rectangle_tool
ellipse_tool
flood_tool
filter_tool
selection_tool

Description The tool selection will remain in effect until either a different tool is selected or the session is ended.

See also [GR_TOOL_PICK@](#)

GR/GFX_MAIL@

Mails the current Graphics document

Format GR_MAIL@()
GFX_MAIL@(gfx)

Method [this.mail@](#)

Arguments gfx A graphics handle.

Description Invokes a dialog box that, when executed, mails the current Graphics document.

GR/GFX_MAP_ICON_CURSOR@

Binds a cursor to a window type (by icon)

Format GR_MAP_ICON_CURSOR@(cursorName)
GFX_MAP_ICON_CURSOR@(gfx, cursorName)

Method [this.map_icon_cursor@\(cursorName\)](#)

Arguments gfx A graphics handle.
cursorName The name of a cursor.

Description Names the cursor that will be used in the Graphics editor.

GR_MERGE_ATTRS@

Creates attribute containing newAttr elements that differ from oldAttr

Format format gr_attribute@ newAttr = GR_MERGE_ATTRS@(format gr_attribute@ firstAttr,
format gr_attribute@ secondAttr)

Method [format gr_attribute@ newAttr = this.merge_attrs@\(format gr_attribute@ firstAttr, format
gr_attribute@ secondAttr\)](#)

Arguments firstAttr The first of the attribute structures being merged.
secondAttr The second of the attribute structures being merged.

Description Creates a new attribute structure that contains only the attributes that are different in two structures. For example, if firstAttr.backfill is equal to secondAttr.backfill, the newAttr.backfill element will equal NULL. However, if the firstAttr.backfill element is not equal to secondAttr.backfill, then newAttr.backfill will be set equal to secondAttr.backfill.

GR/GFX_MERGE_FILE@

Merges a Graphics file with the current Graphics image

Format GR_MERGE_FILE@(infoArray)
GFX_MERGE_FILE@(gfx, infoArray)

Method [this.merge_file@](#)(infoArray)

Arguments gfx A graphics handle.
infoArray An array containing the following information:
 infoArray[0] File name
 infoArray[1] X offset
 infoArray[2] Y offset

Description Writes the contents of infoArray[0] into the current Graphics at the named position. The contents of this file are placed as another part within the current Graphics. It does not overwrite or otherwise change existing information.

GR/GFX_MERGE_LISTS@

Merges two unsorted vectors or elements together

Format list3 = GR_MERGE_LISTS@(list1, list2)

Method [list3 = this.merge_lists@](#)(list1, list2)

Arguments list1 The first list being merged.
list2 The second list being merged.

Description Returns a list that created by merging two other lists together. If an element is in both list, it is only included once in the returned list.
Although this command is designed to merge two unsorted lists, it will function correctly if one or both of the parameters are simple elements. Unlike most merge operations, this merge operation can merge unsorted lists.

While this macro can merge any two lists, it is used in Graphics to merge together lists created by the [GR_QUERY@](#) command.

GR_MIX_ATTRS@

Returns attributes existing in both attribute arguments

Format `format gr_attribute@ mixedAttr = GR_MIX_ATTRS@(format gr_attribute@ oldAttribute, format gr_attribute@ newAttribute)`

Method `format gr_attribute@ mixedAttr = this.mix_attrs@(format gr_attribute@ oldAttribute, format gr_attribute@ newAttribute)`

Arguments `oldAttribute` The structure that will contribute information to the returned structure.
`newAttribute` That structure that is compared against `oldAttribute` to determine if the attribute exists in both `oldAttribute` and `newAttribute`.

Description Compares each attribute in `oldAttribute` with the corresponding attribute in `newAttribute`. If the attributes are the same, the attribute is copied to `mixedAttr`. If they differ, the corresponding attribute within `mixedAttr` is set to NULL.

See also [GR_CHART_GET_ATTR@](#) for a definition of the `gr_attribute@` format.

GR/GFX_MODIFIED@

Indicates if a Graphics document is modified

Format `modifiedFlag = GR_MODIFIED@()`
`modifiedFlag = GFX_MODIFIED@(gfx)`

Method `modifiedFlag = this.modified@`

Arguments `gfx` A graphics handle.

Description Returns TRUE if the current Graphics document has been modified since it was last saved. It returns FALSE if no modifications have been made.

If the document is a new document that has never been saved to a file, TRUE is returned if any editing occurred in the Graphics window.

GR/GFX_MOUSE_DOUBLE_DOWN@

Performs a mouse down event immediately after down/up events

Format GR_MOUSE_DOUBLE_DOWN@(x, y, format gr_mouse@ mouse)
GFX_MOUSE_DOUBLE_DOWN@(gfx, x, y, format gr_mouse@ mouse)

Method [this.mouse_double_down@](#)(x, y, format gr_mouse@ mouse)

Arguments

gfx	A graphics handle.
x	The mouse's X coordinate.
y	The mouse's Y coordinate.
mouse	A format that contains button status information. For a description of this format, see GR_MOUSE_DOWN@ .

Description Sends an event that is the equivalent of the user pressing the mouse button at the indicated position immediately after a mouse down and a mouse up event occurred.

See also [GR_MOUSE_DOWN@](#)
[GR_MOUSE_MOVE@](#)
[GR_MOUSE_UP@](#)

GR/GFX_MOUSE_DOWN@

Performs a mouse down event

Format GR_MOUSE_DOWN@(x, y, format gr_mouse@ mouse)
GFX_MOUSE_DOWN@(gfx, x, y, format gr_mouse@ mouse)

Method [this.mouse_down@](#)(x, y, format gr_mouse@ mouse)

Arguments

gfx	A graphics handle.
x	The mouse's X coordinate.
y	The mouse's Y coordinate.
mouse	A format (described below) that contains button status information.

Description Sends an event that is the equivalent of the user pressing the mouse button at the indicated position.

The definition of `gr_mouse@` is as follows:

```
format gr_mouse@
  window_relative, 'Default is TRUE
  button1,         'Default is TRUE
  button2,         'Default is FALSE
  button3,         'Default is FALSE
  button4,         'Default is FALSE
  button5,         'Default is FALSE
  shifted,         'Default is FALSE
  controled,       'Default is FALSE
  alted            'Default is FALSE
```

In most cases, you can pass a mouse value of `NULL`. This indicates that mouse events occur relative to the current window's zoom setting and that the first button is pressed or released. *Relative* to the window's current zoom setting means that a movement is never adjusted. That is, a change in the *X* position of 1 unit is always 1 unit relative to the current window's zoom. For example, assume that you create a macro that draws a rectangle. If the window's zoom level changes, the size of the rectangle that is created will differ.

If `window_relative` is set to `FALSE`, the zoom level will be ignored. That is, if you are drawing a rectangle with a macro, the rectangle's size will not change if the zoom level changes.

See also [GR_MOUSE_DOUBLE_DOWN@](#)
[GR_MOUSE_MOVE@](#)
[GR_MOUSE_UP@](#)

GR/GFX_MOUSE_MOVE@

Performs a mouse move event with a button pressed

Format `GR_MOUSE_MOVE@(x, y, format gr_mouse@ mouse)`
`GFX_MOUSE_MOVE@(gfx, x, y, format gr_mouse@ mouse)`

Method `this.mouse_move@(x, y, format gr_mouse@ mouse)`

Arguments

<code>gfx</code>	A graphics handle.
<code>x</code>	The mouse's <i>X</i> coordinate.
<code>y</code>	The mouse's <i>Y</i> coordinate.
<code>mouse</code>	A format that contains button status information. For a description of this format, see <u>GR_MOUSE_DOWN@</u> .

Description Sends an event that is the equivalent of the user moving the mouse with a button pressed. It is normal for more than one *move* event to be recorded while the user moves the mouse. However, if you are creating a macro based on a recording, you can remove intermediate *move* events.

See also [GR_MOUSE_DOUBLE_DOWN@](#)
[GR_MOUSE_DOWN@](#)
[GR_MOUSE_UP@](#)

GR/GFX_MOUSE_UP@

Performs a mouse up event

Format GR_MOUSE_UP@(x, y, format gr_mouse@ mouse)
GFX_MOUSE_UP@(gfx, x, y, format gr_mouse@ mouse)

Method [this.mouse_up@](#)(x, y, format gr_mouse@ mouse)

Arguments

gfx	A graphics handle.
x	The mouse's X coordinate.
y	The mouse's Y coordinate.
mouse	A format that contains button status information. For a description of this format, see GR_MOUSE_DOWN@ .

Description Sends an event that is the equivalent of the user releasing the mouse button at the indicated position.

See also [GR_MOUSE_DOUBLE_DOWN@](#)
[GR_MOUSE_DOWN@](#)
[GR_MOUSE_MOVE@](#)

GR/GFX_MOVE_BACK@

Moves the selected object in back of all other objects

Format GR_MOVE_BACK@()
GFX_MOVE_BACK@(gfx)

Method [this.move_back@](#)

Arguments gfx A graphics handle.

Description Moves the selected object to the back in the current Graphics document. The object moved to the back by GR_MOVE_BACK@ is treated as the first object created. All other objects in the document will overlap the back object when placed on top of it.

If multiple objects are selected when GR_MOVE_BACK@ is executed, the objects are moved to the back in the reverse order of their current order ranking. Thus, the backmost object of all the selected objects will become the backmost object in the drawing area.

GR_MOVE_BACK@ is called by the Arrange ® Move To Back menu option.

See also [GR_MOVE_FRONT@](#)

GR/GFX_MOVE_BACKWARD@

Moves an object back

Format GR_MOVE_BACKWARD@()
GFX_MOVE_BACKWARD@(gfx)

Method [this.move_backward@](#)

Arguments gfx A graphics handle.

Description Moves the selected object(s) behind the object created immediately before the currently selected object. The front and back orientation of objects is determined by the order in which you create them. The object you created first is in the back and the object you created most recently is in the front.

For example, if there are four overlapping objects, executing this macro on the first object moves it backward so that it is between what was previously the second and third objects. That is, it is now the second object. The second object will become the topmost (or most forward) object.

To place the selected object in front of the object created immediately after the currently selected object, see [GR_MOVE_FORWARD@](#).

To activate this option you must first select at least one object.

GR/GFX_MOVE_EXACT@

Moves selected objects the number of inches specified

Format GR_MOVE_EXACT@(horizontal, vertical)
GFX_MOVE_EXACT@(gfx, horizontal, vertical)

Method [this.move_exact@](#)(horizontal, vertical)

Arguments

gfx	A graphics handle.
horizontal	The number of units to move selected objects horizontally. If horizontal is a positive number, the objects are moved to the right the specified number of units. If horizontal is a negative number, the objects are moved to the left the specified number of units.
vertical	The number of units to move selected objects vertically. If vertical is a positive number, the objects are moved down the specified number of units. If vertical is a negative number, the objects are moved up the specified number of units.

Description When more than one object is selected, they are all moved by the specified amount. Setting the horizontal or vertical argument to 0 indicates that the objects will not be moved in that direction. If the movement amount would move the object beyond the borders of the screen, an error is thrown.

GR/GFX_MOVE_FORWARD@

Moves an object in front of another object

Format GR_MOVE_FORWARD@()
GFX_MOVE_FORWARD@(gfx)

Method [this.move_forward@](#)

Arguments gfx A graphics handle.

Description Moves the selected object(s) in front of the object created immediately after the currently selected object. The front and back orientation of objects is determined by the order in which you create them. The object you created first is in the back and the object you created most recently is in the front.

For example, if there are four overlapping objects, executing this macro on the third object moves it forward so that it is between what was previously the second and first objects. That is, it is now the second object. The second object will become the third object.

To place the selected object in back of the object created immediately before the currently selected object, see [GR_MOVE_BACKWARD@](#).

To activate this option you must first select at least one object.

GR/GFX_MOVE_FRONT@

Moves the selected object in front of all other objects

Format GR_MOVE_FRONT@()
GFX_MOVE_FRONT@(gfx)

Method [this.move_front@](#)

Arguments gfx A graphics handle.

Description Moves the selected object to the front in the current Graphics document. The object moved to the front by GR_MOVE_FRONT@ is treated as the last object created. This object will overlap all other objects in the document when placed over them.

If multiple objects are selected when GR_MOVE_FRONT@ is executed, the objects are moved to the front in the reverse order of their current order ranking. Thus, the front-most object of all the selected objects will become the front-most object in the drawing area.

GR_MOVE_FRONT@ is called by the Arrange ® Move To Front menu option.

See also [GR_MOVE_BACK@](#)

GR/GFX_NEXT_PAGE@

Moves to the next graphics page

Format GR_NEXT_PAGE@()
GFX_NEXT_PAGE@(gfx)

Method [this.next_page@](#)

Arguments gfx A graphics handle.

GR/GFX_NEXT_PAGE_KEY@

Scrolls to the next page

Format GR_NEXT_PAGE_KEY@()
GFX_NEXT_PAGE_KEY@(gfx)

Method [this.next_page_key@](#)

Arguments gfx A graphics handle.

Description Scrolls the drawing area of the Graphics window to the next page of the document.

See also [GR_PREV_PAGE_KEY@](#)

GR/GFX_NEXT_SCREEN_KEY@

Scrolls the screen down

Format GR_NEXT_SCREEN_KEY@()
GFX_NEXT_SCREEN_KEY@(gfx)

Method [this.next_screen_key@](#)

Arguments gfx A graphics handle.

Description Scrolls the drawing area of the Graphics window so that the area underneath the current screen is displayed.

See also [GR_PREV_SCREEN_KEY@](#)

GR_OLAP_ATTRS@

Returns parts of newAttributes that differ from old & identical parts

Format format gr_attribute@ olapAttr = GR_OLAP_ATTRS@(format gr_attribute@ oldAttribute,
format gr_attribute@ newAttribute)

Method format gr_attribute@ olapAttr = this.olap_attrs@(format gr_attribute@ oldAttribute,
format gr_attribute@ newAttribute)

Arguments oldAttribute The basis for the returned attribute.
newAttribute An attribute containing changes that are to be added to the returned attribute.

Description Compares each attribute in oldAttribute with the corresponding attribute in newAttribute. If the attributes differ, the attribute is copied to olapAttr. If they are the same, the corresponding attribute from oldAttribute is copied to olapAttr.

See also [GR_CHART_GET_ATTR@](#) for a definition of the gr_attribute@ format.

GR_OUTLINE_MOVE_TEXT_DOWN@

Move a line of text down
in a Presents outline

Format GR_OUTLINE_MOVE_TEXT_DOWN@()
GFX_OUTLINE_MOVE_TEXT_DOWN@(gfx)

Arguments gfx agraphics handle as returned by CREATE_GRAPHIC@

Description Moves the current line of text down one line in a Graphics Presents outline. The current line is placed below the line that follows it in the outline. No formatting changes are made to the current line.

See also [GR_OUTLINE_MOVE_TEXT_UP@](#)

GR_OUTLINE_MOVE_TEXT_UP@

Move a line of text up
in a Presents outline

Format GR_OUTLINE_MOVE_TEXT_UP@()
GFX_OUTLINE_MOVE_TEXT_UP@(gfx)

Arguments gfx agraphics handle as returned by CREATE_GRAPHIC@

Description Moves the current line of text up one line in a Graphics Presents outline. The current line is placed above the line that precedes it in the outline. No formatting changes are made to the current line.

See also [GR_OUTLINE_MOVE_TEXT_DOWN@](#)

GR/GFX_PAGEBREAKS@

Sets the page break display

Format GR_PAGEBREAKS@()
GFX_PAGEBREAKS@(gfx)

Method [this.pagebreaks@](#)

Arguments gfx A graphics handle.

Description Toggles the display of page breaks in the Graphics document. If page breaks are not displayed, GR_PAGEBREAKS@ displays page breaks. If page breaks are displayed, GR_PAGEBREAKS@ turns off the display of page breaks.

GR_PAGEBREAKS@ is called by the View ® Page Breaks menu option.

GR/GFX_PASTE@

Pastes the clipboard's contents Graphics window

Format GR_PASTE@()
GFX_PASTE@(gfx)

Method [this.paste@](#)

Arguments gfx A graphics handle.

Description Pastes the contents of the clipboard at the center of the current Graphics document. GR_PASTE@ is called by the Edit ® Paste menu option.

GR_PASTE_EPS@

Pastes an EPS file

Format GR_PASTE_EPS@(pathname)

Method `this.paste_eps@(pathname)`

Arguments filename The path name of the Encapsulated PostScript file.

Description Converts then pastes an Encapsulated PostScript into the center of the current Graphics window.

For more information, see [GE PASTE ?@](#).

GR_PASTE_FAX@

Pastes a Group 3 FAX file

Format `GR_PASTE_FAX@(pathname)`

Method `this.paste_fax@(pathname)`

Arguments pathname The path name of the FAX file.

Description Converts then pastes a Group 3 Fax file into the center of the current Graphics window.

For more information, see [GE PASTE ?@](#).

GR/GFX_PASTE_GFX@

Pastes graphics gfx into the middle of the current graphic

Format `GR_PASTE_GFX@(sourcegfx)`

`GFX_PASTE_GFX@(targetgfx, sourcegfx)`

Method `this.paste_gfx@(sourcegfx)`

Arguments sourcegfx A graphics handle; this graphic will be pasted into another graphic.

targetgfx If using the `GFX_PASTE_GFX@` macro, this is the handle of the graphic into which sourcegfx will be inserted.

Description Pastes the graphics pointed to by sourcegfx into the middle of the current graphic. This graphic can be a graphic window or it can be another graphic handle (targetgfx).

GR_PASTE_GR@

Pastes an Applixware Graphics file

Format GR_PASTE_GR@(pathname)

Method [this.paste_gr@\(pathname\)](#)

Arguments pathname The path name of the Applixware Graphic file.

Description Pastes an Applixware Graphics file into the center of the current Graphics window.

For more information, see [GE PASTE ?@](#).

GR_PASTE_XWD@

Pastes X Windows Dump file

Format GR_PASTE_XWD@(pathname)

Method [this.paste_xwd@\(pathname\)](#)

Arguments pathname The path name of the X Windows Dump file.

Description Converts then pastes an X Windows Dump file into the center of the current Graphics window.

For more information, see [GE PASTE ?@](#).

GR/GFX_PICK_UP_ATTR@

Copies the current attributes

Format GR_PICK_UP_ATTR@()

GFX_PICK_UP_ATTR@(gfx)

Method [this.pick_up_attr@](#)

Arguments gfx A graphics handle.

Description Stores most graphic attributes in temporary memory. These attributes are taken from the selected objects (if a selection is present) or from the attributes currently used.

Use `GR_PICK_UP_ATTR@` to apply the following attributes from an object so they can be applied to another object:

- Line fill patterns and color
- Line width
- Type
- Joins
- Arrowheads and arrow tails
- The object fill patterns and colors
- Shadow type, offset and color

For more information, see [GR_APPLY_ATTR@](#).

GR/GFX_PIXEL_EDIT_MODE@

Indicates whether a Graphics window is in pixel edit mode

Format `flag = GR_PIXEL_EDIT_MODE@()`
`flag = GFX_PIXEL_EDIT_MODE@(gfx)`

Method `flag = this.pixel_edit_mode@`

Arguments `gfx` A graphics handle.

Description Returns TRUE if the current Graphics window is in pixel edit mode. It returns FALSE if the window is not in Pixel Edit mode.

GR_PRESENTATION_MODE@

Returns TRUE if the current document is Applixware Presents

Format `GR_PRESENTATION_MODE@()`

Description Returns TRUE (1) if the current application is Applixware Presents. Returns FALSE (0) if the current application is not Applixware Presents.

See also `GFX_PRESENTATION_MODE@`.

GR/GFX_PREV_PAGE_KEY@

Scrolls to the previous page

Format GR_PREV_PAGE_KEY@()
GFX_PREV_PAGE_KEY@(gfx)

Method [this.prev_page_key@](#)

Arguments gfx A graphics handle.

See also [GR_NEXT_PAGE_KEY@](#)

GR/GFX_PREV_SCREEN_KEY@

Scrolls to the previous screen

Format GR_PREV_SCREEN_KEY@()
GFX_PREV_SCREEN_KEY@(gfx)

Method [this.prev_screen_key@](#)

Arguments gfx A graphics handle.

See also [GR_NEXT_SCREEN_KEY@](#)

GR/GFX_PREVIOUS_PAGE@

Displays the previous page

Format GR_PREVIOUS_PAGE@(
GFX_PREVIOUS_PAGE@(gfx)

Method [this.previous_page@](#)

Arguments gfx A graphics handle.

Description Moves to a previous page. A previous page is a page whose page number is lower than the current page.

GR/GFX_PRINT@

Prints the current Graphics document.

Format GR_PRINT@(printer, colorFlag, copies, bannerFlag, allPagesFlag, startPage, endPage, tempFile[, backgroundFlag][, class] [, baggage], page_type)
GFX_PRINT@(gfx, printer, colorFlag, copies, bannerFlag, allPagesFlag, startPage, endPage, tempFile[, backgroundFlag][, class] [, baggage], page_type)

Method [this.print@](#)(printer, colorFlag, copies, bannerFlag, allPagesFlag, startPage, endPage, tempFile[, backgroundFlag][, class] [, baggage] , page_type)

Arguments

gfx	A graphics handle.
printer	A string giving the printer name on which to print the document. If you want to print to a file rather than to a printer, specify an empty string; the print file is placed in the directory specified by tempFile, or in the Ap- plixware temporary directory if no directory is specified with tempFile.
colorFlag	Indicates if the document is being printed on a color printer. If you are printing on a color printer, set color to TRUE; otherwise, set color to FALSE. Default is FALSE.
copies	The number of copies to print.
bannerFlag	Indicates whether to include a banner page with the printed document. Set to TRUE if you want a banner page, FALSE if you don't. Default is FALSE.
allPagesFlag	Whether to print all the document pages. Set to TRUE to print all pages, FALSE if you want to print a page range.
startPage	A number indicating the first page to print if you are printing a page range. If TRUE, start_page is ignored.
endPage	A number indicating the last page to print if you are printing a range of pages. If set to TRUE, end_page is ignored.
tempFile	The name of the print file. This file is created to print the document and is deleted when printing is completed (unless printer is set to NULL). If you do not supply a path name for temp_file, the print file is created in the Ap- plixware temporary directory.
backgroundFlag	Indicates whether to print the document in the ``background.'' Set to FALSE to not print in the background. If set to TRUE, a new axmain (an additional Ap- plixware server process) will be created, without checking out

	a new license. That new process will be killed upon completing the print job.
class	One of the following constants that indicates if printing will occur on a PostScript or PCL5 file: PostScript PCL5
baggage	an array of format print_baggage@ .
page type	indicates the form of the presentation that you want to print, as follows: 0 - Slides 1 - Speaker Notes 2 - Handouts (2 per page) 3 - Handouts (3 per page) 4 - Handouts (6 Per Page) 5 - Outline

Description Prints the Graphics or Presents document using the attributes you specify. It is best to assign FALSE to background unless you are printing a very long document (approximately 20 pages or more). Otherwise, it may take longer to invoke the new Applixware process than to print in the foreground.

GR/GFX_PX_BG_IM_PEL@

Sets the background color

Format GR_PX_BG_IM_PEL@()
GFX_PX_BG_IM_PEL@(gfx)

Method [this.px_bg_im_pel@](#)

Arguments gfx A graphics handle.

Description Changes the color image's background color to the foreground color of whichever pixel is selected. To do the same with the foreground color, see [GR_PX_FG_IM_PEL@](#).

GR/GFX_PX_COLORSHIFT@

Shifts colors

Format GR_PX_COLORSHIFT@(red, green, blue)
GFX_PX_COLORSHIFT@(gfx, red, green, blue)

Method [this.px_colorshift@](#)(red, green, blue)

Arguments

gfx	A graphics handle.
red	Indicates how much ``redness" you want to add or remove. Specify a value from -1 to +1.
green	Indicates how much ``greenness" you want to add or remove. Specify a value from -1 to +1.
blue	Indicates how much ``blueness" you want to add or remove. Specify a value from -1 to +1.

Description Modifies the reds, greens and blues (RGBs) of a color image. Each of three broad color components (reds, greens and blues) can either be left unchanged (with a 0 shift), increased (with a shift ranging to a maximum of 1), or decreased (with a shift ranging to a maximum of -1).

GR/GFX_PX_COLOR_ERASE@

Changes one color to another

Format GR_PX_COLOR_ERASE@(colorToErase, desiredColor, entireImageFlag)
GFX_PX_COLOR_ERASE@(gfx, colorToErase, desiredColor, entireImageFlag)

Method [this.px_color_erase@](#)(colorToErase, desiredColor, entireImageFlag)

Arguments

gfx	A graphics handle.
colorToErase	The color that will be erased by changing it to desiredColor.
desiredColor	The color to which colorToErase is changed.
entireImageFlag	A Boolean value which if set to TRUE tells Graphics to make the change. If FALSE, the erasure is set up for interactive use where the user must make the change.

Description Erases pixels by changing them from one color to another. Typically, you would change one color (colorToErase) to the background color.

GR/GFX_PX_FG_IM_PEL@

Sets the foreground color

Format GR_PX_FG_IM_PEL@()
GFX_PX_FG_IM_PEL@(gfx)

Method [this.px_fg_im_pel@](#)

Arguments gfx A graphics handle.

Description Changes the color image's foreground color to the foreground color of whichever pixel is selected. To do the same with the background color, see [GR_PX_BG_IM_PEL@](#).

GR/GFX_PX_FIND_EDGES@

Whitens all four-connected pixels

Format GR_PX_FIND_EDGES@()

Method [this.px_find_edges@](#)
GFX_PX_FIND_EDGES@(gfx)

Arguments gfx A graphics handle.

Description Switches to white the foreground color of any pixel whose four sides are surrounded by pixels of its same foreground color. A four-connected pixel is a pixel whose four sides border pixels of the same foreground color. The effect is to whiten the center of any grouping of same-color pixels that is at least three pixels wide and three pixels high.

GR/GFX_PX_FLOOD@

Fills adjacent cells with the current foreground

Format GR_PX_FLOOD@(fourConnectedFlag, color)

GFX_PX_FLOOD@(gfx, fourConnectedFlag, color)

Method this.px_flood@(fourConnectedFlag, color)

Arguments gfx A graphics handle.
fourConnectedFlag
 A Boolean value which if set to TRUE indicates that all unset pixels in the cell region will be set to color.
 A four-connected pixel is a pixel whose four sides border on pixels of the same foreground color.
color The color to be used when flooding a region.

Description Indicates how an area is flooded and the color used to flood the region. The fourConnectedFlag is set to TRUE when you select Fill Across Diagonals from the Brush Settings dialog box within the Pixel Editor.

GR/GFX_PX_GET_INFO@

Returns Pixel Editor color and fill information

Format infoArray = GR_PX_GET_INFO@()
infoArray = GFX_PX_GET_INFO@(gfx)

Method infoArray = this.px_get_info@

Arguments gfx A graphics handle.

Description Returns the foreground color, background color, and fill across diagonals setting for the selected color image in the Pixel Editor:

infoArray[0] (Foreground color)

 The colormap id for the foreground color.

infoArray[1] (Background color)

 The colormap id for the background color.

infoArray[2] (Fill flag)

 Indicates if Fill across diagonals is on in the Pixel Editor's Fill mode:

0 Fill across diagonals is off.

1 Fill across diagonals is on.

GR/GFX_PX_INVERT@

Inverts colors

Format GR_PX_INVERT@()
GFX_PX_INVERT@(gfx)

Method [this.px_invert@](#)

Arguments gfx A graphics handle.

Description For a color image, inverts the color throughout an image by switching the RGB setting of every pixel to its complement. GR_PX_INVERT@ is called by the Invert toggle button in the Pixel Editor's Image Filters dialog box.

If you are editing a black and white image, this macro toggles black and white pixels.

GR/GFX_PX_MODIFIED@

Indicates if the image has been edited/changed

Format flag = GR_PX_MODIFIED@()
flag = GFX_PX_MODIFIED@(gfx)

Method [flag = this.px_modified@](#)

Arguments gfx A graphics handle.

Description Returns a TRUE or FALSE value where TRUE indicates that the current image contains changes that are not saved to disk.

GR/GFX_PX_RESTART@

Restarts the Pixel Editor

Format GR_PX_RESTART@()
GFX_PX_RESTART@(gfx)

Method [this.px_restart@](#)

Arguments gfx A graphics handle.

Description Restarts the Pixel Editor with the same image as it contained when you first entered the Pixel Editor. However, if you have saved the current image, the Pixel Editor is restarted with the last saved version of the image.
No warning is given that edits will be lost.

GR_PX_SET_BRUSH_BG@

Set the background brush color

Format GR_PX_SET_BRUSH_BG@(colorName)

Method [this.px_set_brush_bg@\(colorName\)](#)

Arguments colorName The string name of the color to which the background will be set.

GR_PX_SET_BRUSH_FG@

Set the foreground brush color

Format GR_PX_SET_BRUSH_FG@(colorName)

Method [this.px_set_brush_fg@\(colorName\)](#)

Arguments colorName The string name of the color to which the foreground will be set.

GR/GFX_PX_SET_FLOOD_STYLE@

Indicates whether same-color Fill mode should apply

Format GR_PX_SET_FLOOD_STYLE@(value)

GFX_PX_SET_FLOOD_STYLE@(gfx, value)

Method [this.px_set_flood_style@\(value\)](#)

Arguments gfx A graphics handle.
value Indicates whether Fill mode should extend to even those same-color pixels that are connected diagonally.
0 Fill only "four-connected pixels"; those same-color pixels that share a whole side with other same-color pixels adjacent to the selection.

- 1 Fill all ``eight-connected pixels"; all contiguous same-color pixels, even those that share a mere edge.

Description A toggle that determines whether the Pixel Editor's same-color Fill mode should apply only to ``four-connected" pixels, or also to ``eight-connected" pixels.

An eight-connected pixel is one that has the same foreground color as all eight pixels that surround it (the four that share one side, plus the four that share just one corner, or vertex).

A four-connected pixel is one that has the same foreground color as all four pixels that share a side with it.

Fill mode changes to the foreground color all pixels that are of the same color as, and are adjacent to, either the selected pixel, or a pixel adjacent to the selected pixel, or a pixel adjacent to that, and so on.

GR/GFX_PX_SET_PEN_DENSITY@

Sets the Spray tool's density

Format GR_PX_SET_PEN_DENSITY@(density)
GFX_PX_SET_PEN_DENSITY@(gfx, density)

Method [this.px_set_pen_density@\(density\)](#)

Arguments

gfx	A graphics handle.
density	The heaviness, or density, desired when using the Spray tool:
0	Solid
1	Heavy
2	Medium
3	Light

Description Sets the density of the Spray tool in the Pixel Editor. GR_PX_SET_PEN_DENSITY@ is called by the Spray level radio button group in the Pixel Editor's Brush Settings dialog box.

GR/GFX_PX_SET_PEN_STYLE@

Sets the pen's style

Format GR_PX_SET_PEN_STYLE@(flag)
GFX_PX_SET_PEN_STYLE@(gfx, flag)

Method `this.px_set_pen_style@(flag)`

Arguments `gfx` A graphics handle.
`flag` A Boolean value which if set to TRUE on a monochrome system toggles the drawing state so the first time the brush is moved, pixels are drawn. The next time pixels are erased, and so on.

Description Sets the pen's drawing state for the monochrome pixel tool (that is, this is equivalent to pressing the Paint in Toggle Mode button within the Brush Settings palette.)

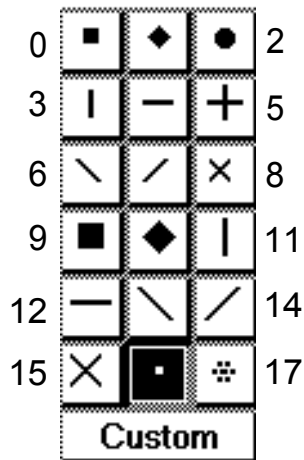
GR/GFX_PX_SET_TOOL@

Sets the "brush shape" for all tools

Format `GR_PX_SET_TOOL@(shape)`
`GFX_PX_SET_TOOL@(gfx, shape)`

Method `this.px_set_tool@(shape)`

Arguments `gfx` A graphics handle.
`shape` The pattern of pixels to be filled when dragging the mouse in the Pixel Editor. You can choose from 18 styles (from 0 to 17).



Shape can also be the path name containing the custom bitmap used for the brush shape.

Description Sets the brush shape for all tools used in the Pixel Editor. The brush shape determines the pattern of pixels to be filled when dragging your mouse in the Pixel Editor.

GR_PX_SET_TOOL@ is called by the panel of Brush shape icons in the Pixel Editor's Brush Settings dialog box.

GR_PX_SHIFT_RGB@

Shifts pixel colors

Format GR_PX_SHIFT_RGB@(red, green, blue)

Method [this.px_shift_rgb@](#)(red, green, blue)

Arguments

red	The <i>redness</i> value to be shifted.
green	The <i>greenness</i> value to be shifted.
blue	The <i>blueness</i> value to be shifted.

Description Changes the red, green, and blue components of the color. The values of the three parameters range from -1 to +1.

GR/GFX_PX_SOFTEN@

Softens the edges throughout a color image

Format GR_PX_SOFTEN@()
GFX_PX_SOFTEN@(gfx)

Method [this.px_soften@](#)

Arguments gfx A graphics handle.

Description Softens the edges throughout a color image by averaging the RGBs of all adjacent pixels. This macro affects only those different color pixels that are immediately adjacent.

For example, if this macro is executed in a color image that consists of two adjacent thick lines of different colors, the only pixels whose color settings will be changed are those at the immediate edge of the two lines.

GR_PX_SOFTEN@ is called by the Soften toggle button in the Pixel Editor's Image Filters dialog box.

GR/GFX_PX_TOGGLE@

Changes the image type

Format GR_PX_TOGGLE@()

Method this.px_toggle@
GFX_PX_TOGGLE@(gfx)

Arguments gfx A graphics handle.

Description Either converts a color image to a black and white image or vice versa. Non-transparent pixels are changed to black while transparent pixels are changed to white.

GR/GFX_QUERY@

Returns a list of handles meeting retrieval criteria

Format handleList = GR_QUERY@(handleList, expression, values)
handleList = GFX_QUERY@(gfx, handleList, expression, values)

Method handleList = this.query@(handleList, expression, values)

Arguments gfx A graphics handle.

queryList An array of handles or the string all.

expression One of the following:

type
selected
at
inside
matching
tagged

You may also use the string not to invert the meaning; for example, not selected.

values Depending on the kind of expression, as follows:

type string
 bounded text

	rectangle
	curve
	rounded rectangle
	ellipse
	regular polygon
	stroke
	line
	polyline
	image mask
	group
selected	NULL
at	gr_area@
inside	gr_area@
matching	gr_attribute@
tagged	tag string

Description Returns a list of handles that meet the retrieval criteria you specify. It does this by merging lists for OR operations; it also performs successive queries on a list for AND operations.

To perform an OR operation between two queries, merge the two lists using the **GR_MERGE_LISTS@** macro.

To create AND operations, perform successive queries on a list.

For example:

- Return a list of all selected objects:

```
list = GR_QUERY@("all", "selected")
```

- Return a list of all selected rounded rectangles:

```
list1 = GR_QUERY@("all", "selected")
```

```
list2 = GR_QUERY@("list1", "type", "rounded rectangle")
```

GFX Read File Macros

Associate a foreign file with a graphics handle

Format GFX_READ_CGM_FILE@(gfx, fileName)
 GFX_READ_DXF_FILE@(gfx, fileName)
 GFX_READ_EPSI_FILE@(gfx, fileName)
 GFX_READ_FAX_FILE@(gfx, fileName)
 GFX_READ_GEM_FILE@(gfx, fileName)
 GFX_READ_GIF_FILE@(gfx, fileName)
 GFX_READ_GP4_FILE@(gfx, fileName)
 GFX_READ_HPGL_FILE@(gfx, fileName)

GFX_READ_IGES_FILE@(gfx, fileName)
 GFX_READ_ILBM_FILE@(gfx, fileName)
 GFX_READ_IRIS_FILE@(gfx, fileName)
 GFX_READ_JPEG_FILE@(gfx, fileName)
 GFX_READ_MPNT_FILE@(gfx, fileName)
 GFX_READ_MSWB_FILE@(gfx, fileName)
 GFX_READ_PBM_FILE@(gfx, fileName)
 GFX_READ_PCX_FILE@(gfx, fileName)
 GFX_READ_PGM_FILE@(gfx, fileName)
 GFX_READ_PICT2_FILE@(gfx, fileName)
 GFX_READ_PICT_FILE@(gfx, fileName)
 GFX_READ_PPM_FILE@(gfx, fileName)
 GFX_READ_PPT_FILE@(gfx, fileName)
 GFX_READ_RAW_FILE@(gfx, fileName)
 GFX_READ_RS_FILE@(gfx, fileName)
 GFX_READ_TGA_FILE@(gfx, fileName)
 GFX_READ_TIF_FILE@(gfx, fileName)
 GFX_READ_WMF_FILE@(gfx, fileName)
 GFX_READ_WPG_FILE@(gfx, fileName)
 GFX_READ_XBM_FILE@(gfx, fileName)
 GFX_READ_XPM_FILE@(gfx, fileName)
 GFX_READ_XWD_FILE@(gfx, fileName)

Arguments

gfx	A graphics handle as returned by CREATE_GRAPHIC@.
filename	The name of a graphics file. This can be either a file on the local file system, or the URL of a graphics file on the internet.

Description GFX read file macros read the contents of a file and associate the data with a graphics handle. Internally, these macros call the macro GFX_READ_FOREIGN_FILE@ with the filter macro and file type appropriate for the given file. The following lists the file type associated with each GFX read file macro.

GFX_READ_CGM_FILE@(gfx, url)	CGM
GFX_READ_DXF_FILE@(gfx, url)	DXF
GFX_READ_EPSI_FILE@(gfx, url)	Encapsulated postscript
GFX_READ_FAX_FILE@(gfx, url)	CCITT Group 3 FAX
GFX_READ_GEM_FILE@(gfx, url)	IMG Raster
GFX_READ_GIF_FILE@(gfx, url)	GIF
GFX_READ_GP4_FILE@(gfx, url)	CCITT Group 4 FAX
GFX_READ_HPGL_FILE@(gfx, url)	Hewlett_Packard plotter format
GFX_READ_IGES_FILE@(gfx, url)	IGES file
GFX_READ_ILBM_FILE@(gfx, url)	ILBM

GFX_READ_IRIS_FILE@(gfx, url)	Silicon Graphics Iris file
GFX_READ_JPEG_FILE@(gfx, url)	JPEG file
GFX_READ_MPNT_FILE@(gfx, url)	Macpaint file
GFX_READ_MSWB_FILE@(gfx, url)	MS Windows Bitmap file
GFX_READ_PBM_FILE@(gfx, url)	Portable Bitmap file
GFX_READ_PCX_FILE@(gfx, url)	PCX file
GFX_READ_PGM_FILE@(gfx, url)	Portable Greymap file
GFX_READ_PICT2_FILE@(gfx, url)	PICT2 file
GFX_READ_PICT_FILE@(gfx, url)	PICT file
GFX_READ_PPM_FILE@(gfx, url)	X11 Portable Pixmap file
GFX_READ_PPT_FILE@(gfx, url)	PPT file
GFX_READ_RAW_FILE@(gfx, url)	Raw bitmap file
GFX_READ_RS_FILE@(gfx, url)	Sun Raster file
GFX_READ_TGA_FILE@(gfx, url)	TGA file
GFX_READ_TIF_FILE@(gfx, url)	TIFF file
GFX_READ_WMF_FILE@(gfx, url)	Windows metafile
GFX_READ_WPG_FILE@(gfx, url)	Wordperfect Graphics file
GFX_READ_XBM_FILE@(gfx, url)	X Windows Bitmap file
GFX_READ_XPM_FILE@(gfx, url)	X Windows Pixmap file
GFX_READ_XWD_FILE@(gfx, url)	X Windows dump

GFX_READ_GR_BUFFER@

Reads the buffer onto the graphic

Format GFX_READ_GR_BUFFER@(gfx, buffer)

Arguments

gfx	A graphics handle
buffer	The variable containing the graphic information

Description Creates an association between a graphics handle and graphics information.

GFX_READ_GR_FILE@

Reads the file onto the graphic

Format GFX_READ_GR_FILE@(gfx, filename)

Arguments gfx A graphics handle
filename The name of a file containing Applixware graphics information.

Description Reads the graphics information in filename into memory and associates a graphic handle with this information.

GFX_READ_FOREIGN_FILE@

Reads the file onto the graphic

Format GFX_READ_FOREIGN_FILE@(gfx, url, doc_type, filter_macro)

Arguments gfx A graphics handle as returned by CREATE_GRAPHIC@.
filename The name of a graphics file. This can be either a file on the local file system, or the URL of a graphics file on the internet.
docType The document format of the file referenced by filename.
filterMacro An Applixware filter macro required to import the target file into an Applixware document.

Description Reads the graphics information in filename into memory and associates a graphic handle with this information.

The supported docTypes and corresponding file filter macros are as follows:

docType	filterMacro
AX_DOC_TYPE_XWD	filter_xwd_to_gr@
AX_DOC_TYPE_FAX_I	filter_fax_to_gr@
AX_DOC_TYPE_GP4	filter_gp4_to_gr@
AX_DOC_TYPE_EPSI	filter_epsi_to_gr@
AX_DOC_TYPE_HPGL	filter_hpgl_to_gr@
AX_DOC_TYPE_X_BITMAP	filter_xbm_to_gr@
AX_DOC_TYPE_PCX	filter_pcx_to_gr@

AX_DOC_TYPE_SUN_RASTER	filter_rs_to_gr@
AX_DOC_TYPE_MAC_PAINT	filter_mpnt_to_gr@
AX_DOC_TYPE_MS_WINDOWS_BITMAP	filter_mswb_to_gr@
AX_DOC_TYPE_CGM	filter_cgm_to_gr@
AX_DOC_TYPE_DXF	filter_dxf_to_gr@
AX_DOC_TYPE_IGES	filter_iges_to_gr@
AX_DOC_TYPE_TIFF_I	filter_tif_to_gr@
AX_DOC_TYPE_GIF	filter_gif_to_gr@
AX_DOC_TYPE_PICT	filter_pict_to_gr@
AX_DOC_TYPE_PICT2	filter_pict2_to_gr@
AX_DOC_TYPE_PPT	filter_ppt_to_gr@
AX_DOC_TYPE_TGA	filter_tga_to_gr@
AX_DOC_TYPE_ILBM	filter_ilbm_to_gr@
AX_DOC_TYPE_GEM	filter_gem_to_gr@
AX_DOC_TYPE_PBM	filter_pbm_to_gr@
AX_DOC_TYPE_PGM	filter_pgm_to_gr@
AX_DOC_TYPE_PPM	filter_ppm_to_gr@
AX_DOC_TYPE_WMF	filter_wmf_to_gr@
AX_DOC_TYPE_WPG	filter_wpg_to_gr@
AX_DOC_TYPE_XPM	filter_xpm_to_gr@
AX_DOC_TYPE_RAW	filter_raw_to_gr@
AX_DOC_TYPE_IRIS	filter_iris_to_gr@
AX_DOC_TYPE_JPEG	filter_jpeg_to_gr@

The AX_DOC_TYPE constants are defined in the file recgfile_.am. This file resides in the Applixware installation path under <install_dir>/axdata/elf.

See also [GFX_READ_GR_FILE@](#)

GR_RECORD_MACRO@

Begins the recording of a macro

Format GR_RECORD_MACRO@()

[Method](#) this.record_macro@

GR/GFX_REMOVE_CURVE@

Removes polycurve curvature

Format GR_REMOVE_CURVE@()
GFX_REMOVE_CURVE@(gfx)

[Method](#) this.remove_curve@

Arguments gfx A graphics handle.

Description Changes the vertices of a curved object from curves to angles. An object must be selected when this macro executes.

This macro is bound to Transform ® Unsmooth.

To change the vertices of a polyline back to a polycurve, use [GR_ADD_CURVE@](#).

GR/GFX_RENAME_FILL@

Defines a new name for a fill pattern

Format GR_RENAME_FILL@(index, newName)
GFX_RENAME_FILL@(gfx, index, newName)

[Method](#) this.rename_fill@(index, newName)

Arguments gfx A graphics handle.
index A number indicating the fill pattern whose name is being changed.
newName The fill pattern's new name.

GR/GFX_RENAME@

Renames the current graphics document

Format GR_RENAME@(newName)
GFX_RENAME@(gfx, newName)

Method `this.rename@(newName)`

Arguments `gfx` A graphics handle
`newName` A string containing the new name of the document

Description Renames the current graphics document to the name specified in the `newName` string.

GR/GFX_RENDER_GRAPHIC@

Places a graphic inset within a spreadsheet

Format `GR_RENDER_GRAPHIC@(format gr_inset@ info)`
`GFX_RENDER_GRAPHIC@(gfx, format gr_inset@ info)`

Method `this.render_graphic@(format gr_inset@ info)`

Arguments `gfx` A graphics handle.
`info` Information describing the file that will be inset.

Description Places a graphic inset within an Appixware Spreadsheets document.

The definition of the `gr_inset@` format is as follows:

```
format gr_inset@
    scale_mode,
    proportional,
    prescale_x,
    prescale_y,
    src_x,
    src_y,
    src_wid,
    src_hyt,
    dst_x,
    dst_y,
    dst_wid,
    dst_hyt,
    window_id
```

GR_REOPEN@

Opens a new document into the current window

Format GR_REOPEN@(filename)

Method [this.reopen@](#)(filename)

Arguments filename The file to be loaded.

Description Loads a new Graphics document into the current Graphics window. If the current document is modified, the user is asked if the document should be saved.

GR/GFX_REPAINT_WINDOW@

Redisplays the Graphics window

Format GR_REPAINT_WINDOW@()

GFX_REPAINT_WINDOW@(gfx)

Method [this.repaint_window@](#)

Arguments gfx A graphics handle.

Description Redraws all the entire Graphics window and all objects within it.

GR/GFX_RETURN_KEY@

Enters a RETURN within text

Format GR_RETURN_KEY@()

GFX_RETURN_KEY@(gfx)

Method [this.return_key@](#)

Arguments gfx A graphics handle.

Description Enters a RETURN within text. If you are using the Text tool, you will create a second text object. If you are using the Formatted Text tool, you will simply be adding another line within the text area.

GR/GFX_REVEAL@

Reveals all hidden objects

Format GR_REVEAL@()
GFX_REVEAL@(gfx)

Method [this.reveal@](#)

Arguments gfx A graphics handle.

Description Reveals all objects that have previously been hidden using either [GR_HIDE_SELECTED@](#), or [GR_HIDE_UNSELECTED@](#). To reveal *and select* all hidden objects, see [GR_SELECT_REVEAL@](#).

GR/GFX_REVEAL_LAYER@

Displays a hidden layer

Format GR_REVEAL_LAYER@(layerName)
GFX_REVEAL_LAYER@(gfx, layerName)

Method [this.reveal_layer@\(layerName\)](#)

Arguments gfx A graphics handle.
layerName The name of a layer. If this layer does not exist, it will be created. Because it will have been just created, no objects will be on it; hence, nothing will be displayed.

Description Displays all objects that are contained on a layer. These objects were previously hidden (made invisible) using [GR_HIDE_LAYER@](#).

GR/GFX_REVERT@

Restores a Graphics document to its previously saved state

Format GR_REVERT@()
GFX_REVERT@(gfx)

Method [this.revert@](#)

Arguments gfx A graphics handle.

Description Restores a Graphics document to its state when last saved, thereby eliminating all changes made to the document since this save. The document remains open. Unlike the File ® Revert menu option, GR_REVERT@ does not prompt to verify that you really wants to restore the document.

GR/GFX_RIGHT_ARROW_KEY@

Moves right

Format GR_RIGHT_ARROW_KEY@(
GFX_RIGHT_ARROW_KEY@(gfx)

Method [this.right_arrow_key@](#)

Arguments gfx A graphics handle.

Description Moves selected objects to the right or moves the cursor within text one character to the right.

See also [GR_LEFT_ARROW_KEY@](#)

GR/GFX_RIGHT_SCREEN_KEY@

Scrolls the display one screen to the right

Format GR_RIGHT_SCREEN_KEY@(
GFX_RIGHT_SCREEN_KEY@(gfx)

Method [this.right_screen_key@](#)

Arguments gfx A graphics handle.

Description Displays the graphic objects that are immediately to the right of the objects currently being displayed.

GR/GFX_ROTATE@

Places an object in rotate mode

Format GR_ROTATE@()
GFX_ROTATE@(gfx)

Method [this.rotate@](#)

Arguments gfx A graphics handle.

Description Places the selected object or objects in rotate mode. While in rotate mode, an object can be rotated by moving the mouse pointer to a center of rotation and dragging the mouse to rotate the object. If multiple objects are in rotate mode, the objects are rotated as a group.

GR_ROTATE@ is only effective for a single rotation. If you perform an action other than rotating after executing GR_ROTATE@, rotate mode is ended.

GR_ROTATE@ is called by the Transform ® Rotate menu option.

GR/GFX_ROTATE_EXACT@

Rotates an object or objects

Format GR_ROTATE_EXACT@(degrees)
GFX_ROTATE_EXACT@(gfx, degrees)

Method [this.rotate_exact@\(degrees\)](#)

Arguments gfx A graphics handle.
degrees A number indicating the number of degrees to rotate the selected objects. If degrees is positive, the object is rotated counterclockwise. If degrees is negative, the object is rotated clockwise.

Description Rotates an object or objects the number of degrees specified. Rotation is with respect to object centers. If no objects are selected, an error is thrown. If a rectangle or ellipse is rotated at other than a 90, 180, 270, or 360 degree angle, the object will be redefined as a polyline.

GR/GFX_RULERS@

Sets the ruler display in the current Graphics document

Format GR_RULERS@()

Method [this.rulers@](#)

GFX_RULERS@(gfx)

Arguments gfx A graphics handle.

Description Toggles the display of the horizontal and vertical rulers in the Graphics document on and off. If rulers are not currently displayed, GR_RULERS@ displays ruler. If rulers are currently displayed, GR_RULERS@ turns off the display of rulers.

GR_RULERS@ is called by the View ® Rulers menu option.

GR/GFX_SAVE@

Saves the current Graphics document

Format GR_SAVE@()

GFX_SAVE@(gfx)

Method [this.save@](#)

Arguments gfx A graphics handle.

Description Writes the current Graphics document to a file. If the document has not been saved before, it will be saved to a file having the name displayed in the window title.

GR/GFX_SAVE_AS@

Saves a Graphics document using specified attributes

Format GR_SAVE_AS@(format doc_format_info)

GFX_SAVE_AS@(gfx, format doc_format_ info)

Method this.save_as@(format doc_format_ info)

Arguments gfx A graphics handle.
info An array of information indicating the name, mode, and permissions for the file. The FORMAT template for array is doc_format_. The header file containing this template is fileinf_.am.

Description Saves a Graphics document using the attributes you specify.

The definition for the doc_format_ format is as follows:

format doc_format_

name, The full path name of the document to be saved, including the .ag extension. If array.name already exists, the existing file is replaced with this new one without any warning.

docid, Not used; set it to NULL.

on_disk, Not used; set it to NULL.

save_mode, (Optional:) The format in which to save the document. This mode can be:
 1 Uncompressed Bit 6
 11 Compressed Bit 6
 15 Uncompressed Hex
 16 Compressed Hex

grp_access, (Optional:) A number indicating the read and write permissions for the file for a member of the same group. array.grp_access can be one of the following:
 0 (The default:) No read or write permission for the file
 1 Read permission for the file.
 2 Read and write permission for the file.

all_access, (Optional:) A number indicating the read and write permissions for the file for any user. array.all_access can be 0, 1, or 2, as described above. The default is 0.

writeable Not Used.

GR/GFX_SCALE_EXACT@

Reduces or increases an object's size

Format GR_SCALE_EXACT@(horizontal, vertical)

GFX_SCALE_EXACT@(gfx, horizontal, vertical)

[Method](#) this.scale_exact@(horizontal, vertical)

Arguments gfx A graphics handle.

horizontal A number indicating the factor by which to scale the object on the horizontal axis.

To shrink the object horizontally, set horizontal to a number in the range from 0 to 1. For example, a horizontal setting of .5 will reduce the selected object horizontally to half its current size.

To enlarge the object horizontally, set horizontal to a decimal number greater than 1. For example, a setting of 3 will enlarge the selected object horizontally to three times its current size.

vertical A number indicating the factor by which to scale the object on the vertical axis.

To shrink the object vertically, set vertical to a number in the range from 0 to 1. To enlarge the object vertically, set vertical to a decimal number greater than 1.

Description Scales selected objects horizontally and vertically according to the scaling factors specified. To scale an object proportionally, specify the same number for both the horizontal and vertical arguments. An objects' center remains in the same position during scaling.

GR/GFX_SELECT_ALL@

Selects all objects

Format GR_SELECT_ALL@()
GFX_SELECT_ALL@(gfx)

[Method](#) this.select_all@

Arguments gfx A graphics handle.

Description Selects all objects in the current Graphics document. GR_SELECT_ALL@ is called by the Edit ® Select ® All menu option.

See also [GR_SELECT_CLEAR@](#)
[GR_SELECT_LAST@](#)
[GR_SELECT_NEXT@](#)
[GR_SELECT_PREVIOUS@](#)

[GR SELECT REVERSE@](#)
[GR SELECT TYPE@](#)
[GR SELECT ALL IN LAYER@](#)

GR/GFX_SELECT_ALL_IN_LAYER@

Selects all objects in a layer

Format GR_SELECT_ALL_IN_LAYER@(layerName)
GFX_SELECT_ALL_IN_LAYER@(gfx, layerName)

Method [this.select_all_in_layer@\(layerName\)](#)

Arguments gfx A graphics handle.
layerName The name of a layer. If the layer does not exist, it will be created.

See also [GR SELECT ALL@](#)
[GR SELECT CLEAR@](#)
[GR SELECT LAST@](#)
[GR SELECT NEXT@](#)
[GR SELECT PREVIOUS@](#)
[GR SELECT REVERSE@](#)
[GR SELECT TYPE@](#)

GR/GFX_SELECT_CLEAR@

Unselects selected items

Format GR_SELECT_CLEAR@()
GFX_SELECT_CLEAR@(gfx)

Method [this.select_clear@](#)

Arguments gfx A graphics handle.

Description Deselects all objects; that is, after this macro executes, no object within the Graphics window is selected.

See also [GR SELECT ALL@](#)

[GR_SELECT_LAST@](#)
[GR_SELECT_NEXT@](#)
[GR_SELECT_PREVIOUS@](#)
[GR_SELECT_REVERSE@](#)
[GR_SELECT_TYPE@](#)
[GR_SELECT_ALL_IN_LAYER@](#)

GR/GFX_SELECT_LAST@

Selects/unselects last object created

Format GR_SELECT_LAST@(
gfx, GFX_SELECT_LAST@()

Method [this.select_last@](#)

Arguments gfx A graphics handle.

See also [GR_SELECT_ALL@](#)
[GR_SELECT_CLEAR@](#)
[GR_SELECT_NEXT@](#)
[GR_SELECT_PREVIOUS@](#)
[GR_SELECT_REVERSE@](#)
[GR_SELECT_TYPE@](#)
[GR_SELECT_ALL_IN_LAYER@](#)

GR/GFX_SELECT_NEXT@

Selects *next* object

Format GR_SELECT_NEXT@(
GFX_SELECT_NEXT@(gfx)

Method [this.select_next@](#)

Arguments gfx A graphics handle.

Description Selects the object created immediately after the currently selected object.

See also [GR_SELECT_ALL@](#)
[GR_SELECT_CLEAR@](#)
[GR_SELECT_LAST@](#)
[GR_SELECT_PREVIOUS@](#)
[GR_SELECT_REVERSE@](#)
[GR_SELECT_TYPE@](#)
[GR_SELECT_ALL_IN_LAYER@](#)

GR/GFX_SELECT_PREVIOUS@

Selects *previous* object

Format GR_SELECT_PREVIOUS@()
GFX_SELECT_PREVIOUS@(gfx)

Method [this.select_previous@](#)

Arguments gfx A graphics handle.

Description Selects the object created immediately before the currently selected object.

See also [GR_SELECT_ALL@](#)
[GR_SELECT_CLEAR@](#)
[GR_SELECT_LAST@](#)
[GR_SELECT_NEXT@](#)
[GR_SELECT_REVERSE@](#)
[GR_SELECT_TYPE@](#)
[GR_SELECT_ALL_IN_LAYER@](#)

GR/GFX_SELECT_REVEAL@

Reveals all hidden objects in the current Graphics document

Format GR_SELECT_REVEAL@()
GFX_SELECT_REVEAL@(gfx)

Method [this.select_reveal@](#)

Arguments gfx A graphics handle.

Description Reveals all objects previously hidden using of the following macros:

[GR_HIDE_SELECTED@](#)

[GR_HIDE_UNSELECTED@](#)

After selecting this macro, the revealed objects are selected.

GR_SELECT_REVEAL@ is called by the View ® Reveal All menu option.

To reveal objects without selecting them, use [GR_REVEAL@](#).

GR/GFX_SELECT_REVERSE@

Selects any objects that are not selected and deselects all objects that are selected

Format GR_SELECT_REVERSE@()
GFX_SELECT_REVERSE@(gfx)

[Method](#) this.select_reverse@

Arguments gfx A graphics handle.

Description GR_SELECT_REVERSE@ is called by the Edit ® Select ® Reverse menu option.

See also [GR_SELECT_ALL@](#)
[GR_SELECT_CLEAR@](#)
[GR_SELECT_LAST@](#)
[GR_SELECT_NEXT@](#)
[GR_SELECT_PREVIOUS@](#)
[GR_SELECT_TYPE@](#)
[GR_SELECT_ALL_IN_LAYER@](#)

GR/GFX_SELECT_TEXT_RANGE@

Selects characters

Format GR_SELECT_TEXT_RANGE@(pos, numChar)
GFX_SELECT_TEXT_RANGE@(gfx, pos, numChar)

Method this.select_text_range@(pos, numChar)

Arguments gfx A graphics handle.
pos The position at which selection will begin.
numChar The number of characters to be selected.

Description Selects one or more characters within text beginning at position pos within the text.

GR_SELECT_TYPE@

Selects all objects of a particular type

Format GR_SELECT_TYPE@()

Method this.select_type@

Description Selects all objects in the current Graphics document that are of the same type as the tool currently selected in the tools area. For example, if the rectangle tool is selected when GR_SELECT_TYPE@ is executed, all rectangle objects in the document are selected.

GR_SELECT_TYPE@ is called by the Edit ® Select ® Object Type menu option.

See also [GR_SELECT_ALL@](#)
[GR_SELECT_CLEAR@](#)
[GR_SELECT_LAST@](#)
[GR_SELECT_NEXT@](#)
[GR_SELECT_PREVIOUS@](#)
[GR_SELECT_REVERSE@](#)
[GR_SELECT_ALL_IN_LAYER@](#)

GR/GFX_SET_ATTR@

Sets graphic attributes

Format GR_SET_ATTR@(format gr_attribute@ attr)
GFX_SET_ATTR@(gfx, format gr_attribute@ attr)

Method this.set_attr@(format gr_attribute@ attr)

Arguments gfx A graphics handle.
attr The attributes being set. For a definition of the gr_attribute@ format, see [GR_CHART_GET_ATTR@](#).

Description Sets the following kinds of graphic attributes: back fill, line fill, shadow, line style, text field, and text attributes.

See also [GR_GET_ATTR@](#)

GR/GFX_SET_BG_PIXEL_COLOR@

Sets the background color for the color image

Format GR_SET_BG_PIXEL_COLOR@(bgColorIndex)

Method this.set_bg_pixel_color@(bgColorIndex)
GFX_SET_BG_PIXEL_COLOR@(gfx, bgColorIndex)

Arguments gfx A graphics handle.
bgColorIndex An index into the current colormap.

Description Sets the background color to a color within the colormap.

See also [GR_SET_FG_PIXEL_COLOR@](#)

GR/GFX_SET_CALLBACK@

Associates a callback macro with the current selection

Format GR_SET_CALLBACK@(index)
GFX_SET_CALLBACK@(gfx, index)

Method this.set_callback@(index)

Arguments gfx A graphics handle.
index A number indicating one of the callback macros associated with this graphics document. To obtain a list of all callback macros, use [GR_GET_CALLBACKS@](#).

Description Associates a callback macro with the current selection. This callback macro was originally associated with the Graphics document using [GR_CREATE_CALLBACK@](#).

GR/GFX_SET_CHAR_ATTS@

Sets the attributes for text

Format GR_SET_CHAR_ATTS@(face, size, ruling, boldFlag, italicFlag, scale, shear, angle, xOffset, yOffset, strikeThroughFlag)

GFX_SET_CHAR_ATTS@(gfx, face, size, ruling, boldFlag, italicFlag, scale, shear, angle, xOffset, yOffset, strikeThroughFlag)

Method [this.set_char_atts@](#)(face, size, ruling, boldFlag, italicFlag, scale, shear, angle, xOffset, yOffset, strikeThroughFlag)

Arguments

gfx	A graphics handle.
face	The text font.
size	The point size.
ruling	A numeric value indicating if the text is drawn with: 0 No underlining 1 Single underlining 2 Double underlining
boldFlag	A Boolean value where TRUE indicates that the text is displayed in bold face.
italicFlag	A Boolean value where TRUE indicates that the text is displayed in italics.
scale	A number indicating a horizontal scaling factor applied to text. A number greater than 1 increases the horizontal size of the text; a number between 1 and 0 decreases the horizontal size of the text. A number between -1 and 0 decreases the horizontal size of the text and reverses the text; a number less than -1 increases the horizontal size of the text and reverses the text.
shear	A numeric value that indicates the shear factor applied to text. Shearing text stretches it so that the bottom of the text remains in the same position but the top of the text is shifted horizontally, giving the text a sloping effect. Entering -1 stretches selected text 45 degrees to the right. Entering 1 stretches selected text 45 degrees to the left.

angle	A number (in degrees) indicating the angle at which the text is rotated. A positive number rotates the text counter-clockwise; a negative number rotates the text clockwise.
xOffset	A number (indicating points) that changes the spacing between characters of selected text. A positive number increases the spacing between characters and a negative number reduces the spacing between characters.
yOffset	A number (indicating points) that moves the selected text up or down. (This is used to create subscripts and superscripts.) A positive number moves the selected text up and a negative number moves the selected text down.
strikeThroughFlag	A Boolean value where TRUE indicates that a line is drawn through text.

Description Sets the text attributes listed above.

GR/GFX_SET_COLOR_BY_CMAP_INDEX@

Sets color attributes

Format GR_SET_COLOR_BY_CMAP_INDEX@(infoArray)
GFX_SET_COLOR_BY_CMAP_INDEX@(gfx, infoArray)

Method [this.set_color_by_cmap_index@\(infoArray\)](#)

Arguments

gfx	A graphics handle.
infoArray[0]	The color map index.
infoArray[1]	One of the following values:
0	Set the fill foreground color
1	Set the fill background color
2	Set the line foreground color
3	Set the line background color
4	Set the drop shadow color

Description Sets the color for one of the elements listed above.

GR/GFX_SET_COLORMAP@

Sets a new color map

Format GR_SET_COLORMAP@(format arrayof gr_color@ color[, colorRemapTable])
GFX_SET_COLORMAP@(gfx, format arrayof gr_color@ color[, colorRemapTable])

Method [this.set_colormap@](#)(format arrayof gr_color@ color[, colorRemapTable])

Arguments gfx A graphics handle.
color The colors in the new color map table
colorRemapTable
 Color value integers that equate different color values

Description Sets a new color map for Applixware Graphics. This color map replaces the existing color map.

The definition of the gr_color@ format is as follows:

format gr_color@
 c, m, y, k,
 see_thru,
 ink_type,
 name

GR/GFX_SET_DISPLAY_RESOLUTION@

Sets the zoom factor

Format GR_SET_DISPLAY_RESOLUTION@(xResolution%, yResolution%)
GFX_SET_DISPLAY_RESOLUTION@(gfx, xResolution%, yResolution%)

Method [this.set_display_resolution@](#)(xResolution%, yResolution%)

Arguments gfx A graphics handle.
xResolution% The display resolution in X. This number is expressed as a percent;
 that is, 80 represents 80%.
yResolution% The display resolution in I. This number is expressed as a percent;
 that is, 80 represents 80%.

GR/GFX_SET_DPI@

Sets the DPI to 1000

Format 1000 = GR_SET_DPI@()
1000 = GFX_SET_DPI@(gfx)

Method 1000 = this.set_dpi@

Arguments gfx A graphics handle.

Description Sets the Dots Per Inch value of Graphics to 1000.

See also [GR_GET_DPI@](#)

GR/GFX_SET_ELL@

Sets the starting and ending angles for an ellipse

Format GR_SET_ELL@(format gr_ellipse_info@ info)
GFX_SET_ELL@(gfx, format gr_ellipse_info@ info)

Method this.set_ell@(format gr_ellipse_info@ info)

Arguments gfx A graphics handle.
info A variable containing the starting and ending angles.

Description Sets the starting and ending angles for the portion of the ellipse that is displayed. The definition of gr_ellipse_info@ is as follows:

format gr_ellipse_info@
 angle1, 'Starting angle
 angle2 'Ending angle

GR/GFX_SET_EDIT_MODE@

Sets add, delete, or select mode

Format GR_SET_EDIT_MODE@([num])
GFX_SET_EDIT_MODE@(gfx, [num])

Method `this.set_edit_mode@([num])`

Arguments gfx A graphics handle.
 num

Description Places the mouse into modes where one of the following actions can be performed:

- 0 Selects a point on a line
- 1 Adds a point to a line
- 2 Deletes a point on a line

GR/GFX_SET_ENABLE_ADD_MODE@

Mouse clicks add points to a line

Format `GR_SET_ENABLE_ADD_MODE@()`
`GFX_SET_ENABLE_ADD_MODE@(gfx)`

Method `this.set_enable_add_mode@`

Arguments gfx A graphics handle.

Description Places the mouse into a mode where mouse clicks add points to a line. To exit from this mode, place the mouse in delete or select mode.

See also [GR SET_ENABLE_DELETE_MODE@](#)
[GR SET_ENABLE_SELECT_MODE@](#)

GR/GFX_SET_ENABLE_DELETE_MODE@

Mouse clicks delete points on a line

Format `GR_SET_ENABLE_DELETE_MODE@()`
`GFX_SET_ENABLE_DELETE_MODE@(gfx)`

Method `this.set_enable_delete_mode@`

Arguments gfx A graphics handle.

Description Places the mouse into a mode where mouse clicks delete points on a line. To exit from this mode, place the mouse in add or select mode.

See also [GR_SET_ENABLE_ADD_MODE@](#)
[GR_SET_ENABLE_SELECT_MODE@](#)

GR/GFX_SET_ENABLE_SELECT_MODE@

Mouse clicks select points on a line

Format GR_SET_ENABLE_SELECT_MODE@()
GFX_SET_ENABLE_SELECT_MODE@(gfx)

Method [this.set_enable_select_mode@](#)

Arguments gfx A graphics handle.

Description Places the mouse into a mode where mouse clicks select points on a line. To exit from this mode, place the mouse in add or delete mode.

See also [GR_SET_ENABLE_ADD_MODE@](#)
[GR_SET_ENABLE_DELETE_MODE@](#)

GR/GFX_SET_FACE@

Defines the global typeface

Format GR_SET_FACE@(fontName)
GFX_SET_FACE@(gfx, fontName)

Method [this.set_face@\(fontName\)](#)

Arguments gfx A graphics handle.

typefaceName The name of a font used within Graphics. For a list of these fonts, use [GR_LIST_FONT_FAMILIES@](#).

GR/GFX_SET_FG_PIXEL_COLOR@

Sets the foreground color

Format GR_SET_FG_PIXEL_COLOR@(fgColorIndex)
GFX_SET_FG_PIXEL_COLOR@(gfx, fgColorIndex)

Method `this.set_fg_pixel_color@(fgColorIndex)`

Arguments `gfx` A graphics handle.
`fgColorIndex` A number indicating a colormap value.

Description Sets the foreground color to the color pointed to by the `fgColorIndex`.
See also [GR SET BG PIXEL COLOR@](#).

GR/GFX_SET_FIELD_ATTS@

Sets field attributes

Format `GR_SET_FIELD_ATTS@(lineHeight, horizScale, vertScale, shear, angle, horizAlign, vAlign, leftMargin, rightMargin, topMargin, bottomMargin)`
`GFX_SET_FIELD_ATTS@(gfx, lineHeight, horizScale, vertScale, shear, angle, horizAlign, vAlign, leftMargin, rightMargin, topMargin, bottomMargin)`

Method `this.set_field_atts@(lineHeight, horizScale, vertScale, shear, angle, horizAlign, vAlign, leftMargin, rightMargin, topMargin, bottomMargin)`

Arguments

<code>gfx</code>	A graphics handle.
<code>lineHeight</code>	Sets the amount of space between lines of formatted text. Enter the number, in points, to separate each line by in this entry area.
<code>horizScale</code>	Sets the percentage to use to scale the selected text in a horizontal direction. A number greater than 1 increases the horizontal size of the text; a number between 1 and 0 decreases the horizontal size of the text. A number between -1 and 0 decreases the horizontal size of the text and reverses the text; a number less than -1 increases the horizontal size of the text and reverses the text.
<code>vertScale</code>	Sets the percentage to use to scale the selected text in a vertical direction. A number greater than 1 increases the vertical size of the text; a number between 1 and 0 decreases the vertical size of the text. A number between -1 and 0 decreases the vertical size of the text and reverses the text; a number less than -1 increases the vertical size of the text and reverses the text.
<code>shear</code>	Shearing text stretches it so that the bottom of the text remains in the same position but the top of the text is shifted horizontally, giving the text a sloping effect.

	To shear selected text, use a number which represents the factor to shear by in this entry box. Entering -1 stretches selected text 45 degrees to the right. Entering 1 stretches selected text 45 degrees to the left.
angle	Sets the number of degrees by which the selected text is rotated around its center. A positive number rotates the text counter-clockwise; a negative number rotates the text clockwise.
horizAlign	One of the following values: 0 left 1 center 2 right 3 justify
vAlign	Reserved for future use.
leftMargin	The field's left margin.
rightMargin	The field's right margin.
topMargin	The field's top margin.
bottomMargin	The field's bottom margin

Description Sets the field attributes listed above.

GR/GFX_SET_FILL@

Sets the fill type

Format GR_SET_FILL@(index)
GFX_SET_FILL@(gfx, index)

Method [this.set_fill@\(index\)](#)

Arguments gfx A graphics handle.
index A number between 0 and 28 representing the fill pattern shown in the illustration within the description section of this macro.

Description Sets the pattern that will be used to fill objects.

0 1 13 2 11 3 22 4 14 5

6 7 8 9 10 21 12 19 20 23
24 15 16 18 17 25 26 27 28

The numbers above and below the fill diagram indicate the index number. (This command does not let you set a custom fill pattern.)

GR/GFX_SET_FILL_BITMAP@

Defines a fill bitmap

Format GR_SET_FILL_BITMAP@(bitmapName, path, useColorFlag)
GFX_SET_FILL_BITMAP@(gfx, bitmapName, path, useColorFlag)

Method [this.set_fill_bitmap@](#)(bitmapName, path, useColorFlag)

Arguments gfx A graphics handle.
bitmapName A name; this is an arbitrary name that you will use when manipulating the bitmap.
path The full path name of the file containing the bitmap.
useColorFlag A Boolean value which if set to TRUE indicates that the bitmap uses color.

Description Adds a new bitmap to the bitmap table.

GR/GFX_SET_FILL_GRADIENT@

Sets the object gradient fill pattern

Format GR_SET_FILL_GRADIENT@(type)

Method [this.set_fill_gradient@](#)(type)

GFX_SET_FILL_GRADIENT@(gfx, type)

Arguments gfx A graphics handle.
 type One of the following:
 linear
 radial
 rectangular

Description Sets the kind of gradient fill pattern that will be used when filling an object.

See also [GR SET GRADIENT@](#)

GR/GFX_SET_GRADIENT@

Sets line or fill gradients

Format GR_SET_GRADIENT@(lineFlag, type)
 GFX_SET_GRADIENT@(gfx, lineFlag, type)

Method this.set_gradient@(lineFlag, type)

Arguments gfx A graphics handle.
 lineFlag A Boolean value where TRUE indicates that the gradient is being applied to a line pattern. FALSE indicates that it is being applied to an object's fill pattern.
 type One of the following:
 linear
 radial
 rectangular

Description Indicates the gradient style to be applied to the line or fill area.

See also [GR SET FILL GRADIENT@](#)
[GR SET LINE GRADIENT@](#)

GR/GFX_SET_GRID_FACTOR@

Defines the grid spacing

Format GR_SET_GRID_FACTOR@(factor)
GFX_SET_GRID_FACTOR@(gfx, factor)

Method [this.set_grid_factor@](#)(factor)

Arguments gfx A graphics handle.
factor A decimal number, where 1.0 means (depending of the current units) either 1 dot to the inch or 1 dot to the centimeter

Description Sets the interval at which grids occur.

GR/GFX_SET_HEADERS_AND_FOOTERS@

Sets the text and format of the document's headers and footers

Format GR_SET_HEADERS_AND_FOOTERS@(format hdrftr_info info)
GFX_SET_HEADERS_AND_FOOTERS@(gfx, format hdrftr_info info)

Method [this.set_headers_and_footers@](#)(format hdrftr_info info)

Arguments gfx A graphics handle.
info A data structure containing header and footer information.

Description Defines the format and the text of the document's header and footer. For a definition of the hdrftr_info format, see [GR SET HEADERS AND FOOTERS@](#).

GR/GFX_SET_HOOK@

Causes a macro to execute every time you open a Graphics document

Format GR_SET_HOOK@(macroName)
GFX_SET_HOOK@(gfx, macroName)

Method [this.set_hook@](#)(macroName)

Arguments gfx A graphics handle.
macroName The name of the macro. If macroName is not loaded when this macro executes, ELF will load, compile, and execute a file with the same name (after a .am extension is appended) as macroName that exists in your default path.

Description Embeds within a Graphics document an instruction to run a given macro every time a Graphics document is opened. To unset the GR_SET_HOOK@ ``hook" in a Graphics document, run GR_SET_HOOK@ without any arguments.

See also [GR_GET_HOOK@](#)

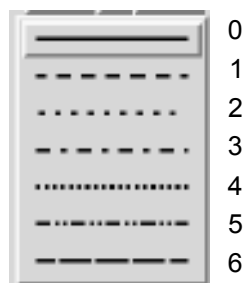
GR/GFX_SET_LINE_ATT@

Sets line attributes

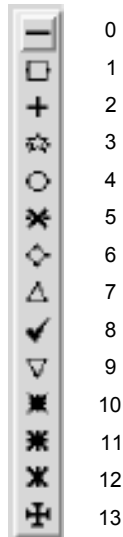
Format GR_SET_LINE_ATT@(style, joinSymbol, firstSymbol, finalSymbol)
GFX_SET_LINE_ATT@(gfx, style, joinSymbol, firstSymbol, finalSymbol)

[Method](#) this.set_line_atts@(style, joinSymbol, firstSymbol, finalSymbol)

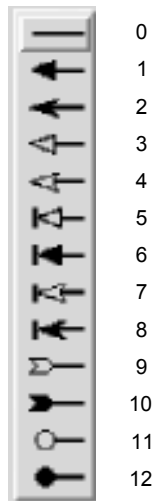
Arguments gfx A graphics handle.
style The style to be used when drawing a line. Use one of the following numeric values:



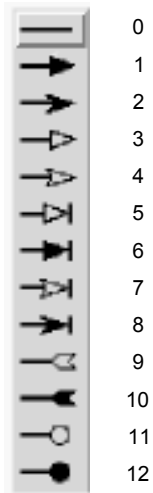
joinSymbol The symbol to be used when lines are joined. Use one of the following numeric values:



firstSymbol The symbol to be used at a line's beginning. Use one of the following numeric values:



finalSymbol The symbol used at the end of a line, as follows:



Description Sets the style used to draw a line as well as the symbols that are used at the line's beginning, midpoints (if a polyline), and ends.

GR/GFX_SET_LINE_BITMAP@

Defines the bitmap pattern used to draw a line

Format GR_SET_LINE_BITMAP@(bitmapName, path, useColorFlag)
 GFX_SET_LINE_BITMAP@(gfx, bitmapName, path, useColorFlag)

Method [this.set_line_bitmap@](#)(bitmapName, path, useColorFlag)

Arguments gfx A graphics handle.
 bitmapName The name of the bitmap that will be used.
 path The full path name of the file containing the bitmap image.
 useColorFlag A Boolean value where TRUE means that color will be used when displaying the bitmap (if the bitmap contains color values).

GR/GFX_SET_LINE_FILL@

Sets the line's fill type

Format GR_SET_LINE_FILL@(type)
 GFX_SET_LINE_FILL@(gfx, type)

Method `this.set_line_fill@(type)`

Arguments `gfx` A graphics handle.
`type` One of the following:
linear
radial
rectangular

Description Sets the style of fill pattern to be used when drawing a line.

See also [GR_GET_LINEFILL@](#)

GR/GFX_SET_LINE_GRADIENT@

Sets a line's gradient fill

Format `GR_SET_LINE_GRADIENT@(type)`
`GFX_SET_LINE_GRADIENT@(gfx, type)`

Method `this.set_line_gradient@(type)`

Arguments `gfx` A graphics handle.
`type` One of the following:
linear
radial
rectangular

Description Sets the kind of gradient fill that will be applied to a line.

See also [GR_SET_GRADIENT@](#)

GR/GFX_SET_LINE_WIDTH@

Sets a line's width

Format `GR_SET_LINE_WIDTH@(weight)`
`GFX_SET_LINE_WIDTH@(gfx, weight)`

Method `this.set_line_width@(weight)`

Arguments `gfx` A graphics handle.

weight Width of a line in points.

Description Sets a line's width in points.

GR/GFX_SET_PAGE@

Moves to the specified graphics page

Format GR_SET_PAGE@(pageNumber)
GFX_SET_PAGE@(gfx, pageNumber)

Method [this.set_page@\(pageNumber\)](#)

Arguments gfx A graphics handle.
pageNumber A page number within the graphics document.

GR/GFX_SET_PAGE_SETUP@

Sets print setup attributes for a Graphics document

Format GR_SET_PAGE_SETUP@(format page_setup_ attributes)
GFX_SET_PAGE_SETUP@(gfx, format page_setup_ attributes)

Method [this.set_page_setup@\(format page_setup_ attributes\)](#)

Arguments gfx A graphics handle.
attributes An array indicating the attributes to set for a graphics document. (The FORMAT template for GR_SET_PAGE_SETUP@ is named page_setup_. The header file containing the FORMAT template is hdrftr_.am.) The elements of the array are as follows:

- height A number indicating the height of the document.
- width A number indicating the width of the document.
- landscape Indicates the orientation of the document. Specify 0 for portrait orientation, 1 for landscape orientation.

Description Sets the print setup attributes for the current Graphics document.

See also [GR_GET_PAGE_SETUP@](#)

GR/GFX_SET_PALETTE_LINE@

Defines the width of lines 1, 2, or 3 in the palette

Format GR_SET_PALETTE_LINE@(lineIndex, width)

GFX_SET_PALETTE_LINE@(gfx)

Method [this.set_palette_line@](#)(lineIndex, width)

Arguments

gfx	A graphics handle.
lineIndex	A number (either 1, 2, or 3) indicating which of the three palette lines is being set.
width	The new width of the palette line.

GR/GFX_SET_PERCENT_FILL@

Sets the percent to which an area is filled

Format GR_SET_PERCENT_FILL@(value)

GFX_SET_PERCENT_FILL@(gfx, value)

Method [this.set_percent_fill@](#)(value)

Arguments

gfx	A graphics handle.
value	A number from 0 to 1000 indicating the extent to which an area is filled.

GR/GFX_SET_PERCENT_LINE_FILL@

Sets the percent to which the line is filled

Format GR_SET_PERCENT_LINE_FILL@(value)

GFX_SET_PERCENT_LINE_FILL@(gfx, value)

Method [this.set_percent_line_fill@](#)(value)

Arguments

gfx	A graphics handle.
value	A number from 0 to 100 indicating the extent to which an area is filled.

GR/GFX_SET_PIXEL_COLOR@

Sets the Pixel Editor's pixel color

Format GR_SET_PIXEL_COLOR@(color)
GFX_SET_PIXEL_COLOR@(gfx, color)

Method [this.set_pixel_color@](#)(color)

Arguments gfx A graphics handle.
color The number of the color that will become the default color.

GR/GFX_SET_POINT_SIZE@

Defines the default point size

Format GR_SET_POINT_SIZE@(size)

Method [this.set_point_size@](#)(size)
GFX_SET_POINT_SIZE@(gfx, size)

Arguments gfx A graphics handle.
size The new default point size.

GR/GFX_SET_PRINT_INFO@

Defines print information for a graphic file

Format GR_SET_PRINT_INFO@(format print_info@ setup)
GFX_SET_PRINT_INFO@(gfx, format print_info@ setup, rescale, page_type)

Method [this.set_print_info@](#)(format print_info@ setup)

Arguments gfx A graphics handle.
setup The information being defined for printing a Graphics document.
rescale 0 = Objects Remain in their current Positions

1 = Re-align objects to upper left margins
 2 = Center object on the new paper center
 3 = Scale objects to new paper size
 4 = Center and scale objects proportionally
 page_type 0 = Graphics Editor
 1 = Slide Editor
 2 = Slide Master
 4 = Handout Master
 6 = Outline Master
 7 = Notes Editor
 8 = Notes Master

Description Uses the information in the `print_info@` format to define how a Graphics document will be printed. The definition of this format is:

```

format print_info@
  height,
  width,
  upper_margin,
  lower_margin,
  inner_margin,
  outer_margin,
  bleed,
  landscape,
  crop_marks,
  registration_marks,
  content_border,
  printer,
  printable_x_pos,
  printable_y_pos,
  printable_width,
  printable_height
  
```

See also [GR GET PRINT INFO@](#)

GR/GFX_SET_RIBBON_INFO@

Loads the appropriate ribbon

Format `GR_SET_RIBBON_INFO@(index)`
`GFX_SET_RIBBON_INFO@(gfx, index)`

Method `this.set_ribbon_info@(index)`

Arguments gfx A graphics handle.
 index The color map index

Description Loads a ribbon. In Graphics, use an index value of 0. In the Pixel Editor, set index to the color map ID.

GR/GFX_SET_RPOLY@

Defines the starting angle and number of sides

Format GR_SET_RPOLY@(format gr_regular_poly_info@ info)
 GFX_SET_RPOLY@(gfx, format gr_regular_poly_info@ info)

Method [this.set_rpoly@](#)(format gr_regular_poly_info@ info)

Arguments gfx A graphics handle.
 info A format containing the data being set.

Description Defines the characteristics of a regular polygon using a gr_regular_poly_info@ format. The definition of this format is as follows:

```
format gr_regular_poly_info@
      start_angle,
      n_sides
```

See also [GR_GET_RPOLY@](#)

GR/GFX_SET_RRECT@

Sets the data for a rounded rectangle

Format GR_SET_RRECT@(format gr_rect_info@ info)
 GFX_SET_RRECT@(gfx, format gr_rect_info@ info)

Method [this.set_rrect@](#)(format gr_rect_info@ info)

Arguments gfx A graphics handle.
 info The definition of the rounded rectangle.

Description Sets the scale and radius for a rounded rectangle. The definition of gr_rect_info@ is as follows:

format gr_rect_info@
scale_corners,
x_corner_radius,
y_corner_radius

See also [GR_GET_RRECT@](#)

GR_SET_SLIDE_BG@

Establishes the background of the current slide

Format GR_SET_SLIDE_BG@(format gr_background@ infoArray)
GFX_SET_SLIDE_BG@(gfx, format gr_background@ infoArray)

Method [this.set_slide_bg@](#)(format gr_background@ infoArray)

Arguments gfx a Graphics handle, as returned by CREATE_GRAPHIC@
infoArray an array of type gr_background@

Description Sets the display characteristics of the background of the current slide. You can set the fill type, the foreground color and the background color.

format gr_background@

format gr_fill_attr_type@

hide ' a boolean indicating whether or not the
' background of the slide is hidden. If hide is -1,
' the background is hidden.

format gr_fill_attr_type@

type, ' The name of fill used for the background. This
' name is the string "built-in " and the number of
' the fill. The fills are available from the Slide
' Background option on the Utilities menu in
' Applixware Presents.
fg_color, ' a string: the name of a color in the colormap
bg_color, ' a string: the name of a color in the colormap
angle, ' Not used by GR_SET_SLIDE_BG@
offset ' Not used by GR_SET_SLIDE_BG@

See also [GR_GET_SLIDE_BG@](#)

GR_SET_SLIDE_COLOR_SCHEME@

Establishes the color scheme of the current slide

Format GR_SET_SLIDE_COLOR_SCHEME@(format gr_color_scheme@ array)
GFX_SET_SLIDE_COLOR_SCHEME@(gfx, format gr_color_scheme@ array)

Method [this.set_slide_color_scheme@](#)(format gr_color_scheme@ array)

Arguments gfx A graphics handle, as returned by CREATE_GRAPHIC@
array An array format gr_color_scheme@

Description Establishes a color scheme for the current slide. The array argument is a gr_color_scheme@ format containing information about the color scheme of the current slide. The gr_color_scheme@ format contains the following fields:

format gr_color_scheme@

title_fg,	' Foreground color for title text in each slide
title_bg,	' Background color for title text in each slide
text_fg,	' Foreground color for regular text
text_bg,	' Background color for regular text
line_fg,	' Foreground color for lines
line_bg,	' Background color for lines
fill_fg,	' Foreground color for fills
fill_bg,	' Background color for fills
shadow_color,	' Color for shadows
accent_color`	' An array of 15 accent colors used by charts and other objects

Color schemes are established for each presentation through the Utilities ® Presentations ® Color Scheme dialog in Presents.

See also [GR_GET_SLIDE_COLOR_SCHEME@](#),
[Utilities --> Presentations --> Color Scheme](#)

GR_SET_SLIDE_LAYOUT@

Establishes the layout of a slide

Format GR_SET_SLIDE_LAYOUT@(format gr_layout@ layout)
GFX_SET_SLIDE_LAYOUT@(gfx, format gr_layout@ layout)

Method [this.set_slide_layout@](#)(format gr_layout@ layout)

Arguments gfx A graphics handle, as returned by CREATE_GRAPHIC@
layout an array of format gr_layout@

Description Establishes the layout characteristics of a slide in Applixware Presents. The gr_layout@ format contains the following fields:

format gr_layout@

TYPE,	' A string containing the slide type. These are ' as follows:	
' Title Slide	List and Chart	List and ClipArt
' Bulleted List	Chart and List	ClipArt and List
' Chart	List and Object	Object
' 2 Column List	List and 2 objects	Object and list
' Object over List	2 Objects and List	2 Objects over list
' List over Object	4 Objects	Title Only
' Blank	Custom	
OBJECT1,	' Strings containing the Object types	
OBJECT2,	' These strings can be can be Text, Bulleted List	
OBJECT3,	' Clipart, or Object. If there are fewer than four	
OBJECT4	' objects on a slide, some of these variables can ' contain the string "none".	

See also [GR_GET_SLIDE_LAYOUT@](#)

GR_SET_SLIDE_SHOW_OPTIONS@

Establishes the Slide Show
Options of the current presentation

Format GR_SET_SLIDE_SHOW_OPTIONS@(format gr_slide_show_options@ options)
GFX_SET_SLIDE_SHOW_OPTIONS@(gfx, format gr_slide_show_options@ options)

Method [this.set_slide_show_options@](#)(format gr_slide_show_options@ options)

Arguments gfx A graphics handle, as returned by CREATE_GRAPHIC@
options an array of format gr_layout@

Description Established the slide show options for the current document in Applixware Presents.
The gr_slide_show_options@ format contains the following fields:

format gr_slide_show_options@

full_screen, '-1 (full screen) or 0 (not full screen)
all_pages, '-1 means use all pages in the presentation
 ' for the slide show. 0 indicates use pages
 ' between page1 and pageN.
page1, ' First page of the slide show
pageN, ' Last Page of the slide show
at_end ' a string = "restart", "exit", or "wait" (default). This
 ` establishes what to do when all the indicated
 ` slides have been displayed.

See also [GR_GET_SLIDE_SHOW_OPTIONS@](#)

GR_SET_SLIDE_TRANSITION@

Returns an array of Slide transitions for the current presentation

Format GR_SET_SLIDE_TRANSITION@(page)
GFX_SET_SLIDE_TRANSITION@(gfx, page)

Method [this.set_slide_transition@](#)(page)

Arguments gfx A graphics handle as returned by CREATE_GRAPHIC@

page a number indicating the page whose transitions you want returned. This is a zero-based number. Therefore, the first slide in your presentation is slide 0.

Description Establishes the type of transitions from one slide to the next to use in an Applixware Presents slide show. The `gr_slide_show@` format contains the following fields:

format `gr_slide_show@`

transition, ' a string containing The type of transition
' for the page, and the direction. The
' possible strings are as follows:

none fade explode uncover
implode meltdown spread in wipe
spread out blinds split checkers
slide

delay, ' The number of seconds to delay before
' advancing to the next slide. -1 indicates
' that you want to advance on mouse click.

enable_builds, ' Reserved for future use
dim_inactive_bullets, ' Reserved for future use
dim_color, ' Reserved for future use
build_transition ' Reserved for future use

See also [**GR_GET_SLIDE_TRANSITION@**](#)

`GR/GFX_SET_SHADOW@`

Sets shadow attributes

Format `GR_SET_SHADOW@`(format `gr_shadow_attr_type@` info[, color, x, y])
`GFX_SET_SHADOW@`(gfx, format `gr_shadow_attr_type@` info[, color, x, y])

Method `this.set_shadow@`(format `gr_shadow_attr_type@` info[, color, x, y])

Arguments gfx A graphics handle.
info The shadow attributes being set.

color	The string name of the color.
x	The shadow's horizontal offset.
y	The shadow's vertical offset.

Description Sets the kind of drop shadow that will be drawn (local or background), as well as the shadow's horizontal and vertical offset.

The definition of `gr_shadow_attr_type@` is as follows:

```
format gr_shadow_attr_type@
    type,          ' The kind of drop shadow, as follows:
                  ' 0 none
                  ' 1 background
                  ' 2 local
    color,         ' The string name of the color
    horizontal_offset,
                  ' The shadow's horizontal offset
    vertical_offset
                  ' The shadow's vertical offset
```

Ordinarily, all information is set within the info format. However, for compatibility, you can pass individual arguments. In this case, the info argument becomes the type; that is, it is the kind of drop shadow wanted.

If you are passing data using the structure, the optional arguments are not used.

See also [GR_GET_SHADOW@](#)

GR/GFX_SET_STRUCTURED_COMMENT@

Sets an internal comment

Format `GR_SET_STRUCTURED_COMMENT@(variable, value)`
`GFX_SET_STRUCTURED_COMMENT@(gfx, variable, value)`

Method `this.set_structured_comment@(variable, value)`

Arguments `gfx` A graphics handle.
`variable` The name of the comment.
`value` The value to which the comment is set.

Description Sets the value of a Graphics structured comment. This value is retrieved using [GR_GET_STRUCTURED_COMMENT@](#).

Structured comment variables (and their values) are stored within the Graphics document. These values are retained between Applixware sessions.

GR/GFX_SET_TAG@

Applies a tag to the current selection

Format GR_SET_TAG@(tagString)
GFX_SET_TAG@(gfx, tagString)

Method [this.set_tag@\(tagString\)](#)

Arguments gfx A graphics handle.
tagString The name of a tag.

Description Sets a tag name to the selected object (or objects). After an object is tagged, the object's handle can be retrieved using [GR_QUERY@](#).
See also [GR_GET_TAG@](#).

GR_SET_TEMPLATE_BG@

Establishes the background
of slides in the current presentation

Format GR_SET_TEMPLATE_BG@(format gr_background@ infoArray)

Method [this.set_template_bg@\(format gr_background@ infoArray\)](#)

Arguments infoArray an array of type gr_background@

Description Sets the display characteristics of the background of every slide in the current presentation. The gr_background@ format contain the following fields:

format gr_background@

format gr_fill_attr_type@

hide ' a boolean indicating whether or not the
 ' background of the slide is hidden. If hide is -1,
 ' the background is hidden.

format gr_fill_attr_type@

type, ' The name of fill used for the background. This

' name is the string "built-in " and the number of
 ' the fill. The fills are available from the Slide
 ' Background option on the Utilities menu in
 ' Applixware Presents.
 fg_color, ' a string: the name of a color in the colormap
 bg_color, ' a string: the name of a color in the colormap
 angle, ' Not used by GR_SET_SLIDE_BG@
 offset ' Not used by GR_SET_SLIDE_BG@

See also [GR SET SLIDE BG@](#)

GR/GFX_SET_UNITS@

Sets the unit type for a Graphics grid

Format GR_SET_UNITS@(NAME,DOTS_PER_UNIT,PRECISION)
 GFX_SET_UNITS@(gfx, NAME,DOTS_PER_UNIT,PRECISION)

Arguments gfx A graphics handle.
 Name 'cm' sets the grid to centimeters
 Dots_Per_Unit Number of dots between grid lines.
 Precision Used to convert between dots and units.

Description Use this macro to set the units for the grid feature in graphics. Once the unit is set, you can call GR_SET_GRID_FACTOR@ to display the new grid.

See also [GR SET GRID FACTOR@](#)

GR/GFX_SET_ZOOM@

Sets the zoom of Graphics window in percent

Format GR_SET_ZOOM@(percentZoom)
 GFX_SET_ZOOM@(gfx, percentZoom)

Method [this.set_zoom@\(percentZoom\)](#)

Arguments gfx A graphics handle.

percentZoom The zoom factor.

Description Sets the size at which information is displayed within the Graphics and Pixel Editors. In the Pixel Editor, the minimum zoom is 100%. In the Graphics Editor, the minimum zoom is 10%.

GR/GFX_SHEAR@

Sets the shear

Format GR_SHEAR@()
GFX_SHEAR@(gfx)

Method [this.shear@](#)

Arguments gfx A graphics handle.

Description Stretches an object horizontally or vertically. After invoking this macro, the cursor changes into double arrows when you place it over a selection handle on the selected object.

To shear the object horizontally, place the cursor over a selection handle along the top or bottom of the object so that the cursor changes to a horizontal double arrow, press and hold down the left mouse button and drag the mouse pointer left or right.

To shear the object vertically, place the cursor over a selection handle on one of the midpoints along a side of the object so that the cursor changes to a vertical double arrow, press and hold down the left mouse button and drag the mouse pointer up or down.

GR/GFX_SLOW_MOTION@

Causes an object to move slowly

Format GR_SLOW_MOTION@()
GFX_SLOW_MOTION@(gfx)

Method [this.slow_motion@](#)

Arguments gfx A graphics handle.

Description Any objects moved immediately after GR_SLOW_MOTION@ is executed will move slower than normal. This is useful for precision placement of objects.

GR_SLOW_MOTION@ is only effective for a single movement. If you perform an action other than moving an object after executing GR_SLOW_MOTION@, the slow motion attribute is ended.

GR_SLOW_MOTION@ is called by the Arrange ® Move Slowly menu option.

GR/GFX_SMALLER@

Decreases the size of selected text or objects, or decreases the size of pixels in the pixel edit area

Format GR_SMALLER@()
GFX_SMALLER@(gfx)

Method [this.smaller@](#)

Arguments gfx A graphics handle.

Description If the current Graphics window is not in pixel editing mode, GR_SMALLER@ decreases the size of selected text and objects. Selected text is decreased to the next smaller point size. Selected objects are decreased in size by 10 percent in both the horizontal and vertical direction.

If the current Graphics window is in pixel editing mode, GR_SMALLER@ decreases the size of pixels in the pixel editing area.

GR_SMALLER@ is called by the Transform ® Scale Down menu option.

See also [GR_LARGER@](#)

GR/GFX_SMOOTH@

Smooths selected polyline or free form objects

Format GR_SMOOTH@()
GFX_SMOOTH@(gfx)

Method [this.smooth@](#)

Arguments gfx A graphics handle.

Description Smoothing decreases the jagged appearance of an object. You can smooth an object multiple times. Using this macro only affects selected polyline and free form objects.

This macro is called by the Transform ® Smooth menu option.

GR/GFX_SORT_PARTS@

Sorts custom parts in the current Graphics document

Format GR_SORT_PARTS@()
GFX_SORT_PARTS@(gfx)

Method [this.sort_parts@](#)

Arguments gfx A graphics handle.

Description Places all custom parts in the custom parts list in alphabetical order (A-Z, a-z). If there are no custom parts in the Graphics document, an error is thrown. GR_SORT_PARTS@ is called by the Utilities® Parts® Sort Parts menu option.

GR_SORT_SLIDES@

Turns on Slide Sorter view in Applixware Presents

Format GR_SORT_SLIDES@()

Method [this.sort_slides@\(\)](#)

Description Turns on Slide Sorter view in Applixware Presents. The Slide Sorter allows you to rearrange slides in your presentation.

See also GFX_SORT_SLIDES@

GR/GFX_SUPPRESS_DRAWING@

If set, the drawing is not changed as objects change

Format GR_SUPPRESS_DRAWING@(cacheFlag)
GFX_SUPPRESS_DRAWING@(gfx, cacheFlag)

Method [this.suppress_drawing@\(cacheFlag\)](#)

Arguments gfx A graphics handle.
cacheFlag A Boolean value which if set to TRUE indicates that the display is not changed as graphic objects are added, deleted, or changed.

GR/GFX_STICKY_POINTS@

Sets the sticky points feature

Format GR_STICKY_POINTS@()

Method this.sticky_points@
GFX_STICKY_POINTS@(gfx)

Arguments gfx A graphics handle.

Description Toggles the sticky points feature on and off. If sticky points is not currently on, GR_STICKY_POINTS@ turns on sticky points. When sticky points is on, objects are drawn together when they are placed near each other. If sticky points is currently on, GR_STICKY_POINTS@ turns off sticky points.
GR_STICKY_POINTS@ is called by the Arrange ® Sticky Points menu option.

GR/GFX_TAB_KEY@

Inserts a tab character within text

Format GR_TAB_KEY@()
GFX_TAB_KEY@(gfx)

Method this.tab_key@

Arguments gfx A graphics handle.

GR/GFX_TEXT_?@

Applies the specified typeface to selected text in a Graphics document

Format GR_TEXT_?@()
GFX_TEXT_?@(gfx)

Method this.text_?@

Arguments gfx A graphics handle.

Description GR_TEXT_?@ is called by the Text ® Character Settings option. The specified type size becomes the default. The following typeface macros are supported:

GR_TEXT_AVANT_GARDE@
GF̄X_TEXT̄_AVANT_GARDE@(gfx)
GR_TEXT_BOOKMAN@
GF̄X_TEXT̄_BOOKMAN@(gfx)
GR_TEXT_CHANCERY@
GF̄X_TEXT̄_CHANCERY@(gfx)
GR_TEXT_COURIER@
GF̄X_TEXT̄_COURIER@(gfx)
GR_TEXT_DINGBATS@
GF̄X_TEXT̄_DINGBATS@(gfx)
GR_TEXT_HELVETICA@
GF̄X_TEXT̄_HELVETICA@(gfx)
GR_TEXT_HELV_NARROW@
GF̄X_TEXT̄_HELV_NARRO@(gfx)
GR_TEXT_MONOSPACE@
GF̄X_TEXT̄_MONOSPACE@(gfx)
GR_TEXT_PALATINO@
GF̄X_TEXT̄_PALATINO@(gfx)
GR_TEXT_SANS_SERIF@
GF̄X_TEXT̄_SANS_SERIF@(gfx)
GR_TEXT_SCHOOLBOOK@
GF̄X_TEXT̄_SCHOOLBOOK@(gfx)
GR_TEXT_SERIF@
GF̄X_TEXT̄_SERIF@(gfx)
GR_TEXT_SPECIAL@
GF̄X_TEXT̄_SPECIAL@(gfx)
GR_TEXT_SYMBOL@
GF̄X_TEXT̄_SYMBOL@(gfx)
GR_TEXT_TIMES@
GF̄X_TEXT̄_TIMES@(gfx)

GR/GFX_TEXT_?_POINT@

Sets text size

Format GR_TEXT_?_POINT@()
GF̄X_TEXT̄_?_POINT@(gfx)

Method this.text_?_point@

Arguments gfx A graphics handle.

Description Sets the size of selected text in a Graphics document to 6, 8, 10, 12, 14, 18, 24, or 36 points. GR_TEXT_?_POINT@ is called by the Text ® Size ® ? point menu option. The specified type size becomes the default. The following point size macros are supported:

```
GR_TEXT_6_POINT@
GFX_TEXT_6_POINT@(gfx)
GR_TEXT_8_POINT@
GFX_TEXT_8_POINT@(gfx)
GR_TEXT_10_POINT@
GFX_TEXT_10_POINT@(gfx)
GR_TEXT_12_POINT@
GFX_TEXT_12_POINT@(gfx)
GR_TEXT_14_POINT@
GFX_TEXT_14_POINT@(gfx)
GR_TEXT_18_POINT@
GFX_TEXT_18_POINT@(gfx)
GR_TEXT_24_POINT@
GFX_TEXT_24_POINT@(gfx)
GR_TEXT_36_POINT@
GFX_TEXT_36_POINT@(gfx)
```

GR/GFX_TEXT_ATTRS@

Sets text attributes

Format GR_TEXT_ATTRS@(typeface, size, underline, justify, bold, italic)
GFX_TEXT_ATTRS@(gfx, typeface, size, underline, justify, bold, italic)

Method this.text_attrs@(typeface, size, underline, justify, bold, italic)

Arguments gfx A graphics handle.
typeface One of the following values:

0	Times
1	Bookman
2	New Century Schoolbook
3	Zapf Chancery
4	Palatino
5	Courier

	6	Helvetica
	7	Helvetica Narrow
	8	Avant Garde
	9	Symbol
	10	Zapf Dingbat
size	One of the following:	
	0	6 point
	1	8 point
	2	10 point
	3	12 point
	4	14 point
	5	18 point
	6	24 point
	7	36 point
underline	One of the following	
	0	No underline
	1	Single underline
	2	Double underline
justify	Not used	
bold	Boolean value where TRUE means set to Bold.	
italic	Boolean value where TRUE means set to Italic.	

Description If text is selected, sets the text attributes for this selected text. Otherwise, it sets the attributes that will be applied to text that is typed after the current cursor position.

GR_TEXT_CENTER_ALIGNMENT@

Centers text in a graphics text box

Format GR_TEXT_CENTER_ALIGNMENT@()

Description Centers text in a graphics text box.

See also GFX_TEXT_CENTER_ALIGNMENT@

[**GR_TEXT_JUSTIFY_ALIGNMENT@**](#)

[**GR_TEXT_LEFT_ALIGNMENT@**](#)

[**GR_TEXT_RIGHT_ALIGNMENT@**](#)

GR_TEXT_JUSTIFY_ALIGNMENT@

Right and left justifies text
in a graphics text box

Format GR_TEXT_JUSTIFY_ALIGNMENT@()

Description Makes text in a graphics text box right and left justified.

See also GFX_TEXT_JUSTIFY_ALIGNMENT@

[GR_TEXT_CENTER_ALIGNMENT@](#)

[GR_TEXT_LEFT_ALIGNMENT@](#)

[GR_TEXT_RIGHT_ALIGNMENT@](#)

GR_TEXT_LEFT_ALIGNMENT@

Left justifies text in a graphics text box

Format GR_TEXT_LEFT_ALIGNMENT@()

Description Makes text in a graphics text box left justified.

See also GFX_TEXT_LEFT_ALIGNMENT@

[GR_TEXT_RIGHT_ALIGNMENT@](#)

[GR_TEXT_CENTER_ALIGNMENT@](#)

[GR_TEXT_JUSTIFY_ALIGNMENT@](#)

GR_TEXT_RIGHT_ALIGNMENT@

Right-aligns text in a graphics text box

Format GR_TEXT_RIGHT_ALIGNMENT@

Description Makes text in a graphics text box right justified.

See also GFX_TEXT_RIGHT_ALIGNMENT@

[GR_TEXT_LEFT_ALIGNMENT@](#)

[GR TEXT CENTER ALIGNMENT@](#)

[GR TEXT JUSTIFY ALIGNMENT@](#)

GR/GFX_TOGGLE_SELECT_BY_HANDLE@

Toggles the selection state

Format GR_TOGGLE_SELECT_BY_HANDLE@(handleList)
GFX_TOGGLE_SELECT_BY_HANDLE@(gfx, handleList)

Method [this.toggle_select_by_handle@\(handleList\)](#)

Arguments gfx A graphics handle.
handleList An array of handles.

Description Selects (or deselects) the objects identified by handleList. The handleList is a list of object IDs. Typically, you would obtain these ids using [GR QUERY@](#).

GR/GFX_TOOL_LOCK@

Locks the tool so that it remains the current tool

Format GR_TOOL_LOCK@(tool)
GFX_TOOL_LOCK@(gfx, tool)

Method [this.tool_lock@\(tool\)](#)

Arguments gfx A graphics handle.
tool One of the following strings:

Drawing Tools:

selection_tool
text_tool
line_tool
polyline_tool
rectangle_tool
ellipse_tool
freeform_tool
mono_digitizer_tool
rounded_rectangle_tool
polygon_tool

curve_tool
color_digitizer_tool
zoom_tool
formatted_text_tool

Pixel Editor Tools:

paint_tool
wash_tool
pen_tool
blend_tool
erase_tool
color_erase_tool
line_tool
rectangle_tool
ellipse_tool
flood_tool
filter_tool
selection_tool

Description Locks the tool so that it remains the current tool. This means, for example, that if you need to draw four circles, you will not need to select the ellipse tool four times.

GR/GFX_TOOL_PICK@

Selects a drawing tool

Format GR_TOOL_PICK@(tool)
GFX_TOOL_PICK@(gfx, tool)

Method [this.tool_pick@](#)(tool)

Arguments gfx A graphics handle.
 tool One of the following strings:

Drawing Tools:

selection_tool
text_tool
line_tool
polyline_tool
rectangle_tool
ellipse_tool
freeform_tool
mono_digitizer_tool

rounded_rectangle_tool
polygon_tool
curve_tool
color_digitizer_tool
zoom_tool
formatted_text_tool

Pixel Editor Tools:

paint_tool
wash_tool
pen_tool
blend_tool
erase_tool
color_erase_tool
line_tool
rectangle_tool
ellipse_tool
flood_tool
filter_tool
selection_tool

Description For tools 0 through 13, the tool selection will revert back to the default tool after the selected tool is used unless the [Single Click Tool Lock](#) option is chosen in the Graphics Preferences dialog box. To lock the tool for repeated use regardless of your Graphics Preferences settings, use [GR_LOCK_TOOL@](#).

Example

GR/GFX_TYPE@

Types text

Format GR_TYPE@(string)
GFX_TYPE@(gfx, string)

Method [this.type@](#)(string)

Arguments gfx A graphics handle.
 string The characters being displayed in the text area.

Description Types text within a Text or a Formatted Text area.

GR/GFX_UNCOMBINE@

Separates combined objects

Format GR_UNCOMBINE@()
GFX_UNCOMBINE@(gfx)

Method [this.uncombine@](#)

Arguments gfx A graphics handle.

Description Separates combined objects. When you use GR_UNCOMBINE@ the objects do not return to their original state. Instead, all objects retain the attributes of the combined object.

See also [GR_COMBINE@](#)

GR/GFX_UNDERLINE@

Makes selected text underlined in a Graphics document

Format GR_UNDERLINE@()
GFX_UNDERLINE@(gfx)

Method [this.underline@](#)

Arguments gfx A graphics handle.

Description Makes selected text underlined in a Graphics document. If no text is selected, underline becomes a default attribute for the document. GR_UNDERLINE@ is called by the Text ® Underline menu option.

See also [GR_BOLD@](#)
[GR_ITALICS@](#)

GR/GFX_UNDO@

Reverses the action last performed

Format GR_UNDO@()

GFX_UNDO@(gfx)

[Method](#) this.undo@

Arguments gfx A graphics handle.

Description Reverses the action that was performed in the Graphics window immediately prior to its execution. GR_UNDO@ is called by the Edit ® Undo menu option.

GR/GFX_UNGROUP@

Ungroups selected objects

Format GR_UNGROUP@()
GFX_UNGROUP@(gfx)

[Method](#) this.ungroup@

Arguments gfx A graphics handle.

Description Removes one level of grouping. You can use GR_UNGROUP@ multiple times to continually separate object groups until no more objects are grouped.
GR_UNGROUP@ is called by the Transform ® Flip Horizontal menu option.

See also [GR_GROUP@](#)

GR/GFX_UNLOCK_ALL@

Unlocks all locked objects

Format GR_UNLOCK_ALL@()
GFX_UNLOCK_ALL@(gfx)

[Method](#) this.unlock_all@

Arguments gfx A graphics handle.

Description Unlocks all objects, making the objects selectable and editable.

See also [GR_LOCK_SELECTED@](#)
[GR_UNLOCK_SELECTED@](#)

GR/GFX_UNLOCK_LAYER@

Unlocks a layer

Format GR_UNLOCK_LAYER@(layerName)
GFX_UNLOCK_LAYER@(gfx, layerName)

Method [this.unlock_layer@](#)(layerName)

Arguments gfx A graphics handle.
layerName The name of the layer to unlock.

Description Unlocks the passed layerName, making objects in layerName selectable and editable.

See also [GR_LOCK_LAYER@](#)

GR/GFX_UNLOCK_SELECTED@

Unlocks the selected objects

Format GR_UNLOCK_SELECTED@(
GFX_UNLOCK_SELECTED@(gfx)

Method [this.unlock_selected@](#)

Arguments gfx A graphics handle.

Description Unlocks selected objects, making the objects selectable and editable.

See also [GR_LOCK_SELECTED@](#)
[GR_UNLOCK_ALL@](#)

GR/GFX_UP_ARROW_KEY@

Moves up

Format GR_UP_ARROW_KEY@(
GFX_UP_ARROW_KEY@(gfx)

Method [this.up_arrow_key@](#)

Arguments gfx A graphics handle.

Description If the cursor is within a formatted text area, this macro moves the cursor up one line.
If an object or a group is selected, this macro moves the object up by 10 mils.

See also [GR_DOWN_ARROW_KEY@](#)

GR_VERTICAL_GUIDE@

Limits object movement to vertical

Format GR_VERTICAL_GUIDE@()

Method [this.vertical_guide@](#)

Description Limits the direction an object can move to vertical. Any objects moved immediately after GR_VERTICAL_GUIDE@ is executed can only be moved vertically. GR_VERTICAL_GUIDE@ is only effective for a single movement. If you perform an action other than moving an object after executing GR_VERTICAL_GUIDE@, the vertical movement limitation is ended.

See also [GR_HORIZ_GUIDE@](#)

GR_VIEW_EXPRESSLINE@

Toggles the display of the *ExpressLine*

Format GR_VIEW_EXPRESSLINE@()

Method [this.view_expressline@](#)

GR/GFX_WORD_BACK_KEY@

Moves the cursor backwards one word

Format GR_WORD_BACK_KEY@(
GFX_WORD_BACK_KEY@(gfx)

Method [this.word_back_key@](#)

Arguments gfx A graphics handle.

Description If the cursor is positioned within text, this macro moves the cursor backwards one word.

See also [GR_WORD_FORWARD_KEY@](#)

GR/GFX_WORD_FORWARD_KEY@

Moves the cursor forward one word

Format GR_WORD_FORWARD_KEY@()
GFX_WORD_FORWARD_KEY@(gfx)

Method [this.word_forward_key@](#)

Arguments gfx A graphics handle.

Description If the cursor is positioned within text, this macro moves the cursor forward one word.

See also [GR_WORD_BACK_KEY@](#)

GR/GFX_X_MIRROR@

Flips selected objects horizontally

Format GR_X_MIRROR@()
GFX_X_MIRROR@(gfx)

Method [this.x_mirror@](#)

Arguments gfx A graphics handle.

Description Flips selected objects in the current Graphics document horizontally. If multiple objects are selected, the objects are flipped as a group.

GR_X_MIRROR@ is called by the Transform ® Flip Horizontal menu option.

See also [GR_Y_MIRROR@](#)

GR/GFX_Y_MIRROR@

Flips selected objects vertically

Format GR_Y_MIRROR@()

GFX_Y_MIRROR@(gfx)

Method [this.y_mirror@](#)

Arguments gfx A graphics handle.

Description Flips selected objects in the current Graphics document vertically. If multiple objects are selected, the objects are flipped as a group.

GR_Y_MIRROR@ is called by the Transform ® Flip Vertical menu option.

See also [GR X MIRROR@](#)

END_STROKE@

Indicates the ending position of a mouse pointer drag

Format END_STROKE@ (xpos, ypos[, button, shift, control, hvStatus])

Arguments

xpos	The pixel position, relative to the drawing area, of the mouse pointer on the x-axis.
ypos	The pixel position, relative to the drawing area, of the mouse pointer on the y-axis.
button	A number indicating the mouse button pressed during the pointer drag. 0 left button 1 center button 2 right button.
shift	Indicates whether the SHIFT key is pressed while dragging. 1 SHIFT is pressed 0 SHIFT is not pressed.
control	Indicates whether the CONTROL key is pressed while dragging. 1 CONTROL is pressed 0 CONTROL is not pressed.
hvStatus	Only applicable to the line tool, this argument indicates whether a strictly horizontal or strictly vertical line was drawn. 1 A vertical line 2 A horizontal line

This argument is only relevant to keystroke recordings involving `END_STROKE@`. If you use `END_STROKE@` to specify the ending position of a stroke in Graphics, do not use the `hvStatus` parameter.

Description Indicates the mouse pointer position at the end of a drag in a drawing area and specifies whether `SHIFT` or `CONTROL` is pressed during the drag.

`END_STROKE@`, though still supported, has been replaced with **`STROKE_END@`**, which uses fewer arguments.

FILTER ?_TO_GR@

Imports a file into Graphics

Format `FILTER ?_TO_GR@(infile, outfile)`

Arguments

<code>infile</code>	The full pathname of the non-Appixware graphics file.
<code>outfile</code>	The full pathname of the Appixware Graphics file, including the <code>.ag</code> extension.

Description All `FILTER ?_TO_GR@` macros convert a non-Appixware graphics file into an Appixware Graphics file. The bold macros cannot be used without first obtaining Appixware Filter Pack(s) licenses from Appixware.

One macro exists for each of the following formats:

<code>FILTER_CGM_TO_GR@</code>	CGM file
<code>FILTER_DXF_TO_GR@</code>	DXF file
<code>FILTER_EPSI_TO_GR@</code>	Encapsulated PostScript
<code>FILTER_FAX_TO_GR@</code>	Group 3 FAX file
<code>FILTER_GEM_TO_GR@</code>	GEM file
<code>FILTER_GIF_TO_GR@</code>	GIF file
<code>FILTER_GP4_TO_GR@</code>	Group 4 FAX file
<code>FILTER_GP4_TO_GR_PEND@</code>	Group 4 FAX file (performed in background)
<code>FILTER_HPGL_TO_GR@</code>	HPGL file
<code>FILTER_ILBM_TO_GR@</code>	ILBM file
<code>FILTER_IM_TO_GR@</code>	Appixware Bitmap file
<code>FILTER_JPEG_TO_GR@</code>	JPEG file
<code>FILTER_IRIS_TO_GR@</code>	Silicon Graphics IRIS file
<code>FILTER_MPNT_TO_GR@</code>	MacPaint file
<code>FILTER_MSWB_TO_GR@</code>	MS Windows Bitmap file
<code>FILTER_PBM_TO_GR@</code>	PBM file

FILTER_PCX_TO_GR@ PCX file
 FILTER_PGM_TO_GR@ X11 Portable graymap file
FILTER_PICT2_TO_GR@ PICT2 file
FILTER_PICT_TO_GR@ PICT file
 FILTER_PPM_TO_GR@ X11 Portable Pixmap file
FILTER_PPT_TO_GR@ PowerPoint file
 FILTER_RAW_TO_GR@ Raw file
 FILTER_RS_TO_GR@ Sun Raster file
 FILTER_TGA_TO_GR@ TGA file
 FILTER_TIFF_TO_GR@ TIFF file
FILTER_URL_TO_GR@ Listed separately
 FILTER_WMF_TO_GR@ Windows metafile
 FILTER_WPG_TO_GR@ WordPerfect Graphics file
 FILTER_XBM_TO_GR@ X Windows Bitmap file
 FILTER_XPM_TO_GR@ X Windows Pixmap
 FILTER_XWD_TO_GR@ X Windows Dump

FILTER_GR_TO_?@

Exports an Applixware Graphics file

Format FILTER_GR_TO_?@(infile, outfile[, depth[, gray[, resolution]]])

Arguments

infile	Full pathname of the Applixware Graphics file, including the .ag extension.
outfile	Full pathname of the resulting non-Applixware graphics file.
depth	The number of bits used to express color. The default is 8 bits. Other valid values are 1 (black and white) and 24 (often called true color).
gray	A Boolean value where TRUE means display the file's objects using an 8-bit gray-scale.
resolution	The bitmap resolution in Dots Per Inch (DPI). The default value is 75 DPI.

Description All FILTER_GR_TO_?@ macros convert an Applixware Graphics file into a format supported by another product. Each macro in the following list performs one conversion. The macros displayed in bold cannot be used without first obtaining Applixware Filter Pack(s) licenses from Applixware.

FILTER_GR_TO_CGM@(infile, outfile) Computer Graphics Metafile
 FILTER_GR_TO_EPS@ (infile, outfile, depth, gray, resolution) Encapsulated post-script
 FILTER_GR_TO_FAXI@ (infile, outfile, NULL, NULL, resolution) CCITT Group 3 FAX (lsb) format

FILTER_GR_TO_FAXM@(infile, outfile, NULL, NULL, resolution)
 CCITT Group 3 FAX (msb) format
 FILTER_GR_TO_GIF@(infile, outfile, depth, gray, resolution)
 GIF 87 File (This macro is obsolete. Please use
 FILTER_GR_TO_GIF87 for all new programming.)
 FILTER_GR_TO_GIF87@(infile, outfile, depth, gray, resolution)
 GIF 87 File
 FILTER_GR_TO_GIF89@(infile, outfile, depth, gray, resolution)
 Gif 89 File
 FILTER_GR_TO_GR3@(infile, outfile)
 Applixware Release 3.0 format
 FILTER_GR_TO_GR400@(infile, outfile)
 Applixware Release 4.0 format
 FILTER_GR_TO_HPGL@(infile, outfile)
 Hewlett Packard Graphics Language
 FILTER_GR_TO_IM@(infile, outfile, depth, gray, resolution)
 Applixware Bitmap
 FILTER_GR_TO_IRIS@(infile, outfile, depth, gray, resolution)
 Silicon Graphics Iris file format
 FILTER_GR_TO_JPEG@(infile, outfile, depth, gray, resolution)
 JPEG file format
 FILTER_GR_TO_JPEG50@(infile, outfile, depth, gray, resolution)
 JPEG 50 file format
 FILTER_GR_TO_JPEG75@(infile, outfile, depth, gray, resolution)
 JPEG 75 file format
 FILTER_GR_TO_JPEG90@(infile, outfile, depth, gray, resolution)
 JPEG 90 file format
 FILTER_GR_TO_MSWINBM@(infile, outfile)
 Microsoft Window Bitmap
 FILTER_GR_TO_PBM@(infile, outfile, NULL, NULL, resolution)
 X11 Portable Bitmap
 FILTER_GR_TO_PGM@(infile, outfile, NULL, gray, resolution)
 X11 Portable Greymap
 FILTER_GR_TO_PPM@(infile, outfile, depth, gray, resolution)
 X11 Portable Pixmap
 FILTER_GR_TO_RS@(infile, outfile, depth, gray, resolution)
 Sun Raster
 FILTER_GR_TO_SLIDESHOW@(infile, outfile)
 Applixware Presents Slideshow
 FILTER_GR_TO_TIFFI@(infile, outfile, depth, gray, resolution)
 TIFF for INTEL
 FILTER_GR_TO_TIFFM@(infile, outfile, depth, gray, resolution)
 TIFF for Motorola

FILTER_GR_TO_WMF@(infile, outfile)
MS Windows Metafile
FILTER_GR_TO_XBM@(infile, outfile, NULL, NULL, resolution)
X Windows Bitmap
FILTER_GR_TO_XWD@(infile, outfile, depth, gray, resolution)
X Window Dump

See also [FILTER ? TO GR@](#)
[GE IMPORT ?@](#)
[GR IMPORT HPGL@](#)
[GE PASTE ?@](#)
[GE PASTE TIF@](#)

For more information about graphic filters, see the *Graphics* manual.

FILTER_GR_TO_AP@

Renames a graphics file as a slideshow file

Format FILTER_GR_TO_AP@(infile)

Arguments infile The absolute pathname of an Appixware Graphics file.

Description Renames an Appixware Graphics file, which has an '.ag' extension, to an Appixware Presents slideshow file, which has a '.ap' extension. For example, test.ag would be renamed test.ap.

FILTER_GRFIL TO_RS@

Converts an Appixware Graphics file into a Sun Raster file

Format FILTER_GRFIL TO_RS@(infile, outfile, depth, gray, resolution)

Arguments infile Full pathname of the Appixware Graphics file, including the .ag extension.
outfile Full pathname of the resulting Sun Raster file.
depth The number of bits used to express color. The default is 8 bits. Other valid values are 1 (black and white) and 24 (often called true color).
1.
gray A Boolean value where TRUE means display the file's objects using an 8-bit gray-scale.

resolution The bitmap resolution in Dots Per Inch (DPI). The default value is 75 DPI.

FILTER_URL_TO_GR@

Filters graphic URL into a graphics file

Format FILTER_URL_TO_GR@(URLin, grOut[, baseURL])

Arguments URLin The URL of the graphic object.
grOut The name of the file into which the graphic object will be written.
baseURL If URLin is a relative URL, this portion of the URL will be merged by [WWW_MERGE_URL@](#) with URLin to form the graphic object's URL.

Description Fetches a graphics file from the Web and transforms it into an Applixware Graphics file. Not all formats used in Web documents can be filtered. If a format cannot be filtered, an error is thrown.

See also [FILTER_HTML_TO_WP@](#)
[FILTER_URL_TO_WP@](#)
[WP_IMPORT_HTML@](#)
[WP_IMPORT_URL@](#)
[WWW_FETCH_URL@](#)
[WWW_MERGE_URL@](#)

FILTER_?_TO_IM@

Imports a file, converting it to Bitmap format

Format FILTER_?_TO_IM@(infile, outfile[, depth[, gray[, resolution[, width,[height]]]]])

Arguments infile The file being imported and converted.
outfile The file to which the converted information is written.
depth The number of bits used to express color. The default is 8 bits. Other valid values are 1 (black and white) and 24 (often called true color).
gray A Boolean value where TRUE means display the file's objects using an 8-bit gray-scale.
resolution The bitmap resolution in Dots Per Inch (DPI). The default value is 75 DPI.

Description All **FILTER_?_TO_IM@** macros convert a non-Appixware graphics file into an Appixware Bitmap file. The bold macros cannot be used without first obtaining a Graphics Pack license from Appixware.

Most of the following macros take the same argument. However, **FILTER_RAW_TO_IM@** differs.

One macro exists for each of the following formats:

FILTER_CGM_TO_IM@(infile, outfile, depth, gray, resolution)
CGM file

FILTER_DXF_TO_IM@(infile, outfile, depth, gray, resolution)
DXF file

FILTER_EPSI_TO_IM@(infile, outfile, depth, gray, resolution) Encapsulated Post-Script

FILTER_FAX_TO_IM@(infile, outfile, depth, gray, resolution)
Group 3 FAX file

FILTER_GEM_TO_IM@(infile, outfile, depth, gray, resolution)
GEM file

FILTER_GIF_TO_IM@(infile, outfile, depth, gray, resolution)
GIF file

FILTER_GP4_TO_IM@(infile, outfile, depth, gray, resolution)
Group 4 FAX file

FILTER_HPGL_TO_IM@(infile, outfile, depth, gray, resolution)
HPGL file

FILTER_ILBM_TO_IM@(infile, outfile, depth, gray, resolution)
ILBM file

FILTER_IRIS_TO_IM@(infile, outfile, depth, gray, resolution)
Silicon Graphics IRIS file

FILTER_JPEG_TO_IM@(infile, outfile, depth, gray, resolution, width, height) JPEG file

FILTER_MPNT_TO_IM@(infile, outfile, depth, gray, resolution)
MacPaint file

FILTER_MSWB_TO_IM@(infile, outfile, depth, gray, resolution)
MS Windows Bitmap file

FILTER_PBM_TO_IM@(infile, outfile, depth, gray, resolution)
PBM file

FILTER_PCX_TO_IM@(infile, outfile, depth, gray, resolution)
PCX file

FILTER_PGM_TO_IM@(infile, outfile, depth, gray, resolution)
X11 Portable graymap file

FILTER_PICT2_TO_IM@(infile, outfile, depth, gray, resolution)
PICT2 file

FILTER_PICT_TO_IM@(infile, outfile, depth, gray, resolution)
PICT file

FILTER_PPM_TO_IM@(infile, outfile, depth, gray, resolution)
X11 Portable Pixmap file

FILTER_PPT_TO_IM@(infile, outfile, depth, gray, resolution)
Powerpoint file

FILTER_RAW_TO_IM@(infile, outfile, depth, gray, resolution, width, height)
Raw file

FILTER_RS_TO_IM@(infile, outfile, depth, gray, resolution)
Sun Raster file

FILTER_TGA_TO_IM@(infile, outfile, depth, gray, resolution)
TGA file

FILTER_TIFF_TO_IM@(infile, outfile, depth, gray, resolution)
TIFF file

FILTER_WMF_TO_IM@(infile, outfile, depth, gray, resolution)
Windows metafile

FILTER_WPG_TO_IM@(infile, outfile, depth, gray, resolution)
WordPerfect Graphics file

FILTER_XBM_TO_IM@(infile, outfile, depth, gray, resolution)
X Windows Bitmap file

FILTER_XPM_TO_IM@(infile, outfile, depth, gray, resolution)
X Windows Pixmap

FILTER_XWD_TO_IM@(infile, outfile, depth, gray, resolution)
X Windows Dump

SET_CHART_AS_MACRO_PARENT_TASK@

Sets a chart as the macro parent task

Format taskID = SET_CHART_AS_MACRO_PARENT_TASK@ (chartName)

Arguments chartName The string name of a chart

Description **This macro is obsolete! Do not use this macro in new ELF programs!**

Sets the chart's task as being the parent task. That is, this macro transfers control from the current macro parent task to the chart task.

For more information, see [MACRO_PARENT_TASK@](#) and [SET_MACRO_PARENT_TASK@](#).

START_STROKE@

Indicates the starting position of a mouse pointer drag

Format START_STROKE@ (xpos, ypos[, button, shift, control])

Arguments

xpos	The pixel position, relative to the drawing area, of the mouse pointer on the x-axis.
ypos	The pixel position, relative to the drawing area, of the mouse pointer on the y-axis.
button	A number indicating the mouse button pressed during the pointer drag. 0 left button 1 center button 2 right button
shift	Indicates whether the SHIFT button is pressed while dragging. 1 SHIFT is pressed 0 SHIFT is not pressed
control	Indicates whether the CONTROL button is pressed while dragging. 1 CONTROL is pressed 0 CONTROL is not pressed

Description Indicates the mouse pointer position at the beginning of a drag in a Graphics drawing area and specifies whether SHIFT or CONTROL is pressed during the drag.

See also [STROKE_END@](#)

[STROKE_POINT@](#)

[STROKE_START@](#)

STROKE_END@

Indicates the ending position of a mouse pointer drag

Format STROKE_END@ (xpos, ypos[,hvStatus])

Arguments

xpos	The pixel position, relative to the drawing area, of the mouse pointer on the x-axis.
ypos	The pixel position, relative to the drawing area, of the mouse pointer on the y-axis.
hvStatus	Only applicable to the line tool, this argument indicates whether a strictly horizontal or strictly vertical line was drawn. A hvStatus value of 1 indicates that a strictly vertical line was drawn. A hvStatus value of 2 indicates that a strictly horizontal line was drawn. This argument is only relevant to keystroke recordings involving STROKE_END@.

If STROKE_END@ is used as part of a macro to specify the ending position of a stroke in Graphics, the hvStatus argument is not required.

Description Indicates the mouse pointer position at the end of a drag in a Graphics drawing area.

Example

See also [START_STROKE@](#)
[STROKE_POINT@](#)
[STROKE_START@](#)

STROKE_POINT@

Indicates mouse pointer position during a mouse drag

Format STROKE_POINT@(xpos, ypos)

Arguments

xpos	The pixel position, relative to the drawing area, of the mouse pointer on the x-axis.
ypos	The pixel position, relative to the drawing area, of the mouse pointer on the y-axis.

Description Indicates the mouse pointer position during a drag in a Graphics drawing area. It can be used to draw a free form object in a Graphics window by indicating the mouse pointer position as it is dragged.

Example

See also [STROKE_END@](#)
[STROKE_POINT_BOUNDARY@](#)
[STROKE_START@](#)
[STROKE_START_EXTEND_SELECT@](#)
[STROKE_START_MULTI_SELECT@](#)

STROKE_POINT_BOUNDARY@

Indicates a stroke boundary occurred

Format STROKE_POINT_BOUNDARY@(xpos, ypos)

Arguments xpos The point's x-coordinate.
 ypos The point's y-coordinate.

Description Indicates that a stroke reached a window boundary. (A stroke is a movement of the mouse with a mouse button pressed.) In some cases, you will ignore this event as it means that you are not within the window's boundary. In some cases, however, you may want to scroll your window or perform some other action so to allow the stroking to continue.

See also [STROKE_END@](#)
[STROKE_POINT@](#)
[STROKE_START@](#)
[STROKE_START_EXTEND_SELECT@](#)
[STROKE_START_MULTI_SELECT@](#)

STROKE_START@

Sets the point that begins a stroke

Format STROKE_START@(xpos, ypos)

Arguments xpos The point's x-coordinate.
 ypos The point's y-coordinate.

Description Indicates the point at which a stroke movement begins. (A stroke is a movement of the mouse with a mouse button pressed.)

Example

See also [STROKE_END@](#)
[STROKE_POINT@](#)
[STROKE_POINT_BOUNDARY@](#)
[STROKE_START_EXTEND_SELECT@](#)
[STROKE_START_MULTI_SELECT@](#)

STROKE_START_EXTEND_SELECT@

Starts an extend stroke

Format STROKE_START_EXTEND_SELECT@(xpos, ypos)

Arguments xpos The point's x-coordinate.
 ypos The point's y-coordinate.

Description Indicates the point at which a extended-selection stroke movement begins. (An extended selection stroke is a movement of the mouse with a mouse button pressed and with the keyboard's shift key pressed.)

See also [STROKE_END@](#)
[STROKE_POINT@](#)
[STROKE_POINT_BOUNDARY@](#)
[STROKE_START@](#)
[STROKE_START_MULTI_SELECT@](#)

STROKE_START_MULTI_SELECT@

Starts a multi-select stroke

Format STROKE_START_MULTI_SELECT@(xpos, ypos)

Arguments xpos The point's x-coordinate.
 ypos The point's y-coordinate.

Description Indicates the point at which a multi-selection stroke movement begins. (A multi selection stroke is a movement of the mouse with a mouse button pressed and with the keyboard's control key pressed.)

See also [STROKE_END@](#)
[STROKE_POINT@](#)
[STROKE_POINT_BOUNDARY@](#)
[STROKE_START@](#)
[STROKE_START_EXTEND_SELECT@](#)

graphics .am

```
.....  
.....  
' Basic graphic datatypes  
,
```

```
.....  
format gr_area@ x, y, width, height  
format gr_point@ x, y  
format gr_curve_point@  
    x, y,  
    format gr_point@ ctl1,  
    format gr_point@ ctl2  
format gr_path@  
    format arrayof gr_curve_point@ pt
```

```
format gr_color@  
    c, m, y, k,  
    see_thru,  
    ink_type,  
    name
```

```
.....  
.....  
' Attribute formats
```



```

'
.....
format gr_fill_attr_type@ ' for part_fill, line_fill, and text_fill
    type, ' string: built-in <#>, <filename>, linear gradient
    fg_color, ' string: name of color in colormap
    bg_color, ' string: name of color in colormap
    angle, ' int: in degrees
    offset ' int: in mils (1000 mils = 1 inch)

format gr_shadow_attr_type@
    type, ' string: none, background drop shadow, local drop shadow
    color, ' string: name of color in colormap
    horizontal_offset, ' int: in mils (1000 mils = 1 inch)
    vertical_offset ' int: in mils (1000 mils = 1 inch)

format gr_line_style_type@
    style,
    weight,
    first_symbol,
    join_symbol,
    final_symbol

format gr_text_field_attr_type@
    horizontal_alignment,
    vertical_alignment,
    line_space,
    horizontal_scale,
    vertical_scale,
    shear,
    angle,
    left_margin,
    right_margin,
    top_margin,
    bottom_margin

format gr_text_para_attr_type@
    left_indent1,
    left_indent2,
    right_indent,
    bullet,
    bullet_scale,
    bullet_color,
    bullet_face,
    bullet_character_code,

```

space_above,
space_below,
line_space

format gr_text_char_attr_type@

face,
size,
ruling,
 strike_thru,
bold,
italic,
horizontal_scale,
shear,
angle,
horizontal_offset,
vertical_offset,
horizontal_sub_sup,
vertical_sub_sup

format gr_attribute@

format gr_fill_attr_type@ backfill,
format gr_fill_attr_type@ linefill,
format gr_shadow_attr_type@ shadow,
format gr_line_style_type@ line,
format gr_text_field_attr_type@ field,
format gr_text_char_attr_type@ char,
format gr_text_para_attr_type@ paragraph

format gr_poly_text@

format gr_attribute@ attr,
str

format gr_thing@

type,
format gr_point@ position,
format gr_area@ ink_boundary,
format gr_attribute@ def_attr,
format arrayof gr_path@ path,
format arrayof gr_poly_text@ txt,
id

format gradient_info@

name,
type,

angle,
xoffset,
yoffset

format gr_audio@
enable,
volume,
path

.....
.....
' Some type specific info
,

.....
format gr_rect_info@
scale_corners,
x_corner_radius,
y_corner_radius

format gr_ellipse_info@
angle1,
angle2

format gr_regular_poly_info@
start_angle,
n_sides

.....
.....
' Callback info
,

.....
format gr_callback_info@
macro_name,
trigger_on_drag,
trigger_on_single_click,
trigger_on_double_click,
argument_list

format gr_callback_data@
type, "drag", "single click", "double click"
object_handle, ' ID of thing that callback is attached to
format gr_point@ down, ' mouse down position (current GR coords)
format gr_point@ up, ' mouse up position (current GR coords)
argument_list ' list of arguments specified by callback

.....
.....

' Editor state info
,

.....

format gr_guides@
rulers,
grids,
coordinates,
auto_gridding,
sticky_points,
page_breaks,
target_viewmode,
gridsize,
device_setup

.....
.....

' Layer info
,

.....

format gr_layer@
name,
locked,
hidden,
n_parts,
printable,
background

.....
.....

' Extrusion info
,

.....

format gr_extrusion@
projection,
horizontal,
vertical,
depth

.....
.....

' Slide/Page background and color scheme info
,

.....

format gr_color_scheme@

title_fg,
title_bg,
text_fg,
text_bg,
line_fg,
line_bg,
fill_fg,
fill_bg,
shadow_color,
accent_color

format gr_background@

format gr_fill_attr_type@ fill,
hide

format gr_slide_show@

transition,
delay,
enable_builds,
dim_inactive_bullets, dim_color,
build_transition

format gr_slide_show_options@

full_screen,
all_pages,
page1,
pageN,
at_end ' "restart", "exit", or "wait" (default)

format gr_layout@

type,
object1,
object2,
object3,
object4

.....

.....

' startup formats

,

.....

format gp_data@

```
startup_mode,  
media_type,  
title,  
subtitle,  
outline,  
filename
```

```
.....  
.....
```

```
' Mouse info
```

```
.....
```

```
format gr_mouse@  
window_relative,  
button1,  
button2,  
button3,  
button4,  
button5,  
shifted,  
controled,  
altd
```

```
.....  
.....
```

```
' Inset info
```

```
.....
```

```
format gr_inset@  
clear_mode,      ' "none", "widget", "window"  
scale_mode,      ' "clip to fit", "scale to fit", "scale if clips"  
proportional,    ' true or false, used when scaling  
prescale_x,      ' applied regardless of scale mode - 100 is 1X  
prescale_y,      ' applied regardless of scale mode - 100 is 1X  
format gr_area@ src, ' source pos and size in current graphic units  
format gr_area@ dst, ' destination pos and size in current graphic units  
format gr_area@ clip, ' clip pos and size in current graphic units  
display_mode,    ' special display effects  
toggle_wait_cursor, ' should the inset display the wait cursor  
page_number,     ' which page in the graphic to inset  
widget,          ' widget to draw onto  
audio_enable,    ' audio enable  
audio_volume,    ' audio volume  
audio_path       ' audio path name to use
```

```
.....  
.....  
' pre rev3.0 formats  
,  
.....
```

```
format gr_guides_  
    rulers,  
    grids,  
    coordinates,  
    auto_gridding,  
    sticky_points,  
    page_breaks,  
    target_viewmode,  
    gridsize,  
    device_setup
```

```
format gr_line_atts_  
    pattern,  
    style,  
    thickness,  
    arrowheads,  
    arrowstyle
```

```
format gr_text_atts_  
    style,  
    size,  
    bold,  
    italics,  
    underline,  
    justify
```

```
format gr_color_atts_  
    foreground_color,  
    background_color,  
    reverse_mode,  
    transparent_mode
```

```
format gr_color_attrs_  
    foreground_color,  
    background_color,  
    line_foreground_color,  
    line_background_color,  
    shadow_color,
```

transparent_mode

format ge_part_attribute_
fill_fg_color, fill_bg_color, fill_type, fill_id,
shadow, shadow_color, shadow_x, shadow_y,
line_fg_color, line_bg_color, line_type, line_id,
line_width, line_style, line_end1, line_end2, line_symbol,
face, size, weight, slant, ruling, alignment,
layer_id,
user_id