

PGS.MD2

Copyright © 1995 Soft-Logik Publishing Corporation

COLLABORATORS

	<i>TITLE :</i> PGS.MD2		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	PGS.MD2	1
1.1	editbox	1
1.2	editcolumn	2
1.3	editcompound	2
1.4	editdrawing	3
1.5	editellipse	4
1.6	editeps	5
1.7	editgrid	6
1.8	editgroup	6
1.9	editline	7
1.10	editobject	8
1.11	editpalette	9
1.12	editpath	9
1.13	editpicture	10
1.14	editpolygon	11
1.15	edittext	12
1.16	edittextobj	12
1.17	endcommandcapture	13
1.18	exportgraphic	14
1.19	exporttext	14
1.20	filled	15
1.21	findtext	16
1.22	fontpalette	16
1.23	freearexxlist	17
1.24	freearexxrequester	17
1.25	getarexxgadget	18
1.26	getarticle	19
1.27	getarticledesc	20
1.28	getarticleids	20
1.29	getarticlename	21

1.30	getarticleoverset	21
1.31	getarticlewordcount	22
1.32	getbleed	22
1.33	getbox	23
1.34	getbusyrequester	24
1.35	getchapterdesc	24
1.36	getchapternumber	25
1.37	getchapternumbering	25
1.38	getchapters	26
1.39	getchoice	27
1.40	getcolumn	27
1.41	getcolumnguides	28
1.42	getcompound	29
1.43	getcoord	30
1.44	getcursor	31
1.45	getcursormount	32
1.46	getdimensions	33
1.47	getdisplay	33
1.48	getdocumentdesc	34
1.49	getdocuments	34
1.50	getdocumentstatus	35
1.51	getdrawing	35
1.52	getdrawingdisplay	36
1.53	getellipse	37

Chapter 1

PGS.MD2

1.1 editbox

EDITBOX

Purpose: Sets (edits) the coordinates of a selected box.

Syntax: editbox [POSITION left/D top/D right/D bottom/D]
 [NORMAL | ROUND | SCALLOP | BEVEL | INSET]
 [CORNER radius/D | ECORNER radiusx/D radiusy/D]
 [ROTATE angle/A | SKEW slantangle/A twistangle/A | SLANT angle/A |
 TWIST angle/A] [ABOUT pointx/D pointy/D | ABOUTCENTER]
 [CONSTRAIN | FREE] [PRINT | NOPRINT]
 [DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]

Format:	Parameter	Values to enter
	POSITION	is the coordinates of the box.
	NORMAL	sets square corners. (Corner radii=0)
	ROUND	sets outward rounded corners. (Corner radii>0)
	SCALLOP	sets inward rounded corners. (Corner radii>0)
	BEVEL	sets angled corners. (Corner radii>0)
	INSET	sets square insert corners. (Corner radii>0)
	CORNER	is the corner radius.
	ECORNER	is the elliptical corner radius.
	ROTATE	is the rotation angle.
	SKEW	is the slant and twist angle.
	SLANT	is the slant angle.
	TWIST	is the twist angle.
	ABOUT	is the rotation point.
	ABOUTCENTER	rotates around its center.
	CONSTRAIN	toggles on the resizing constraint.
	FREE	toggles off the resizing constraint.
	PRINT	toggles on the print flag.
	NOPRINT	toggles off the print flag.
	DOCUMENT	is the document name. (Default=current)
	WINDOW	is the window name. (Default=current)
	OBJECTID	is the number of the object to manipulate.

Notes: All options default to the current values if not specified.

Example: editbox position 1.75 1.5 2.75 2.5

```

editbox position 3 3 6 6 page 'project.doc~8'
editbox position 6 7 8 9 ecorner 0.1 0.2 skew 0 45

```

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.2 editcolumn

EDITCOLUMN

Purpose: Sets (edits) the coordinates of a selected text column frame.

Syntax: editcolumn [POSITION left/D top/D right/D bottom/D]
 [COLUMNS number/I] [GUTTER space/D]
 [ROTATE angle/A | SKEW slantangle/A twistangle/A | SLANT angle/A |
 TWIST angle/A] [ABOUT pointx/D pointy/D | ABOUTCENTER]
 [CONSTRAIN | FREE] [PRINT | NOPRINT]
 [DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]

Format:	Parameter	Values to enter
	POSITION	is the coordinates of the text column frame.
	COLUMNS	is the number of columns in the frame.
	GUTTER	is the space between columns.
	ROTATE	is the rotation angle.
	SKEW	is the slant and twist angle.
	SLANT	is the slant angle.
	TWIST	is the twist angle.
	ABOUT	is the rotation point.
	ABOUTCENTER	rotates around its center.
	CONSTRAIN	toggles on the resizing constraint.
	FREE	toggles off the resizing constraint.
	PRINT	toggles on the print flag.
	NOPRINT	toggles off the print flag.
	DOCUMENT	is the document name. (Default=current)
	WINDOW	is the window name. (Default=current)
	OBJECTID	is the number of the object to manipulate.

Notes: All options default to the current values if not specified.

Example: editcolumn position 1 1 7.5 10 columns 2 gutter 0.25
 editcolumn position 1 1 5 6 rotate 45 page 'project.doc~8'

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.3 editcompound

EDITCOMPOUND

Purpose: Sets (edits) the bounding box coordinates of a selected compound

object.

Syntax: editcompound [POSITION left/D top/D right/D bottom/D]
 [CONTENTOFFSET offsetx/D offsety/D] [CONTENTSCALE scalex/P scaley/P]
 [ROTATE angle/A | SKEW slantangle/A twistangle/A | SLANT angle/A |
 TWIST angle/A] [ABOUT pointx/D pointy/D | ABOUTCENTER]
 [CONSTRAIN | FREE] [PRINT | NOPRINT]
 [DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]

Format: Parameter Values to enter
 POSITION is the coordinates of the frame.
 CONTENTOFFSET is the offset in the frame.
 CONTENTSCALE is the scale of the object in the frame.
 ROTATE is the rotation angle.
 SKEW is the slant and twist angle.
 SLANT is the slant angle.
 TWIST is the twist angle.
 ABOUT is the rotation point.
 ABOUTCENTER rotates around its center.
 CONSTRAIN toggles on the resizing constraint.
 FREE toggles off the resizing constraint.
 PRINT toggles on the print flag.
 NOPRINT toggles off the print flag.
 DOCUMENT is the document name.
 WINDOW is the window name.
 OBJECTID is the number of the object to manipulate.

Notes: All options default to the current values if not specified.

Example: editcompound position 1 1 5 6 rotate 45 page 'project.doc~8'

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.4 editdrawing

EDITDRAWING

Purpose: Sets (edits) the coordinates of a selected drawing.

Syntax: editdrawing [POSITION left/D top/D right/D bottom/D]
 [CONTENTOFFSET offsetx/D offsety/D] [CONTENTSCALE scalex/P scaley/P]
 [ROTATE angle/A | SKEW slantangle/A twistangle/A | SLANT angle/A |
 TWIST angle/A] [ABOUT pointx/D pointy/D | ABOUTCENTER]
 [CONSTRAIN | FREE] [PRINT | NOPRINT] [FRAMELESS | FRAMED]
 [INTERNAL | EXTERNAL] [FILE filepath/F]
 [DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]

Format: Parameter Values to enter
 POSITION is the coordinates of the frame.
 CONTENTOFFSET is the offset in the frame.
 CONTENTSCALE is the scale of the object in the frame.
 ROTATE is the rotation angle.
 SKEW is the slant and twist angle.

SLANT	is the slant angle.
TWIST	is the twist angle.
ABOUT	is the rotation point.
ABOUTCENTER	rotates around its center.
CONSTRAIN	toggles on the resizing constraint.
FREE	toggles off the resizing constraint.
PRINT	toggles on the print flag.
NOPRINT	toggles off the print flag.
FRAMELESS	makes the object frameless.
FRAMED	makes the object framed.
INTERNAL	stores the EPS object in the document. (Default)
EXTERNAL	does not store the EPS object in the document.
FILE	is the filepath and name of the drawing.
DOCUMENT	is the document name. (Default=current)
WINDOW	is the window name. (Default=current)
OBJECTID	is the number of the object to manipulate.

Notes: All options default to the current values if not specified.

Example: editdrawing position 1 1 5 6 rotate 45 page internal 'project.doc~8'
editdrawing external file 'Graphics:drawings/house.dr2d'

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.5 editellipse

EDITELLIPSE

Purpose: Sets (edits) the coordinates of a selected ellipse.

Syntax: editellipse [POSITION centerx/D centery/D radiusx/D radiusy/D]
[PIE | ARC | ELLIPSE] [ANGLES startangle/A endangle/A]
[ROTATE angle/A | SKEW slantangle/A twistangle/A | SLANT angle/A |
TWIST angle/A] [ABOUT pointx/D pointy/D | ABOUTCENTER]
[CONSTRAIN | FREE] [PRINT | NOPRINT]
[DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]

Format:	Parameter	Values to enter
	POSITION	is the coordinates of the object.
	PIE	is a closed arc of an ellipse.
	ARC	is an open arc of an ellipse
	ELLIPSE	is a closed ellipse.
	ANGLES	are the starting and ending angles.
	ROTATE	is the rotation angle.
	SKEW	is the slant and twist angle.
	SLANT	is the slant angle.
	TWIST	is the twist angle.
	ABOUT	is the rotation point.
	ABOUTCENTER	rotates around its center.
	CONSTRAIN	toggles on the resizing constraint.
	FREE	toggles off the resizing constraint.
	PRINT	toggles on the print flag.
	NOPRINT	toggles off the print flag.

DOCUMENT is the document name. (Default=current)
 WINDOW is the window name. (Default=current)
 OBJECTID is the number of the object to manipulate.

Notes: All options default to the current values if not specified.

Example: editellipse position 3 3 1.5 1.5
 editellipse position 3 3 1.5 1.5 page 'project.doc~6' constrain
 editellipse position 5.25 6.125 1.5 2.5 pie angles 45 90 print

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.6 editeps

DRAWEPS

Purpose: Sets (edits) the coordinates and options of an EPS illustration.

Syntax: editeps [POSITION left/D top/D right/D bottom/D]
 [CONTENTOFFSET offsetx/D offsety/D] [CONTENTSCALE scalex/P scaley/P]
 [ROTATE angle/A | SKEW slantangle/A twistangle/A | SLANT angle/A |
 TWIST angle/A] [ABOUT pointx/D pointy/D | ABOUTCENTER]
 [CONSTRAIN | FREE] [PRINT | NOPRINT] [FRAMELESS | FRAMED]
 [INTERNAL | EXTERNAL] [FILE filepath/F]
 [DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]

Format:	Parameter	Values to enter
	POSITION	is the coordinates of the frame.
	CONTENTOFFSET	is the offset in the frame.
	CONTENTSCALE	is the scale of the object in the frame.
	ROTATE	is the rotation angle.
	SKEW	is the slant and twist angle.
	SLANT	is the slant angle.
	TWIST	is the twist angle.
	ABOUT	is the rotation point.
	ABOUTCENTER	rotates around its center.
	CONSTRAIN	toggles on the resizing constraint.
	FREE	toggles off the resizing constraint.
	PRINT	toggles on the print flag.
	NOPRINT	toggles off the print flag.
	FRAMELESS	makes the object frameless.
	FRAMED	makes the object framed.
	INTERNAL	stores the EPS object in the document.
	EXTERNAL	does not store the EPS object in the document.
	FILE	is the filepath and name of the EPS object.
	DOCUMENT	is the document name. (Default=current)
	WINDOW	is the window name. (Default=current)
	OBJECTID	is the number of the object to manipulate.

Notes: All options default to the current values if not specified.

Example: editeps position 1.75 1.5 2.75 2.5
 editeps position 3 3 6 6 offset -2 -2 contentscale 80 constrain

```
editeps position 2 2.5 3.125 5.625 page 'project.doc~6'
```

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.7 editgrid

EDITGRID

Purpose: Sets (edits) the coordinates of a selected grid object.

Syntax: editgrid [POSITION pointx1/D pointy1/D pointx2/D pointy2/D | POINTS pointx1/D pointy1/D pointx2/D pointy2/D pointx3/D pointy3/D pointx4/D pointy4/D> [DIVISIONS numberx/I numbery/I] [ROTATE angle/A | SKEW slantangle/A twistangle/A | SLANT angle/A | TWIST angle/A] [ABOUT pointx/D pointy/D | ABOUTCENTER] [CONSTRAIN | FREE] [PRINT | NOPRINT] [DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]

Format:	Parameter	Values to enter
	POSITION	is the coordinates of the object.
	POINTS	is the coordinates of the vertices of a non-rectangular grid. (Numbered counterclockwise)
	DIVISIONS	are the number of grid cells in each direction.
	ROTATE	is the rotation angle.
	SKEW	is the slant and twist angle.
	SLANT	is the slant angle.
	TWIST	is the twist angle.
	ABOUT	is the rotation point.
	ABOUTCENTER	rotates around its center.
	CONSTRAIN	toggles on the resizing constraint.
	FREE	toggles off the resizing constraint.
	PRINT	toggles on the print flag.
	NOPRINT	toggles off the print flag.
	DOCUMENT	is the document name. (Default=current)
	WINDOW	is the window name. (Default=current)
	OBJECTID	is the number of the object to manipulate.

Notes: All options default to the current values if not specified.

Example: editgrid position 2 2 6 6 divisions 5 10
 editgrid position 2 2 6 6 divisions 5 10 constrain noprint
 editgrid points 1 1 1 3 5 6 2 3 page 'project.doc~6'

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.8 editgroup

EDITGROUP

Purpose: Sets (edits) the coordinates of a selected group.

Syntax: editgroup [POSITION left/D top/D right/D bottom/D]
 [CONTENTOFFSET offsetx/D offsety/D] [CONTENTSCALE scalex/P scaley/P]
 [ROTATE angle/A | SKEW slantangle/A twistangle/A | SLANT angle/A |
 TWIST angle/A] [ABOUT pointx/D pointy/D | ABOUTCENTER]
 [CONSTRAIN | FREE] [PRINT | NOPRINT]
 [DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]

Format:	Parameter	Values to enter
	POSITION	is the coordinates of the frame.
	CONTENTOFFSET	is the offset in the frame.
	CONTENTSCALE	is the scale of the object in the frame.
	ROTATE	is the rotation angle.
	SKEW	is the slant and twist angle.
	SLANT	is the slant angle.
	TWIST	is the twist angle.
	ABOUT	is the rotation point.
	ABOUTCENTER	rotates around its center.
	CONSTRAIN	toggles on the resizing constraint.
	FREE	toggles off the resizing constraint.
	PRINT	toggles on the print flag.
	NOPRINT	toggles off the print flag.
	DOCUMENT	is the document name. (Default=current)
	WINDOW	is the window name. (Default=current)
	OBJECTID	is the number of the object to manipulate.

Notes: All options default to the current values if not specified.

Example: editgroup position 1.75 1.5 2.75 2.5
 editgroup position 3 3 6 6 offset -2 -2 contentscale 80 constrain
 editgroup position 2 2.5 3.125 5.625 page 'project.doc~6'

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.9 editline

EDITLINE

Purpose: Sets (edits) the coordinates of a selected line.

Syntax: editline [POSITION left/D top/D right/D bottom/D]
 [ROTATE angle/A | SKEW slantangle/A twistangle/A | SLANT angle/A |
 TWIST angle/A] [ABOUT pointx/D pointy/D | ABOUTCENTER]
 [CONSTRAIN | FREE] [PRINT | NOPRINT]
 [DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]

Format:	Parameter	Values to enter
	POSITION	is the coordinates of the object.
	ROTATE	is the rotation angle.

SKEW is the slant and twist angle.
 SLANT is the slant angle.
 TWIST is the twist angle.
 ABOUT is the rotation point.
 ABOUTCENTER rotates around its center.
 CONSTRAIN toggles on the resizing constraint.
 FREE toggles off the resizing constraint.
 PRINT toggles on the print flag.
 NOPRINT toggles off the print flag.
 DOCUMENT is the document name. (Default=current)
 WINDOW is the window name. (Default=current)
 OBJECTID is the number of the object to manipulate.

Notes: All options default to the current values if not specified.

Example: editline position 1 1 5 5
 editline position 2 1.25 8.5 11 page 'project.doc~6' noprint

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.10 editobject

EDITOBJECT

Purpose: Sets (edits) the edit options for any object(s). This command allows you to edit different types of objects without changing their coordinates.

Syntax: editobject [ROTATE angle/A | SKEW slantangle/A twistangle/A |
 SLANT angle/A | TWIST angle/A]
 [ABOUT pointx/D pointy/D | ABOUTCENTER] [CONSTRAIN | FREE]
 [PRINT | NOPRINT] [DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]

Format: Parameter Values to enter
 ROTATE is the rotation angle.
 SKEW is the slant and twist angle.
 SLANT is the slant angle.
 TWIST is the twist angle.
 ABOUT is the rotation point.
 ABOUTCENTER rotates around its center.
 CONSTRAIN toggles on the resizing constraint.
 FREE toggles off the resizing constraint.
 PRINT toggles on the print flag.
 NOPRINT toggles off the print flag.
 DOCUMENT is the document name. (Default=current)
 WINDOW is the window name. (Default=current)
 OBJECTID is the number of the object to manipulate.

Notes: All options default to the current values if not specified.

Example: editobject rotate 45 about 1 2.25 page 'project.doc~6' noprint

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.11 editpalette

EDITPALETTE

Purpose: Displays the Edit palette.

Syntax: editpalette [SHOW | HIDE | TOGGLE] [AT pointx/I pointy/I] [SAVE]

Format:	Parameter	Values to enter
	SHOW	toggles on the palette.
	HIDE	toggles off the palette.
	TOGGLE	toggles the palette on and off.
	AT	sets the palette position.
	SAVE	saves the values to PageStream3.prefs.

Note: The TOGGLE parameter may not be used in conjunction with the SAVE parameter.

Example: editpalette toggle
 editpalette show at 0 350 save
 editpalette hide

Command Format

1.12 editpath

EDITPATH

Purpose: Sets (edits) the bounding box coordinates of a selected path.

Syntax: editpath [POSITION left/D top/D right/D bottom/D]
 [ROTATE angle/A | SKEW slantangle/A twistangle/A | SLANT angle/A |
 TWIST angle/A] [ABOUT pointx/D pointy/D | ABOUTCENTER]
 [CONSTRAIN | FREE] [PRINT | NOPRINT]
 [DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]

Format:	Parameter	Values to enter
	POSITION	is the coordinates of the object.
	ROTATE	is the rotation angle.
	SKEW	is the slant and twist angle.
	SLANT	is the slant angle.
	TWIST	is the twist angle.
	ABOUT	is the rotation point.
	ABOUTCENTER	rotates around its center.
	CONSTRAIN	toggles on the resizing constraint.
	FREE	toggles off the resizing constraint.
	PRINT	toggles on the print flag.
	NOPRINT	toggles off the print flag.
	DOCUMENT	is the document name. (Default=current)

WINDOW is the window name. (Default=current)
 OBJECTID is the number of the object to manipulate.

Notes: All options default to the current values if not specified.

Example: editpath position 1.75 1.5 2.75 2.5
 editpath position 3 3 6 6 constrain
 editpath position 2 2.5 3.125 5.625 page 'project.doc~6'

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.13 editpicture

EDITPICTURE

Purpose: Sets (edits) the coordinates and options of a picture frame.

Syntax: editpicture [POSITION left/D top/D right/D bottom/D]
 [CONTENTOFFSET offsetx/D offsety/D] [CONTENTSCALE scalex/P scaley/P]
 [ROTATE angle/A | SKEW slantangle/A twistangle/A | SLANT angle/A |
 TWIST angle/A] [ABOUT pointx/D pointy/D | ABOUTCENTER]
 [CONSTRAIN | FREE] [PRINT | NOPRINT] [FRAMELESS | FRAMED]
 [INTERNAL | EXTERNAL] [FILE filepath/F] [DPI xdpi/I ydpi/I]
 [DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]

Format:	Parameter	Values to enter
	POSITION	is the coordinates of the frame.
	CONTENTOFFSET	is the offset in the frame.
	CONTENTSCALE	is the scale of the object in the frame.
	ROTATE	is the rotation angle.
	SKEW	is the slant and twist angle.
	SLANT	is the slant angle.
	TWIST	is the twist angle.
	ABOUT	is the rotation point.
	ABOUTCENTER	rotates around its center.
	CONSTRAIN	toggles on the resizing constraint.
	FREE	toggles off the resizing constraint.
	PRINT	toggles on the print flag.
	NOPRINT	toggles off the print flag.
	FRAMELESS	makes the object frameless.
	FRAMED	makes the object framed.
	INTERNAL	stores the picture in the document.
	EXTERNAL	does not store the picture in the document.
	FILE	is the filepath and name of the picture.
	DPI	is the resolution of the picture.
	DOCUMENT	is the document name. (Default=current)
	WINDOW	is the window name. (Default=current)
	OBJECTID	is the number of the object to manipulate.

Notes: All options default to the current values if not specified.

Example: editpicture internal
 editpicture contentoffset -0.5 -9.25 contentscale 125 125

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.14 editpolygon

EDITPOLYGON

Purpose: Sets (edits) the coordinates of a selected regular polygon.

Syntax: editpolygon [POSITION centerx/D centery/D radiusx/D radiusy/D]
 [SIDES number/I] [OFFSETANGLE angle/A] [DEFLECTION amount/P]
 [ANGLEDEFLECTION amount/P] [NORMAL | STAR | PUFFY | SCALLOP | WAVY]
 [ROTATE angle/A | SKEW slantangle/A twistangle/A | SLANT angle/A |
 TWIST angle/A] [ABOUT pointx/D pointy/D | ABOUTCENTER]
 [CONSTRAIN | FREE] [PRINT | NOPRINT]
 [DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]

Format:	Parameter	Values to enter
	POSITION	is the coordinates of the object.
	SIDES	is the number of sides. (>2)
	OFFSETANGLE	is pre-rotation angle.
	DEFLECTION	is the alternate point radius.
	ANGLEDEFLECTION	is the alternate point angle.
	NORMAL	sets straight sides.
	STAR	sets straight sides with with alternate points.
	PUFFY	sets outward curved sides with alternate points.
	SCALLOP	sets inward curved sides with alternate points.
	WAVY	sets alternating in/outward curved sides with alternate points.
	ROTATE	is the rotation angle.
	SKEW	is the slant and twist angle.
	SLANT	is the slant angle.
	TWIST	is the twist angle.
	ABOUT	is the rotation point.
	ABOUTCENTER	rotates around its center.
	CONSTRAIN	toggles on the resizing constraint.
	FREE	toggles off the resizing constraint.
	PRINT	toggles on the print flag.
	NOPRINT	toggles off the print flag.
	DOCUMENT	is the document name. (Default=current)
	WINDOW	is the window name. (Default=current)
	OBJECTID	is the number of the object to manipulate.

Notes: All options default to the current values if not specified.

Example: editpolygon sides 11 wavy

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.15 edittext

EDITTEXT

Purpose: Deletes the character or word to the left or right of the insertion point.

Syntax: edittext [BACKSPACE | DELETE | DELETEWORDLEFT | DELETEWORDRIGHT]
[DOCUMENT name/S | WINDOW name/S]

Format:	Parameter	Values to enter
	BACKSPACE	deletes the character to the left of the insertion point.
	DELETE	deletes the character to the right of the insertion point.
	DELETEWORDLEFT	deletes the word to the left of the insertion point.
	DELETEWORDRIGHT	deletes the word to the right of the insertion point.
	DOCUMENT	is the document name. (Default=current)
	WINDOW	is the window name. (Default=current)

Example: edittext delete
edittext backspace

Command Format

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.16 edittextobj

EDITTEXTOBJ

Purpose: Sets (edits) the coordinates of a selected frameless text object.

Syntax: edittextobj [POSITION left/D top/D right/D bottom/D]
[ROTATE angle/A | SKEW slantangle/A twistangle/A | SLANT angle/A |
TWIST angle/A] [ABOUT pointx/D pointy/D | ABOUTCENTER]
[CONSTRAIN | FREE] [PRINT | NOPRINT]
[TEXTFX <SHAPE | PATH | NONE> [DISTORTX <TRUE|FALSE>]
[BENDLINES <TRUE|FALSE>] [SCALETOFIT <TRUE|FALSE>] <TEXTFXPATH
[ABOVEPATH | ONPATH | BELOWPATH]
<[MOVE TO pointx/D pointy/D] | [LINE TO pointx/D pointy/D] |
[CURVE TO curvex1/D curvey1/D curvex2/D curvey2/D pointx/D pointy/D] |
[ARCO centerx/D centery/D radiusx/D radiusy/D startangle/A
endangle/A slantangle/A twistangle/A] |
[ARCNTO centerx/D centery/D radiusx/D radiusy/D startangle/A
endangle/A slantangle/A twistangle/A] |
[CLOSEPATH]>>>]
[DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]

Format:	Parameter	Values to enter
	POSITION	is the coordinates of the frameless text object.
	ROTATE	is the rotation angle.
	SKEW	is the slant and twist angle.
	SLANT	is the slant angle.
	TWIST	is the twist angle.

ABOUT is the rotation point.
 ABOUTCENTER rotates around its center.
 CONSTRAIN toggles on the resizing constraint.
 FREE toggles off the resizing constraint.
 PRINT toggles on the print flag.
 NOPRINT toggles off the print flag.
 TEXTFX sets a TextFX shape or curve. (This requires the TextFX extension which is sold separately.)
 SHAPE sets a TextFX warp shape.
 PATH sets a TextFX curve.
 NONE specifies a normal frameless text object.
 DISTORTX toggles horizontal distortion.
 BENDLINES toggles curving of line strokes.
 SCALETOFIT scales text to fit the TextFX curve.
 TEXTFXPATH sets the warp shape or curve path.
 ABOVEPATH sets text above a curve.
 ONPATH sets text on (centered vertically) on a curve.
 BELOWPATH sets text below a curve.
 MOVETO starts the warp shape or curve path.
 LINETO adds a TextFX path line to the point.
 CURVETO adds a TextFX path bézier curve to the point.
 ARCTO adds a TextFX path arc counterclockwise to the point.
 ARCNTO adds a TextFX path arc clockwise to the point.
 CLOSEPATH closes the TextFX path.
 DOCUMENT is the document name. (Default=current)
 WINDOW is the window name. (Default=current)
 OBJECTID is the number of the object to manipulate.

Notes: All options default to the current values if not specified.

TEXTFXPATH and its related parameters require the TextFX extension, which is sold separately.

Example: edittextobj position 1 1 7.5 10
 edittextobj position 1 1 5 6 rotate 45 page 'project.doc~8'
 edittextobj textfx shape textfxpath moveto 0i 0.25i curveto 1i 0i 5i 0i ←
 5.5i 1i
 moveto 2i 3i curveto 3i 2i 5i 2i 6i 2i

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.17 endcommandcapture

ENDCOMMANDCAPTURE

Purpose: End buffering text commands. This is used to avoid a separate redraw for each text attribute application.

Syntax: endcommandcapture [DOCUMENT name/S | WINDOW name/S]

Format: Parameter Values to enter

DOCUMENT is the document name. (Default=current)

WINDOW is the window name. (Default=current)

Example: `endcommandcapture`
`endcommandcapture document 'project.doc'`

See also `begincommandcapture`

Command Format

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.18 exportgraphic

EXPORTGRAPHIC

Purpose: This exports a graphic object.

Syntax: `exportgraphic [FILE filepath/F] [FILTER name/S]`
`[STATUS | NOSTATUS] [FORCE | ALERT | QUIET]`
`[DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]`

Format:	Parameter	Values to enter
	FILE	is the filename and path of the file to export
	FILTER	is the import/export filter to use.
	STATUS	displays a status indicator while saving the file.
	NOSTATUS	does not display a status indicator. (Default)
	FORCE	overwrites an existing file (if present).
	ALERT	opens an alert if it will overwrite an existing file.
	QUIET	overwrites an existing file (if present). (Default)
	DOCUMENT	is the document name. (Default=current)
	WINDOW	is the window name. (Default=current)
	OBJECTID	is the number of the object to export.

Notes: If the FILE parameter is not specified, a file requester will open.

If the FILTER paragraph is not specified, the Place Graphic requester will open. Valid filter parameters at the date of this writing are: IFFILBM, TIFF, GIF, BMP, IFFDR2D, IFFILUS. The names of filters are normally the same as the filename, less the '.filter' extension. You can determine the filter name by typing 'version name.filter' in a shell, where name is the filter to examine.

Example: `exportgraphic file ram:MyPicture.IIIBM filter iffilm status`

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.19 exporttext

EXPORTTEXT

Purpose: This exports the selected text.

Syntax: `exporttext [FILE filepath/F] [FILTER name/S] [DOCUMENT name/S]`
`[STATUS | NOSTATUS] [FORCE | ALERT | QUIET]`
`[CONVERTQUOTE <TRUE|FALSE>] [CONVERTDASH <TRUE|FALSE>]`
`[LINEHASLF <TRUE|FALSE>] [AMIGA | WINDOWS | MSDOS | MACINTOSH]`
`[RETAINFORMAT <TRUE|FALSE>] [TEXTCODE name/S]`

Format:	Parameter	Values to enter
	FILE	is the filename and path of the file to export
	FILTER	is the import/export filter to use.
	DOCUMENT	is the document name. (Default=current)
	STATUS	displays a status indicator while saving the file.
	NOSTATUS	does not display a status indicator. (Default)
	FORCE	overwrites an existing file (if present).
	ALERT	opens an alert if it will overwrite an existing file.
	QUIET	overwrites an existing file (if present). (Default)
	CONVERTQUOTE	converts typewriter quotation marks to typographic quotation marks. (Not applicable to IFF CTXT.) (DEFAULT=TRUE)
	CONVERTDASH	converts double dashes to en dashes and triple dashes to em dashes. (Not applicable to IFF CTXT.) (DEFAULT=TRUE)
	LINEHASLF	assumes that each line ends with a line feed rather than each paragraph. (Applicable to ASCII and IFF FTXT only.) (Default=FALSE)
	RETAINFORMAT	exports the attributes as well as the text. (Not applicable to ASCII.) (Default=TRUE)
	AMIGA	specifies that the Amiga character set should be used. (Applicable to ASCII only.) (Default)
	WINDOWS	specifies that the Windows character set should be used. (Applicable to ASCII only.)
	MSDOS	specifies that the MS-DOS character set should be used. (Applicable to ASCII only.)
	MACINTOSH	specifies that the Mac character set should be used. (Applicable to ASCII only.)
	TEXTCODE	is the text code engine to use. (Default=NONE)

Notes: If the FILE parameter is not specified, a file requester will open.

If the FILTER paragraph is not specified, the Place Graphic requester will open. Valid filter parameters at the date of this writing are: ASCII, IFFFTXT, IFFCTXT, FinalCopy2, FinalWriter, WordWorth, ProWrite, WordPerfect and Word. The names of filters are normally the same as the filename, less the '.filter' extension. You can determine the filter name by typing 'version name.filter' in a shell, where name is the filter to examine.

Example: `exporttext file ram:MyText.FTXT filter iff.txt status`

Command Format

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.20 filled

FILLED

Purpose: Toggles the fill for an object or text, or for a style tag. If object or text is chosen and nothing is selected, it toggles the fill for future objects or text, as applicable.

Syntax: filled <ON | OFF | TOGGLE>
[DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]

Format: Parameter Values to enter
ON toggles on the fill.
OFF toggles off the fill.
TOGGLE toggles the fill.
DOCUMENT is the document name.
WINDOW is the window name.
OBJECTID is the number of the object. (Default=current)

Example: filled on
filled toggle document 'project.doc'

See also SETFILL

Command Format
Object ID numbers
DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.21 findtext

FINDTEXT

Purpose: Searches for a text string.

Syntax: findtext **

Format: Parameter Values to enter
**

Example: **

See also REPLACETEXT.

Command Format
DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.22 fontpalette

FONTPALETTE

Purpose: Displays the font palette.

Syntax: fontpalette [SHOW | HIDE | TOGGLE] [AT pointx/I pointy/I]
[SIZE width/I height/I] [SAVE]

Format: Parameter Values to enter

```

SHOW      toggles on the palette.
HIDE      toggles off the palette.
TOGGLE    toggles the palette on and off.
AT        sets the palette position.
SIZE      sets the palette size in pixels. (Min=160x50)
SAVE      saves the values to PageStream3.prefs.

```

Note: The TOGGLE parameter may not be used in conjunction with the SAVE parameter.

```

Example: fontpalette toggle
fontpalette show at 500 300 size 180 172 save
fontpalette hide

```

Command Format

1.23 freearexxlist

FREEAREXXLIST

External macros only!

Purpose: Unallocates an allocated list. You should always do this when you are finished with a list or when the script is exiting, to prevent loss of memory.

Syntax: freearexxlist <listhandle/I>

Format: Parameter Values to enter
listhandle is the list to unallocate.

Example: freearexxrequester TestRequester

```

See also ADDAREXXLIST      Adds an item to a list.
ALLOCAREXXLIST            Allocates a list.
ADDAREXXGADGET            Adds a gadget.
ALLOCAREXXREQUESTER      Allocates a requester.
DOAREXXREQUESTER          Opens a requester.
FREEAREXXREQUESTER        Unallocates a requester.
GETAREXXGADGET            Gets the final status of a gadget.
SETAREXXGADGET            Sets a gadget.

```

Command Format

1.24 freearexxrequester

FREEAREXXREQUESTER

External macros only!

Purpose: Unallocates an allocated requester. You should always do this when you are finished with a requester or when the script is exiting, to

prevent loss of memory.

Syntax: freearexxrequester <reqhandle/I>

Format: Parameter Values to enter
reqhandle is the requester to unallocate.

Example: freearexxrequester TestRequester

See also ADDAREXXLIST Adds an item to a list.
ALLOCAREXXLIST Allocates a list.
ADDAREXXGADGET Adds a gadget.
ALLOCAREXXREQUESTER Allocates a requester.
DOAREXXREQUESTER Opens a requester.

FREEAREXXLIST
Unallocates an arexx list.
GETAREXXGADGET Gets the final status of a gadget.
SETAREXXGADGET Sets a gadget.

Command Format

1.25 getarexxgadget

GETAREXXGADGET

External macros only!

Purpose: Gets the current state of a gadget. This is normally used after a DOREQUESTER command returns.

Syntax: getarexxgadget <reqhandle/I gadgethandle/I attribute/S>

Format: Parameter Values to enter
reqhandle is the requester to interrogate.
gadgethandle is the is the gadget to interrogate.
attribute is the gadget attribute to get.

Result: The state of the gadget is returned to the RESULT variable.

Returns:	Attribute	Specify	Result
	CHECKBOX	CHECKED	0:unchecked, 1:checked
	TEXT	STRING	text string
	POPUP	CURRENT	selected item #
	CYCLE	CURRENT	selected item #
	SLIDER	CURRENT	selected item #
	SCROLLLIST	CURRENT	selected item #
	MULTILINE	STRING	text string

Note: You do not need to use this command for exit gadgets since the selected exit gadget is returned to the RESULT variable by the DOREQUESTER command.

Example: getarexxgadget reqhandle name_gadget string
name=RESULT

```
getarexxgadget textrequester format cycle
format=RESULT
```

See also

ADDAREXXLIST	Adds an item to a list.
ADDAREXXGADGET	Adds a gadget.
ALLOCAREXXLIST	Allocates a list.
ALLOCAREXXREQUESTER	Allocates a requester.
DOAREXXREQUESTER	Opens a requester.

```
FREEAREXXLIST
  Unallocates an arexx list.
```

```
FREEAREXXREQUESTER
  Unallocates a requester.
```

```
SETAREXXGADGET      Sets a gadget.
```

Command Format

1.26 getarticle

GETARTICLE

External macros only!

Purpose: Gets information concerning an article.

Syntax: `getarticle [FILEINFO stem/V] [FRAMES fstem/V]`
`[ARTICLE name/S | DOCUMENT name/S | WINDOW name/S | ARTICLEID number/I]`

Format:

Parameter	Values to enter
FILEINFO	gets the status of the article.
FRAMES	gets the number and IDs of frames in the article.
ARTICLE	is the name of the article. (Default=current)
DOCUMENT	is the document name. (Default=current article)
WINDOW	is the window name. (Default=current article)
ARTICLEID	is the number of the article. (Default=current)

Result:

FILEINFO:	
stem.mode	INTERNAL EXTERNAL
stem.file	the article filepath and name

FRAMES:

fstem.count	number of frames in the article
fstem.0	the object id of the first frame in the article
fstem.1	the object id of the second frame in the article
etc.	

Example: `getarticle info /* prints the article status to the output console */`
`say 'The article is: ' || stem.mode ||`
`if stem.mode='EXTERNAL' then say 'The article is stored in: ' || stem.path`

Command Format

Article ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.27 getarticledesc

GETARTICLEDESC

External macros only!

Purpose: Gets the description of an article.

Syntax: `getarticledesc`
 [ARTICLE name/S | DOCUMENT name/S | WINDOW name/S | ARTICLEID number/I]

Format: Parameter Values to enter
 ARTICLE is the name of the article. (Default=current)
 DOCUMENT is the document name. (Default=current article)
 WINDOW is the window name. (Default=current article)
 ARTICLEID is the number of the article. (Default=current)

Result: The article description is returned to the RESULT variable.

Example: `getarticledesc /* returns the article description */`
`artdesc=RESULT`

```
getarticledesc article 'Story1'
artdesc=RESULT
```

Command Format

Article ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.28 getarticleids

GETARTICLEIDS

External macros only!

Purpose: Gets the number of articles in a document or chapter and their IDs.

Syntax: `getarticles <stem/V>`
 [DOCUMENT name/S | CHAPTER name/S | WINDOW name/S]

Format: Parameter Values to enter
 stem is the name of a stem variable for the chapter names.
 DOCUMENT is the document name. (Default=current)
 CHAPTER is the document/chapter name.
 WINDOW is the window name. (Default=current)

Result: The number of articles in the document or chapter is returned to RESULT.

The IDs of the articles are returned to `stem.#` where # is a

number from 0 to the number of chapters less 1.

```
Example: getarticles iArticle /* will print the article IDs to the output ↔
console */
iArticle.count=result
do i=0 to iArticle.count-1
  say iArticle.i
end i
```

Command Format

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.29 getarticlename

GETARTICLENAME

External macros only!

Purpose: Gets the name of an article.

Syntax: getarticlename [DOCUMENT name/S | WINDOW name/S | ARTICLEID number/I]

Format: Parameter Values to enter
 DOCUMENT is the document name. (Default=current article)
 WINDOW is the window name. (Default=current article)
 ARTICLEID is the number of the article. (Default=current)

Result: The article name is returned to the RESULT variable.

```
Example: getarticlename /* returns the article name */
artname=RESULT
```

Command Format

Article ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.30 getarticleoverset

GETARTICLEOVERSET

External macros only!

Purpose: Determines whether an article is overset or not.

Syntax: getarticleoverset
 [ARTICLE name/S | DOCUMENT name/S | WINDOW name/S | ARTICLEID number/I]

Format: Parameter Values to enter
 ARTICLE is the name of the article. (Default=current)
 DOCUMENT is the document name. (Default=current article)
 WINDOW is the window name. (Default=current article)
 ARTICLEID is the number of the article. (Default=current)

Result: TRUE is returned to the RESULT variable if it is overset.
 FALSE is returned to the RESULT variable if it isn't overset.

Example: `getarticleoverset`
`if RESULT = TRUE then say "The article is overset."`

Command Format
 Article ID numbers
 DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.31 getarticlewordcount

GETARTICLEWORDCOUNT

External macros only!

Purpose: Gets the number of words in an article.

Syntax: `getarticlewordcount`
`[ARTICLE name/S | DOCUMENT name/S | WINDOW name/S | ARTICLEID number/I]`

Format: Parameter Values to enter
 ARTICLE is the name of the article. (Default=current)
 DOCUMENT is the document name. (Default=current article)
 WINDOW is the window name. (Default=current article)
 ARTICLEID is the number of the article. (Default=current)

Result: The number of words in the article is returned to the RESULT variable.

Example: `getarticlewordcount`
`numwords=RESULT`

`getarticlewordcount article 'Story1'`
`numwords=RESULT`

Command Format
 Article ID numbers
 DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.32 getbleed

GETBLEED

External macros only!

Purpose: Gets the bleed for a master page.

Syntax: `getbleed <stem/V> [MASTERPAGE name/S]`

Format: Parameter Values to enter

stem is the name of a stem variable for the information.
 MASTERPAGE is the master page name. (Default=current)

Result: stem.h Horizontal bleed
 stem.v Vertical bleed

Example: getbleed info /* will print the margin guides */
 say 'Horizontal bleed: ' || info.h
 say 'Vertical bleed: ' || info.v

Command Format

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.33 getbox

GETBOX

External macros only!

Purpose: Gets coordinates and information for a box.

Syntax: getbox [POSITION pstem/V] [CORNER cstem/V]
 [ROTATION rstem/V] [ABOUT rstem/V] [CONSTRAIN cflag/V]
 [PRINT pflag/V] [DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]

Format:	Parameter	Values to enter
	POSITION	gets the coordinates of the box.
	CORNER	gets the corner radii.
	ROTATION	gets the rotation of the box.
	ABOUT	gets the rotation point.
	CONSTRAIN	gets the proportional scale flag state.
	PRINT	gets the print flag state.
	DOCUMENT	is the document name.
	WINDOW	is the window name.
	OBJECTID	is the number of the object. (Default=current)

Result: The object ID is returned to the RESULT variable.

POSITION:
 pstem.left left coordinate
 pstem.top top coordinate
 pstem.right right coordinate
 pstem.bottom bottom coordinate

CORNER:
 cstem.type <NORMAL|ROUND|SCALLOP|BEVEL|INSET>
 cstem.cornerx horizontal corner radius
 cstem.cornery vertical corner radius

ROTATION:
 rstem.mode rotate about <POINT|CENTER>
 rstem.slant slant angle
 rstem.twist twist angle

ABOUT:

```

rstem.x      horizontal point
rstem.y      vertical point

```

CONSTRAIN: returns <ON|OFF>

PRINT: returns <ON|OFF>

```

Example: getbox position coord /* will print the box coordinates to the output ←
console */
say 'Left: ' || coord.left
say 'Top: ' || coord.top
say 'Right: ' || coord.right
say 'Bottom: ' || coord.bottom

```

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.34 getbusyrequester

GETBUSYREQUESTER

External macros only!

Purpose: Gets the Stop gadget state in an open busy requester.

Syntax: getbusyrequester <reqhandle/I>

Format: Parameter Values to enter
reqhandle is the handle of the requester to get.

Result: 0 is returned to the RESULT variable if the Stop gadget has not been pressed; 1 is returned if it has been pressed.

```

Example: openbusyrequester message 'Thinking hard...' thermometer enabled total ←
100 current 0
bh=RESULT
do for i = 1 to 100
  /* script does something here */
  /* abort if stop is pressed, else set thermometer */
  getbusyrequester bh
  if result=1 then break else setbusyrequester bh current i
end i
closebusyrequester bh

```

Command Format

1.35 getchapterdesc

GETCHAPTERDESC

External macros only!

Purpose: Gets the description of a chapter.

Syntax: `getchapterdesc [DOCUMENT name/S | CHAPTER name/S]`

Format: Parameter Values to enter
 DOCUMENT is the document name.
 CHAPTER is the document/chapter name. (Default=current)

Result: The description is returned to RESULT.

Example: `getchapterdesc document 'Project.doc' /* will return the description of ↵
 the current chapter */
 chapdesc=result`

Command Format

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.36 getchapternumber

GETCHAPTERNUMBER

External macros only!

Purpose: Gets the number of a chapter.

Syntax: `getchapternumber [DOCUMENT name/S | CHAPTER name/S]`

Format: Parameter Values to enter
 DOCUMENT is the document name.
 CHAPTER is the document/chapter name. (Default=current)

Result: The number is returned to RESULT.

Example: `getchapternumber document 'Project.doc' /* will return the number of ↵
 the current chapter */
 chapnum=result`

Command Format

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.37 getchapternumbering

GETCHAPTERNUMBERING

External macros only!

Purpose: Gets the chapter numbering system and start number.

Syntax: `getchapternumbering <stem/V>
 [DOCUMENT name/S | CHAPTER name/S | WINDOW name/S]`

Format: Parameter Values to enter
 stem is the name of a stem variable for the information.
 DOCUMENT is the document name. (Default=current)
 CHAPTER is the document/chapter name.

Result: stem.startmode AUTOMATIC|CUSTOM
 stem.start starting chapter number
 stem.format DEFAULT|LONG|ARABIC|ROMANUPPER|ROMANLOWER|
 ALPHAUPPER|ALPHALOWER
 stem.language DEFAULT|name
 stem.prefix string

Notes: This command actually returns more than what is listed, but the other information is not supported at this time.

Example: `getchapternumbering info /* will reset the start chapter number to one ↔
 higher if chapter numbering is set to custom */
 if info.startmode='CUSTOM' then do 'setchapternumbering start custom '|| ↔
 info.start+1`

Command Format

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.38 getchapters

GETCHAPTERS

External macros only!

Purpose: Gets the number of chapters in a document or chapter and their names.

Syntax: `getchapters <stem/V>
 [DOCUMENT name/S | CHAPTER name/S | WINDOW name/S]`

Format: Parameter Values to enter
 stem is the name of a stem variable for the chapter names.
 DOCUMENT is the document name. (Default=current)
 CHAPTER is the document/chapter name.
 WINDOW is the window name. (Default=current)

Result: The number of chapters in the document or chapter is returned to RESULT.

The names of the chapters are returned to stem.# where # is a number from 0 to the number of chapters less 1.

Example: `getchapters chapnames /* will print the chapter names to the output ↔
 console */
 numchaps=result
 do count=0 to numchaps-1
 say chapnames.count
 end count`

Command Format

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.39 getchoice

GETCHOICE

External macros only!

Purpose: Opens a requester with a short message and allows the user to click on one of one, two or three buttons, to make a choice.

Syntax: `getchoice <MESSAGE message/S> <BUTTON1 label/S>
[BUTTON2 label/S] [BUTTON3 label/S]`

Format:

Parameter	Values to enter
MESSAGE	is the message to show the user. (Max length=44)
BUTTON1	is the label for the right button. (Max length=8) This button will be the default gadget, so it should be used for "Ok" or similar purposes.
BUTTON2	is the label for the left button. (Max length=8) This button can be activated by pressing the Esc key, so it should be used for "Cancel" or similar purposes.
BUTTON3	is the label for the center button. (Max length=8)

Result: Returns the button number to the RESULT variable.

Notes: The button order is 2,3,1. This ensures that if only one button is used, it is the right button which is normally the desired style for an exit gadget. If you want a different button layout, or a longer message, design a custom macro requester with the ALLOCAREXXREQUESTER command.

Precede the character to underscore as a bound keyboard equivalent in the label name. For example, "_Ok" would make "O" the keyboard shortcut for the "Ok" gadget.

Example: `'getchoice message "Click on a button." button1 "_Ok" button2 "_Cancel"
buttonid=RESULT`

Command Format

1.40 getcolumn

GETCOLUMN

External macros only!

Purpose: Gets coordinates and information for a text column frame.

Syntax: `getcolumn [POSITION pstem/V] [COLUMNS count/V] [GUTTER space/V]
[ROTATION rstem/V] [ABOUT rstem/V] [CONSTRAIN cflag/V]
[PRINT pflag/V] [DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]`

Format:	Parameter	Values to enter
	POSITION	gets the coordinates of the frame.
	COLUMNS	gets the number of columns in the text frame.
	GUTTER	gets the gutter space between the columns.
	ROTATION	gets the rotation of the frame.
	ABOUT	gets the rotation point.
	CONSTRAIN	gets the proportional scale flag state.
	PRINT	gets the print flag state.
	DOCUMENT	is the document name.
	WINDOW	is the window name.
	OBJECTID	is the number of the object. (Default=current)

Result: The object ID is returned to the RESULT variable.

```
POSITION:
pstem.left      left coordinate
pstem.top       top coordinate
pstem.right     right coordinate
pstem.bottom    bottom coordinate
```

COLUMNS: returns the number of columns.

GUTTER: returns the space between the columns.

```
ROTATION:
rstem.mode      rotate about <POINT|CENTER>
rstem.slant     slant angle
rstem.twist     twist angle
```

```
ABOUT:
rstem.x         horizontal point
rstem.y        vertical point
```

CONSTRAIN: returns <ON|OFF>

PRINT: returns <ON|OFF>

```
Example: getcolumn position coord /* will print the frame coordinates to the ↵
output console */
say 'Left: ' || coord.left
say 'Top: ' || coord.top
say 'Right: ' || coord.right
say 'Bottom: ' || coord.bottom
```

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.41 getcolumnguides

GETCOLUMNGUIDES

External macros only!

Purpose: Gets the column guides for a master page.

Syntax: `getcolumnguides <stem/V> [MASTERPAGE name/S]`

Format: Parameter Values to enter
 stem is the name of a stem variable for the information.
 MASTERPAGE is the master page name. (Default=current)

Result: stem.count number of columns
 stem.gutter space between the columns

Example: `getcolumnguides info /* will print the column guide info to the output ↵
 console */
 say 'Number of columns: 'info.count
 say 'Gutter space: 'info.gutter`

Command Format

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.42 getcompound

GETCOMPOUND

External macros only!

Purpose: Gets coordinates and information for a compound object.

Syntax: `getcompound [POSITION pstem/V] [FRAME fflag/V]
 [CONTENTOFFSET cstem/V] [CONTENTSCALE cstem/V] [ROTATION rstem/V]
 [ABOUT rstem/V] [CONSTRAIN cflag/V] [PRINT pflag/V]
 [DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]`

Format: Parameter Values to enter
 POSITION gets the coordinates of the frame.
 CONTENTOFFSET gets the offset in the frame.
 CONTENTSCALE gets the scale of the object in the frame.
 ROTATION gets the rotation of the frame.
 ABOUT gets the rotation point.
 CONSTRAIN gets the proportional scale flag state.
 PRINT gets the print flag state.
 DOCUMENT is the document name.
 WINDOW is the window name.
 OBJECTID is the number of the object. (Default=current)

Result: The object ID is returned to the RESULT variable.

POSITION:
 pstem.left left coordinate
 pstem.top top coordinate
 pstem.right right coordinate
 pstem.bottom bottom coordinate

FRAME: returns <ON|OFF>

CONTENTOFFSET:

```

cstem.x      horizontal offset
cstem.y      vertical offset

CONTENTSCALE:
cstem.h      horizontal scale
cstem.v      vertical scale

ROTATION:
rstem.mode   rotate about <POINT|CENTER>
rstem.slant  slant angle
rstem.twist  twist angle

ABOUT:
rstem.x      horizontal point
rstem.y      vertical point

CONSTRAIN: returns <ON|OFF>

PRINT: returns <ON|OFF>

```

```

Example: getcompound position coord /* will print the compound object bounding ↔
       box to the output console */
       say 'Left:   ' || coord.left
       say 'Top:    ' || coord.top
       say 'Right:  ' || coord.right
       say 'Bottom: ' || coord.bottom

```

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.43 getcoord

GETCOORD

External macros only!

Purpose: Opens a small message requester to instruct the user to click on the page to return a coordinate value to the macro. The requester has a Cancel gadget.

Syntax: getcoord <stem/V> [MESSAGE message/S]

Format:

Parameter	Values to enter
stem	is the name of a stem variable for the mouse coordinates.
MESSAGE	is the message to display in the requester. (Max length=55)

Result: If Cancel is chosen, it sets RC to 10. If the user clicks on the page, it sets RC to 0 and returns the coordinates to the stem variable.

```

stem.x      horizontal coordinate
stem.y      vertical coordinate

```

```
Example: 'getcoord coord message "Click on the page"' /* will print the ↵
        coordinates to the output console */
        button=RC
        if RC=0 then do
            say coord.x
            say coord.y
        end
```

Command Format

1.44 getcursor

GETCURSOR

External macros only!

Purpose: Gets information about the insertion point or text selection(s).

Syntax: getcursor [INDEX number/I] [TYPE object/B] [ARTICLEID number/V]
 [FRAMESTART number/V] [FRAMEEND number/V] [FRAMESTARTID number/V]
 [FRAMEENDID number/V] [OBJECTID number/v] [LINESTART number/V]
 [LINEEND number/V] [CHARACTERSTART number/V] [CHARACTEREND number/V]
 [POSITIONSTART sstem/V] [POSITIONEND estem/V]
 [DOCUMENT name/S | WINDOW name/S]

Format: Parameter Values to enter

INDEX	is the text selection / insertion point about which to get information. (Default=1)
TYPE	is the type of text object.
ARTICLEID*	is the article ID of the article of the insertion point or text selection.
FRAMESTART*	is the number of the first frame in the selection.
FRAMEEND*	is the number of the last frame in the selection.
FRAMESTARTID*	is the object ID of the first frame in the selection.
FRAMEENDID*	is the object ID of the last frame in the selection.
OBJECTID**	is the object ID of the frameless text object.
LINESTART	is the ID of the first selected line.
LINEEND	is the ID of the last selected line.
CHARACTERSTART	is the ID of the first character selected.
CHARACTEREND	is the ID of the last character selected.
POSITIONSTART	is the position of the insertion point or start of the selected text.
POSITIONEND	is the position of the end of the selected text.
DOCUMENT	is the document name. (Default=current)
WINDOW	is the window name. (Default=current)

* apply only to text frames; will return 0 for frameless text objects.

** apply only to frameless text objects; will return 0 for text frames.

Results: TYPE: returns <ARTICLE|TEXTOBJ> where ARTICLE means the insertion point is in a text frame, and TEXTOBJ means the insertion point is in a text object.

POSITIONSTART:
 sstem.x horizontal coordinate
 sstem.y baseline coordinate
 sstem.top top of slug coordinate
 sstem.bottom bottom of slug coordinate

POSITIONEND:
 estem.x horizontal coordinate
 estem.y baseline coordinate
 estem.top top of slug coordinate
 estem.bottom bottom of slug coordinate

Notes: If there is only one insertion point or text selection, INDEX should be 1. If there are multiple text selections, (found out with

`getcursorcoun`
`)`, you may want to use this command more than once to get information about each selection.

FRAMESTART and FRAMEEND will be equal if no text is selected.

CHARACTERSTART and CHARACTEREND will be equal if no text is selected.

LINESTART, LINEEND, CHARACTERSTART and CHARACTEREND numbers don't have any value by themselves. They take into account hidden commands and are therefore not suitable for counting the number of characters selected.

Example: `getcursor characterstart start characterend end`
`if start = end then say 'No text selected.'`

Command Format
 Article ID numbers
 DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.45 `getcursorcoun`

GETCURSORCOUNT

External macros only!

Purpose: Gets the number of text selections. The user can select multiple blocks of text; this returns how many blocks exist.

Syntax: `getcursorcoun` [DOCUMENT name/S]

Format: Parameter Values to enter
 DOCUMENT is the document name. (Default=current article)

Result: The number of text selections is returned to the RESULT variable.

Notes: A result of 0 indicates that the insertion point is not placed.

Example: `getcursorcoun /* returns the number of text selections */`

cursors=RESULT

Command Format

Article ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.46 getdimensions

GETDIMENSIONS

External macros only!

Purpose: Gets the dimensions of a master page.

Syntax: `getdimensions <stem/V> [MASTERPAGE name/S]`

Format: Parameter Values to enter
 stem is the name of a stem variable for the dimensions.
 MASTERPAGE is the master page name. (Default=current)

Result: stem.width width
 stem.height height
 stem.orientation PORTRAIT|LANDSCAPE
 stem.sides SINGLE|DOUBLE

Example: `getdimensions size /* will print the mpage size to the output console ↔`
`*/`
`say 'Width: '||size.width||' Height: '||size.height`

Command Format

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.47 getdisplay

GETDISPLAY

Purpose: Gets the display settings.

Syntax: `getdisplay <stem/V> [WINDOW name/S]`

Format: Parameter Values to enter
 stem is the name of a stem variable for the settings.
 WINDOW is the window name. (Default=current)

Result: stem.page is the page number.
 stem.mode is the view magnification mode.
 stem.scale is the view magnification scale.
 stem.left is the horizontal page offset from the top left of the window.
 stem.top is the vertical page offset from the top left of the window.

Notes: stem.mode will be equal to <CUSTOM | FULLPAGE | FULLBLEED | FULLPAGEWIDTH | FULLBLEEDWIDTH | FULLPAGEHEIGHT | FULLBLEEDHEIGHT>.

```
Example: getdisplay displaystuff /* prints the display information to the output ↔
console */
say 'Page: ' ||displaystuff.page
say 'Mode: ' ||displaystuff.mode
say 'Scale: ' ||displaystuff.scale
say 'Left: ' ||displaystuff.left
say 'Top: ' ||displaystuff.top
```

Command Format

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.48 getdocumentdesc

GETDOCUMENTDESC

External macros only!

Purpose: Gets the description of a document.

Syntax: getdocumentdesc [DOCUMENT name/S]

Format: Parameter Values to enter
DOCUMENT is the document name. (Default=current)

Result: The description is returned to RESULT.

```
Example: getdocumentdesc document 'Project.doc' /* will return the description ↔
of the specified document */
docdesc=result
```

Command Format

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.49 getdocuments

GETDOCUMENTS

External macros only!

Purpose: Gets the number of open documents and their names.

Syntax: getdocuments <stem/V>

Format: Parameter Values to enter
stem is the name of a stem variable for the document names.

Result: The number of open documents is returned to RESULT.

The names of the open documents are returned to stem.# where #

is a number from 0 to the number of open documents less 1.

```
Example: getdocuments docnames /* will print the document names to the output ←
console */
numdocs=result
do count=0 to numdocs-1
    say docnames.count
end count
```

Command Format

1.50 getdocumentstatus

GETDOCUMENTSTATUS

External macros only!

Purpose: Gets the change status of a document.

Syntax: getdocumentstatus [DOCUMENT name/S]

Format: Parameter Values to enter
DOCUMENT is the document name. (Default=current)

Result: CHANGE or UNCHANGED is returned to RESULT.

```
Example: getdocumentstatus document 'Project.doc' /* will return the status of ←
the specified document */
status=result
```

Command Format

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.51 getdrawing

GETDRAWING

External macros only!

Purpose: Gets coordinates and information for a drawing.

Syntax: getdrawing [POSITION pstem/V] [FRAME fflag/V]
[CONTENTOFFSET cstem/V] [CONTENTSCALE cstem/V] [ROTATION rstem/V]
[ABOUT rstem/V] [CONSTRAIN cflag/V] [PRINT pflag/V]
[FILEINFO fstem/V] [DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]

Format: Parameter Values to enter
POSITION gets the coordinates of the frame.
CONTENTOFFSET gets the offset in the frame.
CONTENTSCALE gets the scale of the object in the frame.
ROTATION gets the rotation of the frame.
ABOUT gets the rotation point.


```

CONSTRRAIN    gets the proportional scale flag state.
PRINT         gets the print flag state.
FILEINFO     gets the file status of the drawing.
DOCUMENT      is the document name.
WINDOW       is the window name.
OBJECTID     is the number of the object. (Default=current)

```

Result: The object ID is returned to the RESULT variable.

```

POSITION:
pstem.left    left coordinate
pstem.top     top coordinate
pstem.right   right coordinate
pstem.bottom  bottom coordinate

FRAME: returns <ON|OFF>

CONTENTOFFSET:
cstem.x       horizontal offset
cstem.y       vertical offset

CONTENTSCALE:
cstem.h       horizontal scale
cstem.v       vertical scale

ROTATION:
rstem.mode    rotate about <POINT|CENTER>
rstem.slant   slant angle
rstem.twist   twist angle

ABOUT:
rstem.x       horizontal point
rstem.y       vertical point

CONSTRRAIN: returns <ON|OFF>

PRINT: returns <ON|OFF>

FILEINFO:
fstem.mode    <INTERNAL|EXTERNAL>
fstem.file    filepath and name

```

```

Example: getdrawing position coord /* will print the drawing bounding box to the ←
output console */
say 'Left: ' || coord.left
say 'Top: ' || coord.top
say 'Right: ' || coord.right
say 'Bottom: ' || coord.bottom

```

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.52 getdrawingdisplay

GETDRAWINGDISPLAY

External macros only!

Purpose: Gets the drawing display status.

Syntax: `getdrawingdisplay [WINDOW name/S]`

Format: Parameter Values to enter
 WINDOW is the window name. (Default=current)

Result: Returns the drawing display status <ON|OFF> to RESULT.

Example: `getdrawingdisplay /* will print the drawing display status to the
 output console */
 say 'Drawing Display: ' || result`

Command Format

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE

1.53 getellipse

GETELLIPSE

External macros only!

Purpose: Gets coordinates and information for an ellipse.

Syntax: `getellipse [POSITION pstem/V] [ANGLES astem/V]
 [ROTATION rstem/V] [ABOUT rstem/V] [CONSTRAIN cflag/V]
 [PRINT pflag/V] [DOCUMENT name/S | WINDOW name/S | OBJECTID number/I]`

Format: Parameter Values to enter
 POSITION gets the coordinates of the ellipse.
 ANGLES gets the ellipse angles and type.
 ROTATION gets the rotation of the ellipse.
 ABOUT gets the rotation point.
 CONSTRAIN gets the proportional scale flag state.
 PRINT gets the print flag state.
 DOCUMENT is the document name.
 WINDOW is the window name.
 OBJECTID is the number of the object. (Default=current)

Result: The object ID is returned to the RESULT variable.

POSITION:

`pstem.centerx` horizontal center coordinate
`pstem.centery` vertical center coordinate
`pstem.radiusx` horizontal radius
`pstem.radiusy` vertical radius

ANGLES:

`astem.type` <ELLIPSE|PIE|ARC>
`astem.begin` beginning angle

astem.end ending angle

ROTATION:

rstem.mode rotate about <POINT|CENTER>

rstem.slant slant angle

rstem.twist twist angle

ABOUT:

rstem.x horizontal point

rstem.y vertical point

CONSTRAIN: returns <ON|OFF>

PRINT: returns <ON|OFF>

Example: getellipse position coord /* will print the ellipse center coordinates ↔
to the output console */

say 'Horizontal Center: '||coord.centerx

say 'Vertical Center: '||coord.centery

Command Format

Object ID numbers

DOCUMENT, CHAPTER, WINDOW, PAGE, MASTERPAGE, MPG, STYLETAG & ARTICLE