

Arexx_PerfectPaint

Georges HALVADJIAN"

Copyright © Copyright 1998-1999 Georges HALVADJIAN"

COLLABORATORS

	<i>TITLE :</i> Arexx_PerfectPaint		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Georges HALVADJIAN"	July 10, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Arexx_PerfectPaint	1
1.1	Arexx Commands	1
1.2	Arexx Commands by Category	7
1.3	Inverse	13
1.4	New	13
1.5	Rot90	13
1.6	pp_Rotate	14
1.7	FlipBrushX	14
1.8	FlipBrushY	14
1.9	FlipX	14
1.10	pp_AliasOff	14
1.11	pp_AliasOn	14
1.12	pp_GetAlias	14
1.13	pp_GetPenSize	15
1.14	GetSizeText	15
1.15	pp_GetPenType	15
1.16	FlipY	15
1.17	Cycle	15
1.18	Integer	16
1.19	CheckBox	17
1.20	Button	17
1.21	Slider	18
1.22	String	19
1.23	Dialog Example	20
1.24	Getfile	20
1.25	Text	21
1.26	DialogInit	22
1.27	Dialog	23
1.28	GetDialog	23
1.29	False Colours	24

1.30	Transparency	24
1.31	Displace	24
1.32	Greyscale	24
1.33	Threshold	24
1.34	Spray	25
1.35	Airbrush	25
1.36	NextFrame	25
1.37	PreviousFrame	25
1.38	GotoFrame	25
1.39	CountFrames	26
1.40	MakeAnim	26
1.41	GetApen	26
1.42	GetBpen	26
1.43	Plot	26
1.44	MultiRequest	26
1.45	LongRequest	27
1.46	AddPoly	27
1.47	Example	27
1.48	Asay	28
1.49	AskFile	28
1.50	Box	28
1.51	BoxF	28
1.52	Circle	28
1.53	CircleF	29
1.54	CloseAsay	29
1.55	ClosestColor	29
1.56	ClosestColorL	29
1.57	ClosestColorD	29
1.58	Cls	30
1.59	EffectOn	30
1.60	EffectOff	30
1.61	Ellipse	30
1.62	EllipseF	30
1.63	pp_BrushEffect	31
1.64	pp_ClearRange	31
1.65	pp_FishEyesLens	31
1.66	Get color value	31
1.67	pp_Gradient	32
1.68	pp_RestoreBrush	32

1.69	pp_Ripple	32
1.70	pp_SetRange	32
1.71	Gradient example	32
1.72	pp_Sin	33
1.73	pp_Straw	33
1.74	pp_Twirl	33
1.75	pp_Wave	34
1.76	EndPoly	34
1.77	EndPolyS	34
1.78	EndPolySF	34
1.79	EndPoint	34
1.80	EndPolyF	35
1.81	Fill	35
1.82	GetDepth	35
1.83	GetHeight	35
1.84	GetWidth	35
1.85	GetDepthB	35
1.86	GetHeightB	36
1.87	GetWidthB	36
1.88	Light	36
1.89	Line	36
1.90	Load	36
1.91	Bload	36
1.92	MaskOn	37
1.93	MaskOff	37
1.94	PenSize	37
1.95	PenType	37
1.96	Refresh	37
1.97	Request	37
1.98	Save	38
1.99	Bsave	38
1.100	Psave	38
1.101	SetApen	38
1.102	SetBpen	38
1.103	Shade	39
1.104	ShowScreen	39
1.105	SareOnOff	39
1.106	Spline	39
1.107	SetBrush	39

1.108Pload	39
1.109SplineF	39
1.110StartPoly	40
1.111UpdateUndo	40
1.112Undo	40
1.113Warn	40
1.114Warp	40
1.115PicttoSpare	41
1.116SparetoPict	41
1.117Spare	41
1.118TextAntialiasing	41
1.119TextColorMake	41
1.120TextSize	42
1.121TextStyle	42
1.122TextMake	42
1.123TextDraw	42
1.124TextEffect	43
1.125TextFontName	43
1.126Text example	43
1.127TextRotate	44
1.128TextItalic	44
1.129TextXDPI	44
1.130TextYDPI	44
1.131Askmultifile	45
1.132BrushOpacity	45
1.133Compose	45
1.134ComposeReqOff	45
1.135ComposeReqOn	46
1.136ConvInit	46
1.137Conv	46
1.138FreeBrush	46
1.139GetWidthT	47
1.140GetHeightT	47
1.141PickBrush	47
1.142Render	47
1.143RenderReq	47
1.144SavePrefs	48
1.145ScaleBrush	48
1.146UpdateUndoBox	48
1.147TextForceType	48

Chapter 1

Arexx_PerfectPaint

1.1 Arexx Commands

IMPORTANT: Do not use spaces in arguments to Arexx functions. Use `↔`
`"*"` instead.

Example: `pp_Warn 'Make*an*anim*first.'`
and not
`pp_Warn 'Make an Anim first.'`

AREXX COMMANDS IN ALPHABETICAL ORDER

Arexx commands by category

A

`pp_AddPoly`
`pp_Airbrush`
`pp_AliasOff`
`pp_AliasOn`
`pp_Asay`
`pp_AskFile`
`pp_AskMultiFile`

B

`pp_Box`
`pp_BoxF`
`pp_Bload`

pp_BrushEffect
V2

pp_BrushOpacity

pp_Bsave

pp_Button
C

pp_CheckBox

pp_Circle

pp_CircleF

pp_ClearRange
V2

pp_CloseAsay

pp_ClosestColor

pp_ClosestColorD

pp_ClosestColorL

pp_Cls

pp_Compose

pp_ComposeReqOff

pp_ComposeReqOn

pp_Conv

pp_ConvInit

pp_CountFrames

pp_Cycle
D

pp_Dialog

pp_DialogInit

pp_Displace
E

pp_EffectOff

pp_EffectOn

pp_Ellipse

pp_EllipseF
pp_EndPoint
pp_EndPoly
pp_EndPolyF
pp_EndPolyS
pp_EndPolySF
F
pp_False
pp_Fill
pp_FishEyesLens
V2
pp_FlipX
pp_FlipY
pp_FlipBrushX
pp_FlipBrushY
pp_FreeBrush
G
pp_GetAlias
pp_GetApen
pp_GetBlueApen
V2
pp_GetBlueBpen
V2
pp_GetBpen
pp_GetDepth
pp_GetDepthB
pp_GetDialog
pp_GetFile
pp_GetGreenApen
V2
pp_GetGreenBpen
V2

pp_GetHeight
pp_GetHeightB
pp_GetHeightT
pp_GetJiffies
pp_GetPenSize
pp_GetPenType
pp_GetRedApen
V2
pp_GetRedBpen
V2
pp_GetSizeText
pp_GetWidth
pp_GetWidthB
pp_GetWidthT
pp_GotoFrame
pp_Gradient
V2
pp_Grey
I
pp_Inv
pp_Integer
L
pp_Light
pp_Line
pp_Load
pp_LongRequest
M
pp_MakeAnim
pp_MaskOff
pp_MaskOn
pp_MultiRequest
N

pp_NextFrame

pp_New
P

pp_PenSize

pp_PenType

pp_PicttoSpare

pp_PickBrush

pp_Pload

pp_Plot

pp_PreviousFrame

pp_Psave
R

pp_Refresh

pp_Render

pp_RenderReq

pp_Request

pp_RestoreBrush
V2

pp_Ripple
V2

pp_Rot90

pp_Rotate
S

pp_Save

pp_SavePrefs

pp_ScaleBrush

pp_SetApen

pp_SetBpen

pp_SetBrush
pp_SetJiffies

pp_SetRange
V2

pp_Shade
pp_ShowScreen
pp_Sin
V2
pp_Slider
pp_SparetoPict
pp_Spare
pp_SpareOnOff
pp_Spline
pp_SplineF
pp_Spray
pp_StartPoly
pp_Straw
V2
pp_String
T
pp_Text
pp_TextAntialiasing
pp_TextColorMake
pp_TextDraw
pp_TextEffect
pp_TextFontName
pp_TextForceType
pp_TextItalic
pp_TextMake
pp_TextRotate
pp_TextSize
pp_TextStyle
pp_TextXDPI
pp_TextYDPI

`pp_Threshold``pp_Trans``pp_Twirl
V2`

U

`pp_Undo``pp_UpdateUndo``pp_UpdateUndoBox
W``pp_Warn``pp_Warp``pp_Wave
V2`

1.2 Arexx Commands by Category

IMPORTANT: Do not use spaces in arguments to Arexx functions. Use `↔`
"*" instead.

Example: `pp_Warn 'Make*an*anim*first.'`
and not
`pp_Warn 'Make an Anim first.'`

AREXX COMMANDS BY CATEGORY

Arexx commands in alphabetical order

Drawing

`pp_AddPoly``pp_Box``pp_BoxF``pp_Circle``pp_CircleF`

pp_Cls
pp_Ellipse
pp_EllipseF
pp_EndPoint
pp_EndPoly
pp_EndPolyF
pp_EndPolyS
pp_EndPolySF
pp_Fill
pp_FreeBrush
pp_GetApen
pp_GetBlueApen
V2
pp_GetBlueBpen
V2
pp_GetBpen
pp_GetDepth
pp_GetDepthB
pp_GetGreenApen
V2
pp_GetGreenBpen
V2
pp_GetHeight
pp_GetHeightB
pp_GetHeightT
pp_GetPenSize
pp_GetPenType
pp_GetWidth
pp_GetWidthB
pp_GetWidthT
pp_Line

pp_PenSize
pp_PenType
pp_PicttoSpare
pp_PickBrush
pp_Plot
pp_GetRedApen
V2
pp_GetRedBpen
V2
pp_SetApen
pp_SetBpen
pp_SetBrush
pp_SparetoPict
pp_Spline
pp_SplineF
pp_StartPoly
Text
pp_TextAntialiasing
pp_TextColorMake
pp_TextDraw
pp_TextEffect
pp_TextFontName
pp_TextForceType
pp_TextItalic
pp_TextMake
pp_TextRotate
pp_GetSizeText
pp_TextSize
pp_TextStyle
pp_TextXDPI

pp_TextYDPI
GUI

pp_Asay

pp_AskFile

pp_AskMultiFile

pp_Button

pp_CheckBox

pp_CloseAsay

pp_Cycle

pp_Dialog

pp_DialogInit

pp_GetDialog

pp_GetFile

pp_Integer

pp_LongRequest

pp_MultiRequest

pp_Request

pp_Slider

pp_String

pp_Text

pp_Warn
File

pp_Bload

pp_Bsave

pp_Load

pp_New

pp_Pload

pp_Psave

pp_Render

pp_RenderReq
pp_Save
pp_SavePrefs
Misc
pp_AliasOff
pp_AliasOn
pp_BrushEffect
V2
pp_BrushOpacity
pp_ClosestColor
pp_ClosestColorD
pp_ClosestColorL
pp_FlipX
pp_FlipY
pp_FlipBrushX
pp_FlipBrushY
pp_RestoreBrush
V2
pp_GetAlias
pp_MaskOff
pp_MaskOn
pp_PicttoSpare
pp_Refresh
pp_Rot90
pp_ScaleBrush
pp_ShowScreen
pp_SpareOnOff
pp_SparetoPict
pp_Undo
pp_UpdateUndo

pp_UpdateUndoBox
Animation

pp_CountFrames
pp_GetJiffies

pp_GotoFrame

pp_MakeAnim

pp_NextFrame

pp_PreviousFrame
pp_SetJiffies

Effects

pp_Airbrush

pp_ClearRange
V2

pp_Compose

pp_ComposeReqOff

pp_ComposeReqOn

pp_Conv

pp_ConvInit

pp_Displace

pp_EffectOff

pp_EffectOn

pp_False

pp_FishEyesLens
V2

pp_Gradient
V2

pp_Grey

pp_Inv

pp_Light

pp_Ripple
V2

pp_Rotate

```
pp_SetRange
V2

pp_Shade

pp_Sin
V2

pp_Spare

pp_Spray

pp_Straw
V2

pp_Threshold

pp_Trans

pp_Twirl
V2

pp_Warp

pp_Wave
V2
```

1.3 Inverse

```
pp_Inv

Activates the colour inversion effect
```

1.4 New

```
pp_New width height bits

Creates a new picture of size width x height x bits
```

1.5 Rot90

```
pp_Rot90

Rotates the current picture by 90\textdegree{}
```

1.6 pp_Rotate

pp_Rotate x

Activates the rotation effect

x: (-180\textdegree{};180\textdegree{})

1.7 FlipBrushX

pp_FlipBrushX

Flips the brush about the X-axis

1.8 FlipBrushY

pp_FlipBrushY

Flips the brush about the Y-axis

1.9 FlipX

pp_FlipX

Flips the current picture about the X-axis

1.10 pp_AliasOff

pp_AliasOff

Turns off antialiasing

1.11 pp_AliasOn

pp_AliasOn

Turns on antialiasing

1.12 pp_GetAlias

pp_GetAlias

Returns 1 if antialiasing is active, 0 otherwise

1.13 pp_GetPenSize

pp_GetPenSize

Returns the size of the current pen

1.14 GetSizeText

pp_GetSizeText

Returns the current font size.

1.15 pp_GetPenType

pp_GetPenType

Returns the current pen type

1.16 FlipY

pp_FlipY

Flips the current picture about the Y-axis

1.17 Cycle

```
pp_Cycle N\textdegree{} x y w h text$ flags options default
```

Initialises a cycle gadget

N\textdegree{}: gadget number (max 10)
x: X position
y: Y position
w: gadget width
h: gadget height
text\$: gadget title
flags: flags
options
default

Dialog example
See also:

pp_CheckBox

pp_Cycle
pp_Dialog
pp_DialogInit
pp_GetDialog
pp_GetFile
pp_Integer
pp_Slider
pp_String
pp_Text

1.18 Integer

```
pp_Integer N\textdegree{} x y w h text$ flags default
```

Initialises an integer gadget

N\textdegree{}: gadget number (max 10)
x: X position
y: Y position
w: gadget width
h: gadget height
text\$: gadget title
flags: flags
default

Dialog example
See also:

pp_CheckBox
pp_Cycle
pp_Dialog
pp_DialogInit
pp_GetDialog
pp_GetFile
pp_Integer
pp_Slider

pp_String

pp_Text

1.19 CheckBox

```
pp_CheckBox N\textdegree{} x y w h text$ flags
```

Initialises a checkbox gadget

N\textdegree{}: gadget number (max 10)
x: X position
y: Y position
w: gadget width
h: gadget height
text\$: gadget title
flags: 0 ou 1 (disactivated/activated)

Dialog example
See also:

pp_CheckBox

pp_Cycle

pp_Dialog

pp_DialogInit

pp_GetDialog

pp_GetFile

pp_Integer

pp_Slider

pp_String

pp_Text

1.20 Button

```
pp_Button N\textdegree{} x y w h text$
```

Initialises a button gadget

N\textdegree{}: gadget number (max 10)

x: X position
y: Y position
w: gadget width
h: gadget height
text\$: gadget title

Dialog example
See also:

pp_CheckBox

pp_Cycle

pp_Dialog

pp_DialogInit

pp_GetDialog

pp_GetFile

pp_Integer

pp_Slider

pp_String

pp_Text

1.21 Slider

```
pp_Slider N\textdegree{} x y w h text$ flags min max default
```

Initialises a slider gadget

N\textdegree{}: gadget number (max 10)
x: X position
y: Y position
w: gadget width
h: gadget height
text\$: gadget title
flags: flags
min: minimum value
max: maximum value
default

Dialog example
See also:

pp_CheckBox

pp_Cycle
pp_Dialog
pp_DialogInit
pp_GetDialog
pp_GetFile
pp_Integer
pp_Slider
pp_String
pp_Text

1.22 String

```
pp_String N\textdegree{} x y w h text$ flags max default
```

Initialises a string gadget

N\textdegree{}: gadget number (max 10)
x: X position
y: Y position
w: gadget width
h: gadget height
text\$: gadget title
flags: flags
max: maximum string length
default

Dialog example
See also:

pp_CheckBox
pp_Cycle
pp_Dialog
pp_DialogInit
pp_GetDialog
pp_GetFile
pp_Integer

```

        pp_Slider

        pp_String

        pp_Text

```

1.23 Dialog Example

```

/**/

options results
parse ARG Port b

ADDRESS value Port

/* Initialise a dialog box consisting of 2 gadgets */
pp_DialogInit 320 200 "*TEST*" 2

/* Initialise the slider gadget */
pp_Slider 0 100 5 100 16 "Slider" 1 10 200 50

/* Initialise the cycle gadget */
pp_Cycle 1 100 25 100 16 "Cycle" 1 "1|2|3|4|5|6" 2

/* Open the dialog box */
pp_Dialog

rc=result
if rc=0 then          /* the "Cancel" button has been selected, so */
do                   /* quit the script */
    EXIT
end

pp_GetDialog 0        /* find the value of gadget 0 */
S=result

pp_GetDialog 1        /* find the value of gadget 1 */
C=result

/* ..... */

```

1.24 Getfile

```
pp_Getfile N\textdegree{} x y gt title$ type path$
```

Initialises a getfile gadget

```

N\textdegree{}: gadget number (max 10)
x: X position
y: Y position

```

gt: number of associated string gadget
title\$
type: 0=File requester
 1=Path requester
path\$: default path

Dialog example
See also:

pp_CheckBox

pp_Cycle

pp_Dialog

pp_DialogInit

pp_GetDialog

pp_GetFile

pp_Integer

pp_Slider

pp_String

pp_Text

1.25 Text

pp_Text N\textdegree{} x y text\$

Initialises a text gadget

N\textdegree{}: gadget number (max 10)
x: X position
y: Y position
text\$: gadget title

Dialog example
See also:

pp_CheckBox

pp_Cycle

pp_Dialog

pp_DialogInit

```
pp_GetDialog  
pp_GetFile  
pp_Integer  
pp_Slider  
pp_String  
pp_Text
```

1.26 DialogInit

```
pp_DialogInit w h title$ Nb
```

Initialises a dialog box
Two buttons will be added, "Ok" and "Cancel", as well as
PerfectPaint's own design

w: box width
h: box height
title\$
Nb: gadget number to initialise

```
Dialog Example  
See also:
```

```
pp_CheckBox  
pp_Cycle  
pp_Dialog  
pp_DialogInit  
pp_GetDialog  
pp_GetFile  
pp_Integer  
pp_Slider  
pp_String  
pp_Text
```

1.27 Dialog

pp_Dialog

Opens the dialog window along with all the initialised gadgets

Values returned:

- 0: 'Cancel' has been pressed
- 1: 'Ok' has been pressed

Dialog Example

See also:

pp_CheckBox

pp_Cycle

pp_Dialog

pp_DialogInit

pp_GetDialog

pp_GetFile

pp_Integer

pp_Slider

pp_String

pp_Text

1.28 GetDialog

pp_GetDialog gt

Returns the value of the gadget numbered gt

Dialog Example

See also:

pp_CheckBox

pp_Cycle

pp_Dialog

pp_DialogInit

pp_GetDialog

pp_GetFile

pp_Integer

pp_Slider

pp_String

pp_Text

1.29 False Colours

pp_False

Activates the false colours effect

1.30 Transparency

pp_Trans x

Activates the transparency effect

x: 0-100%

1.31 Displace

pp_Displace x

Activates the Displace effect

x: 0-20

1.32 Greyscale

pp_Grey

Activates the grey effect

1.33 Threshold

pp_Threshold x

Activates the threshold effect

x: 0-255

1.34 Spray

pp_Spray Colour x

Activates the spray effect

Colour: 0 current colour
1 Mixe between Fg and Bg
2 spare page

x: (Amplitude) 1-100%

1.35 Airbrush

pp_Airbrush Opacity Flow

Activates the airbrush effect

Opacity: 0-100%
Flow: 0-30

1.36 NextFrame

pp_NextFrame

Moves to the following frame
(if an animation has been created)

1.37 PreviousFrame

pp_PreviousFrame

Moves to the previous frame
(if an animation has been created)

1.38 GotoFrame

pp_GotoFrame n

Moves to frame number n
(if an animation has been created)

1.39 CountFrames

pp_CountFrames

Returns the number of frames in the current animation
0: no animation has been created

1.40 MakeAnim

pp_MakeAnim n

Creates an animation of n frames
If an animation is already present, it will be destroyed.

1.41 GetApen

pp_GetApen

Returns the number of the current foreground colour

1.42 GetBpen

pp_GetBpen

Returns the number of the current background colour

1.43 Plot

pp_Plot x y

Places the current object at the position (x,y) on the current picture (brush, pen etc...)

1.44 MultiRequest

pp_MultiRequest n1\$ n2\$ n3\$

Opens a requester in the centre of the screen
This requester may contain one or more buttons and will return the number of the selected button

n1\$: requester title
n2\$: body text
n3\$: button(s) separated by "|"

The rightmost button will always return the value 0 (it is normally used for 'Cancel'). The numbering of the other buttons is as usual (1,2,...) from left to right.

1.45 LongRequest

```
pp_LongRequest n1$ n2$
```

Opens a requester in the centre of the screen
This requester will wait until a number has been entered

n1\$: requester title

n2\$: body text

1.46 AddPoly

```
pp_AddPoly x/N y/N
```

Adds a point (x,y) in the construction of a polygon

Example

1.47 Example

```
pp_StartPoly

pp_AddPoly
  x y

pp_AddPoly
  x1 y1
...

pp_EndPoly
  /* for an empty polygon */
ou

pp_EndPolyF
  /* for a filled polygon */
ou

PP_EndPolyS
  /* Bezier algo */
ou

PP_EndPolyS
```

```
        /* Bezier algo with filling */  
ou  
    PP_EndPoint  
        /* just draw the points */
```

1.48 Asay

```
pp_Asay message$
```

Opens an asynchronous message window
This window must be closed with
pp_CloseAsay

1.49 AskFile

```
pp_AskFile name$
```

Opens a file requester and returns the name of the selected file

1.50 Box

```
pp_Box x y x1 y1
```

Draws an empty box whose top-left corner is at (x,y) and bottom-right corner is at (x1,y1)

Example: pp_Box 10 10 150 30

1.51 BoxF

```
pp_BoxF x y x1 y1
```

Draws a filled box whose top-left corner is at (x,y) and bottom-right corner is at (x1,y1)

Example: pp_BoxF 10 10 150 30

1.52 Circle

```
pp_Circle x y r
```

Draws an empty circle

x,y: co-ordinates of the circle's centre
r: circle radius

1.53 CircleF

```
pp_CircleF x y r
```

Draws a filled circle

x,y: co-ordinates of the circle's centre
r: circle radius

1.54 CloseAsay

```
pp_CloseAsay
```

Closes an asynchronous message window

1.55 ClosestColor

```
pp_ClosestColor r g b
```

Returns the number of the colour in the current palette which most closely matches the supplied rgb values

r,g,b must be between 0 and 255.

1.56 ClosestColorL

```
pp_ClosestColorL r d
```

r: number of the colour
d: darkness (0-100%)

Returns the number of the colour in the current palette which most closely matches the supplied colour r with the supplied darkness.

1.57 ClosestColorD

```
pp_ClosestColorD r b
```

```
r: number of the colour  
d: brightness (0-100%)
```

Returns the number of the colour in the current palette which most closely matches the supplied colour r with the supplied brightness.

1.58 Cls

```
pp_Cls
```

Clears the current picture (takes the current colour and stencil into account)

1.59 EffectOn

```
pp_EffectOn
```

Activates 'Effect' mode

1.60 EffectOff

```
pp_EffectOff
```

Disactivates 'Effect' mode

1.61 Ellipse

```
pp_Ellipse x y r1 r2
```

Draws an empty ellipse

```
x,y:Co-ordinates of ellipse centre  
r1: x-radius of ellipse  
r2: y-radius of ellipse
```

1.62 EllipseF

```
pp_EllipseF x y r1 r2
```

Draws a filled ellipse

```
x,y:Co-ordinates of ellipse centre  
r1: x-radius of ellipse  
r2: y-radius of ellipse
```

1.63 pp_BrushEffect

pp_BrushEffect type x y z A

type: 0 perspective/Rotation
1 Shear x
2 Shear y

x, y, z :rotation

A: 0 no antialiasing
1 Low antialiasing
2 high antialiasing

1.64 pp_ClearRange

pp_ClearRange r

r: Number of the range (1-3)

Clear Range number r

Gradient example

1.65 pp_FishEyesLens

pp_FishEyesLens x

Activate the Fish eyes lens effect

x: -60,60

1.66 Get color value

pp_GetRedApen
pp_GetGreenApen
pp_GetBlueApen

pp_GetRedBpen
pp_GetGreenBpen
pp_GetBlueBpen

Return the Red, Blue or green value of the foreground (Apen) or Background (Bpen) color.

1.67 pp_Gradient

```
pp_Gradient Nb Type x y
```

Nb: Number of the range

Type: 0 Horizontal
1 Vertical
2 Radial Sun
3 Warp H
4 Warp V
5 Radial Warp

x and y: center of the radial effect

Gradient example

1.68 pp_RestoreBrush

```
pp_RestoreBrush
```

1.69 pp_Ripple

```
pp_Ripple x
```

Activate the Ripple effect

x: 1-50

1.70 pp_SetRange

```
pp_SetRange r p Red Green Blue
```

Make the range number r (0-2) and place the RGB color on position p (0-19)

Gradient example

1.71 Gradient example

```
pp_EffectOn  
/* Activate Effect mode */  
  
pp_Gradient 0 2 100 100
```

```
/* Activate Gradient effect
   range=0
   Type=2 (Radian sun)
   100, 100: center of the sun */

pp_ClearRange 0
/* Clear Range=0 */

pp_setRange 0 4 0 0 0
/* Fill range 0
   at position=4, put rgb color (0,0,0) */

pp_setRange 0 19 255 255 255
/* Fill range 0
   at position=19, put rgb color (255,255,255) */

pp_CircleF 100 100 100
/* Draw a circle */
```

1.72 pp_Sin

```
pp_Sin Ay Fx Ax Fy

Activate the Sin effect

Ay (-200,200)
Fx (100-3000)

Ax (-200,200)
Fy (100-3000)
```

1.73 pp_Straw

```
pp_Straw a b

Activate the Straw effect

a:(1-50) Space between Straw
b:(2-50) Size of the Straw
```

1.74 pp_Twirl

```
pp_Twirl a

Activate the Twirl effect

a:(-720,720) angle
```

1.75 pp_Wave

pp_Wave a n

Activate the Wave effect

a: (-300,300) Amplitude

b: (1-300) number of waves

1.76 EndPoly

pp_EndPoly

Creates an empty polygon

Example

1.77 EndPolyS

pp_EndPolyS

Creates an empty polygon with a bezier algo.

Example

1.78 EndPolySF

pp_EndPolySF

Creates a filled polygon with a bezier algo.

Example

1.79 EndPoint

pp_EndPoint

just draw the points.

Example

1.80 EndPolyF

pp_EndPolyF

Creates a filled polygon

Example

1.81 Fill

pp_Fill x y

Flood fill starting from (x,y)

1.82 GetDepth

pp_GetDepth

Returns the depth of the current picture
0: No picture in memory

1.83 GetHeight

pp_GetHeight

Returns the height of the current picture
0: No picture in memory

1.84 GetWidth

pp_GetWidth

Returns the width of the current picture
0: No picture in memory

1.85 GetDepthB

pp_GetDepthB

Returns the depth of the current brush
0: No picture in memory

1.86 GetHeightB

```
pp_GetHeightB
```

Returns the height of the current brush
0: No picture in memory

1.87 GetWidthB

```
pp_GetWidthB
```

Returns the width of the current brush
0: No picture in memory

1.88 Light

```
pp_Light x
```

Activates the light effect
x: 0-100%

1.89 Line

```
pp_Line x y x1 y1
```

Draws a line from A(x,y) to B(x1,y1)

1.90 Load

```
pp_Load name$
```

Loads a picture
name\$: name of the picture, with path

1.91 Bload

```
pp_Bload name$
```

Loads a brush
name\$: name of the brush, with path

1.92 MaskOn

pp_MaskOn

Activates 'Mask' mode

1.93 MaskOff

pp_MaskOff

Disactivates 'Mask' mode

1.94 PenSize

pp_PenSize n

Returns the current pen size

1.95 PenType

pp_PenType n

Selects the type of pen

0: single point
1: filled circle
2: filled square
3: AntiSlash
4: reserved
5: reserved
6: brush

1.96 Refresh

pp_Refresh

Refreshes the window containing the current image

1.97 Request

pp_Request message\$

Opens a message window containing two buttons: 'Yes' and 'No', returning 1 if 'Yes' is selected and 0 otherwise.

1.98 Save

```
pp_Save name$ a
```

Saves the current picture

```
a: 0=save without icon
    1=save with icon
```

1.99 Bsave

```
pp_Bsave name$ a b
```

Saves the current brush

```
a: 0=save without icon
    1=save with icon
```

```
b: 0 1-8Bits
    1 24Bits without Mask
    2 24Bits with Mask
    3 Deep brush (with alpha-channel)
```

1.100 Psave

```
pp_Psave name$ a
```

Saves the current palette

```
a: 0=save without icon
    1=save with icon
```

1.101 SetApen

```
pp_SetApen a
```

Sets the foreground colour to a

1.102 SetBpen

```
pp_SetBpen a
```

Sets the background colour to a

1.103 Shade

```
pp_Shade x
```

Activates the shade effect

x: 0-100%

1.104 ShowScreen

```
pp_ShowScreen
```

Brings the PerfectPaint screen to the front

1.105 SpareOnOff

```
pp_SpareOnOff
```

Activates or disactivates the spare page

1.106 Spline

```
pp_Spline x y x1 y1 x2 y2
```

Draws an empty curve from A(x,y) to B(x1,y1), passing through C(x2,y2)

1.107 SetBrush

```
pp_SetBrush n
```

Activates brush number n

n: (1-2)

1.108 Pload

```
pp_Pload name$
```

Loads the palette name\$

1.109 SplineF

```
pp_SplineF x y x1 y1 x2 y2
```

Draws a filled curve from A(x,y) to B(x1,y1), passing through C(x2,y2)

1.110 StartPoly

```
pp_StartPoly
```

Tell PerfectPaint that we're going to start drawing a polygon

Example

1.111 UpdateUndo

```
pp_UpdateUndo
```

Stores the current image in the undo buffer
(The undo facility is always deactivated during execution of Arexx scripts)

1.112 Undo

```
pp_Undo
```

Same as pressing 'undo'

1.113 Warn

```
pp_Warn message$
```

Opens an alert window with an 'OK' button

1.114 Warp

```
pp_Warp X Y r Dx Dy
```

Performs a warp on the current picture

X: centre x of warp

Y: centre y of warp

r: radius of warp

Dx: x offset from centre

Dy: y offset from centre

1.115 PicttoSpare

```
pp_PicttoSpare
```

Copy current picture over the spare page

1.116 SparetoPict

```
pp_SparetoPict
```

Copy page spare over the current picture.

1.117 Spare

```
pp_Spare
```

Activates the Spare effect.

1.118 TextAntialiasing

```
pp_TextAntialiasing x
```

```
x: 0 -> No antialiasing  
    1 -> Antialiasing low  
    2 -> Antialiasing High
```

Activates Antialiasing.

Text example

1.119 TextColorMake

```
pp_TextColorMake a$ R
```

```
a$: Texte, if a$="", current text is used  
R : 0 use palette font  
    1 Remap with the current palette  
    2 Colorize the font
```

Build a text brush with a color font.

Text example

1.120 TextSize

```
pp_TextSize x
```

x: font size

Text example

1.121 TextStyle

```
pp_TextStyle x
```

x: font style (B=bold,I=italique,U=underline)

0 -> B=0 I=0 U=0

1 -> B=0 I=0 U=1

2 -> B=1 I=0 U=0

3 -> B=1 I=0 U=1

4 -> B=0 I=1 U=0

5 -> B=0 I=1 U=1

6 -> B=1 I=1 U=0

7 -> B=1 I=1 U=1

Text example

1.122 TextMake

```
pp_TextMake a$
```

a\$: Text, if a\$="", current text is used

Make a text brush with a bitmap or outline font.

Text example

1.123 TextDraw

```
pp_TextDraw x y
```

draw text brush on the current picture.

Text example

1.124 TextEffect

```
pp_TextEffect Effect x

Effect: 0 No effect
        1 Outline
        2 Bevel

x: Amplitude

Text example
```

1.125 TextFontName

```
pp_TextFontName fonte$

fonte$: font name

Text example
```

1.126 Text example

```
/* example */

.....

pp_TextAntialiasing 2
/* antialiasing High */
/* without this command, current antialiasing is used */

pp_TextEffect 1 2
/* effect Outline */
/* without this command, current effect is used */

pp_TextFontName 'CGTimes.font'
/* without this command, current font is used */

pp_TextSize 50
/* without this command, current size is used */

pp_TextStyle 0
/* No Style */
/* without this command, current style is used */

pp_TextMake 'Hello*everybody'
/* Build text brush: Hello everybody */
/* with pp_TextMake "", current text is used */
```

```
pp_TextDraw 100 100
  /* draw text on the current picture */
```

.....

1.127 TextRotate

```
pp_TextRotate x
```

x: Rotation of the current font. (Outline font only)

Text example

1.128 TextItalic

```
pp_TextItalic x
```

x (-45\textdegree{ },45\textdegree{ }): Italicize the current font. (Outline font only) ↔

Text example

1.129 TextXDPI

```
pp_TextXDPI x
```

x : DPI (X) of the current font. (Outline font only)
default=72

Text example

1.130 TextYDPI

```
pp_TextYDPI x
```

x : DPI (Y) of the current font. (Outline font only)
default=72

Text example

1.131 Askmultifile

```
pp_AskMultiFile Title
  Title: Title of the requester
```

This command let you select multiple file.
It writes in ram: a file name 'MultiFile' which content all the selected file
This command return the number of file selected

ex:

```
pp_AskMultiFile 'Load*Pictures'
count=result

file="Ram:MultiFile"
IF OPEN("g",file,"read") then DO
  DO i=1 to count
    file=READLN("g")
    say file
  END
  CALL close(file)
END
```

1.132 BrushOpacity

```
pp_BrushOpacity x
x: 0-100%
```

1.133 Compose

```
pp_Compose Mode Opacity Alpha
  Mode: 0 Opacity
        1 Add
        2 Sub
        3 Darken
        4 Alpha
        5 Inv Alpha
  Opacity: 0-100%
  Alpha: -255 -> 255
```

This compose between the current effect and the current picture.

1.134 ComposeReqOff

With this command, the compose requester will not appear after each effect

1.135 ComposeReqOn

With this command, the compose requester will appear after each effect

1.136 ConvInit

```
pp_ConvInit Dim Div Bias
  Dim: 0 3x3
      1 5x5
```

This command initialise a convolution
Use pp_Conv to fill this convolution.

Ex:

```
C=-2
pp_ConvInit 0 1 0 ;initialise the convolution
```

```
pp_Conv 0 0 0 0 0 0 ;fill the convolution
pp_Conv 1 0 C 0 0 0
pp_Conv 2 0 0 5 0 0
pp_Conv 3 0 0 0 C 0
pp_Conv 4 0 0 0 0 0
```

```
pp_BoxF x1 y1 x2 y2 ; draw a box on the picture with the current convolution
```

1.137 Conv

```
pp_Conv y C1 C2 C3 C4 C5
  y: current line
  C1-C5: convolution
```

Use pp_Conv to initialise the convolution

Ex:

```
C=-2
pp_ConvInit 0 1 0 ;initialise the convolution
```

```
pp_Conv 0 0 0 0 0 0 ;fill the convolution
pp_Conv 1 0 C 0 0 0
pp_Conv 2 0 0 5 0 0
pp_Conv 3 0 0 0 C 0
pp_Conv 4 0 0 0 0 0
```

```
pp_BoxF x1 y1 x2 y2 ; draw a box on the picture with the current convolution
```

1.138 FreeBrush

pp_FreeBrush

1.139 GetWidthT

pp_GetWidthT

Return the current width of the text

1.140 GetHeightT

pp_GetHeightT

Return the current height of the text

1.141 PickBrush

pp_PickBrush x y width height

1.142 Render

pp_Render type file
Type: 0 Picture
1 Brush
2 Animation

file: name of the file to be save

This Command will reduce the number of color of your brush, picture or animation and will save it.

See also

pp_RenderReq

1.143 RenderReq

pp_RenderReq Type
Type: 0 Picture
1 Brush
2 Animation

Will open then Render requester
Return '0' if 'Cancel' was selected

See also

pp_Render

1.144 SavePrefs

```
pp_SavePrefs Format ILBMtype JpegQuality JpegSmooth JpegColor
```

```
Format: 0 iff24  
        1 Jpeg
```

```
ILBMtype: 0 Uncompressed  
          1 Compressed  
          2 8bit gray
```

```
JpegQuality 1-100%
```

```
JpegSmooth 0-100%
```

```
JpegColor: 0 RGB  
           1 Gray
```

Use this command before saving your picture with
pp_Save

1.145 ScaleBrush

```
pp_ScaleBrush Width Height
```

This command will scale the current brush.

1.146 UpdateUndoBox

```
pp_UpdateUndoBox x y x1 y1
```

Stores a part of the current image in the undo buffer
(The undo facility is always deactivated during execution of Arexx
scripts)

1.147 TextForceType

```
pp_TextForceType x
```

```
x -> 0 : Bitmap font  
x -> 2 : Color font  
x -> 3 : Outline font
```

Force PerfectPaint to use this type of font.

```
Text example
```
