

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 15, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Arexx_PerfectPaint	1
1.1	Arexx Commands	1
1.2	Arexx Commands by Category	8
1.3	GetCurrentBuffer	15
1.4	GetCurrentFrame	15
1.5	GetCurrentBrush	15
1.6	FindEmptyBuffer	15
1.7	FindEmptyBrush	15
1.8	SetBuffer	15
1.9	ClearCurrentBuffer	16
1.10	Inverse	16
1.11	New	16
1.12	Rot90	16
1.13	pp_Rotate	16
1.14	FlipBrushX	16
1.15	FlipBrushY	16
1.16	FlipX	17
1.17	pp_AliasOff	17
1.18	pp_AliasOn	17
1.19	pp_SparetoSten	17
1.20	pp_StentoSpare	17
1.21	pp_GetAlias	17
1.22	pp_GetPenSize	17
1.23	GetSizeText	18
1.24	pp_GetPenType	18
1.25	FlipY	18
1.26	Cycle	18
1.27	Integer	19
1.28	CheckBox	20
1.29	Button	20

1.30 Slider	21
1.31 String	22
1.32 Dialog Example	23
1.33 Gamma Correction	23
1.34 Color Balance	23
1.35 Getfile	24
1.36 Text	24
1.37 DialogInit	25
1.38 Dialog	26
1.39 GetDialog	26
1.40 False Colours	27
1.41 Transparency	27
1.42 Displace	27
1.43 Greyscale	27
1.44 Threshold	28
1.45 Spray	28
1.46 Airbrush	28
1.47 NextFrame	28
1.48 PreviousFrame	28
1.49 GotoFrame	29
1.50 CountFrames	29
1.51 MakeAnim	29
1.52 GetApen	29
1.53 GetBpen	29
1.54 Plot	29
1.55 MultiRequest	30
1.56 LongRequest	30
1.57 AddPoly	30
1.58 Example	30
1.59 Asay	31
1.60 AskFile	31
1.61 Box	31
1.62 BoxF	31
1.63 Circle	32
1.64 CircleF	32
1.65 CloseAsay	32
1.66 ClosestColor	32
1.67 ClosestColorL	32
1.68 ClosestColorD	33

1.69	Cls	33
1.70	EffectOn	33
1.71	EffectOff	33
1.72	Ellipse	33
1.73	EllipseF	34
1.74	pp_BrushEffect	34
1.75	pp_ClearRange	34
1.76	pp_FishEyesLens	34
1.77	Get color value	34
1.78	pp_Gradient	35
1.79	pp_RestoreBrush	35
1.80	pp_Ripple	35
1.81	pp_SetRange	35
1.82	Gradient example	36
1.83	pp_Sin	36
1.84	pp_Straw	36
1.85	pp_Twirl	37
1.86	pp_Wave	37
1.87	EndPoly	37
1.88	EndPolyS	37
1.89	EndPolySF	37
1.90	EndPoint	38
1.91	EndPolyF	38
1.92	Fill	38
1.93	GetDepth	38
1.94	GetHeight	38
1.95	GetWidth	39
1.96	GetDepthB	39
1.97	GetHeightB	39
1.98	GetWidthB	39
1.99	Light	39
1.100	Line	39
1.101	Load	40
1.102	Bload	40
1.103	MaskOn	40
1.104	MaskOff	40
1.105	PenSize	40
1.106	PenType	40
1.107	Refresh	41

1.108Request	41
1.109Save	41
1.110Bsave	41
1.111Psave	41
1.112SetApen	42
1.113SetBpen	42
1.114Shade	42
1.115ShowScreen	42
1.116SareOnOff	42
1.117Spline	42
1.118SetBrush	42
1.119Pload	43
1.120SplineF	43
1.121StartPoly	43
1.122UpdateUndo	43
1.123Undo	43
1.124Warn	43
1.125Warp	44
1.126PictoSpare	44
1.127SparetoPict	44
1.128Spare	44
1.129TextAntialiasing	44
1.130TextColorMake	45
1.131TextSize	45
1.132TextStyle	45
1.133TextMake	45
1.134TextDraw	46
1.135TextEffect	46
1.136TextFontName	46
1.137Text example	46
1.138TextRotate	47
1.139TextItalic	47
1.140TextXDPI	47
1.141TextYDPI	48
1.142Askmultifile	48
1.143BrushOpacity	48
1.144Compose	49
1.145ComposeReqOff	49
1.146ComposeReqOn	49

1.147ConvInit	49
1.148Conv	49
1.149FreeBrush	50
1.150GetWidthT	50
1.151GetHeightT	50
1.152PickBrush	50
1.153Render	50
1.154RenderReq	51
1.155SavePrefs	51
1.156ScaleBrush	51
1.157UpdateUndoBox	52
1.158LineArt	52
1.159GaussianBlur	52
1.160BumpMapping	52
1.161InnerBevel	53
1.162TextForceType	53

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_BumpMapping
V2.3

pp_Bsave

pp_Button
C

pp_CheckBox

pp_Circle

pp_CircleF

pp_ClearCurrentBuffer
V2.5

pp_ClearRange
V2

pp_CloseAsay

pp_ClosestColor

pp_ClosestColorD

pp_ClosestColorL

pp_Cls

pp_ColorBalance
V2.3

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_FindEmptyBuffer
V2.5

pp_FindEmptyBrush
V2.5

pp_FishEyesLens
V2

pp_FlipX

pp_FlipY

pp_FlipBrushX

pp_FlipBrushY

pp_FreeBrush
G

pp_Gamma
v2.3

pp_GaussianBlur
V2.3

pp_GetAlias

pp_GetApen

pp_GetBlueApen
V2

pp_GetBlueBpen
V2

pp_GetBpen

pp_GetCurrentBuffer
V2.5

pp_GetCurrentBrush
V2.5

pp_GetCurrentFrame
V2.5

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_InnerBevel
V2.3

pp_Inv

pp_Integer
L

pp_Light

pp_Line

pp_LineArt
V2.3

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_SetBuffer
V2.5
pp_SetJiffies
pp_SetRange
V2
pp_Shade
pp_ShowScreen
pp_Sin
V2
pp_Slider
pp_SparetoPict
pp_SparetoSten
V2.3
pp_Spare
pp_SpareOnOff

pp_Spline
pp_SplineF
pp_Spray
pp_StartPoly
pp_StenToSpare
V2.3
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_ClearCurrentBuffer
V2.5

pp_Cls

pp_ColorBalance
V2.3

pp_Ellipse

pp_EllipseF
```

pp_EndPoint

pp_EndPoly

pp_EndPolyF

pp_EndPolyS

pp_EndPolySF

pp_Fill

pp_FindEmptyBuffer
V2.5

pp_FindEmptyBrush
V2.5

pp_FreeBrush

pp_Gamma
v2.3

pp_GetApen

pp_GetBlueApen
V2

pp_GetBlueBpen
V2

pp_GetBpen

pp_GetCurrentBuffer
V2.5

pp_GetCurrentBrush
V2.5

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_SetBuffer
V2.5
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_SparetoSten
V2.3
pp_StenToSpare
V2.3
pp_Undo
pp_UpdateUndo
pp_UpdateUndoBox
Animation
pp_CountFrames
pp_GetCurrentFrame
V2.5

pp_GetJiffies

pp_GotoFrame
pp_MakeAnim
pp_NextFrame
pp_PreviousFrame
pp_SetJiffies

Effects

pp_Airbrush
pp_BumpMapping
V2.3
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_GaussianBlur
V2.3

pp_Gradient
V2

pp_Grey

pp_InnerBevel
V2.3

pp_Inv

pp_LineArt
V2.3

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 GetCurrentBuffer

pp_GetCurrentBuffer

Returns the number of the current buffer

1.4 GetCurrentFrame

pp_GetCurrentFrame

Returns the number of the current frame (if you have an animation).

1.5 GetCurrentBrush

pp_GetCurrentBrush

Returns the number of the current brush.

1.6 FindEmptyBuffer

pp_FindEmptyBuffer

Find an empty buffer and returns its number.

1.7 FindEmptyBrush

pp_FindEmptyBrush

Find an empty brush and returns its number.

1.8 SetBuffer

pp_SetBuffer a

a: (0-9)

Activates the Buffer nb a.

1.9 ClearCurrentBuffer

pp_ClearCurrentBuffer

Erase current buffer

1.10 Inverse

pp_Inv

Activates the colour inversion effect

1.11 New

pp_New width height bits

Creates a new picture of size width x height x bits

1.12 Rot90

pp_Rot90

Rotates the current picture by 90\textdegree

1.13 pp_Rotate

pp_Rotate x

Activates the rotation effect

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

1.14 FlipBrushX

pp_FlipBrushX

Flips the brush about the X-axis

1.15 FlipBrushY

pp_FlipBrushY

Flips the brush about the Y-axis

1.16 FlipX

pp_FlipX

Flips the current picture about the X-axis

1.17 pp_AliasOff

pp_AliasOff

Turns off antialiasing

1.18 pp_AliasOn

pp_AliasOn

Turns on antialiasing

1.19 pp_SparetoSten

pp_SparetoSten

Copy Green value of the spare picture over Stencil

1.20 pp_StentoSpare

pp_StentoSpare

Copy Grey value of the stencil over Spare picture

1.21 pp_GetAlias

pp_GetAlias

Returns 1 if antialiasing is active, 0 otherwise

1.22 pp_GetPenSize

pp_GetPenSize

Returns the size of the current pen

1.23 GetSizeText

pp_GetSizeText

Returns the current font size.

1.24 pp_GetPenType

pp_GetPenType

Returns the current pen type

1.25 FlipY

pp_FlipY

Flips the current picture about the Y-axis

1.26 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
```

1.27 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
```

1.28 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
```

1.29 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

1.30 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

1.31 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

1.32 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

/* ..... */

```

1.33 Gamma Correction

```

pp_Gamma Gamma

Gamma: -99,99

```

1.34 Color Balance

```

pp_ColorBalance R G B S Br C

R: Red -255,255
G: Green -255,255
B: Blue -255,255
S: Saturation -255,255
Br: Brightness -127,127
C: Contrast -255,255

```

1.35 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

1.36 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

1.37 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

1.38 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

1.39 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`

1.40 False Colours

`pp_False`

Activates the false colours effect

1.41 Transparency

`pp_Trans x`

Activates the transparency effect
x: 0-100%

1.42 Displace

`pp_Displace x`

Activates the Displace effect
x: 0-20

1.43 Greyscale

pp_Grey

Activates the grey effect

1.44 Threshold

pp_Threshold x

Activates the threshold effect

x: 0-255

1.45 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%

1.46 Airbrush

pp_Airbrush Opacity Flow

Activates the airbrush effect

Opacity: 0-100%
Flow: 0-30

1.47 NextFrame

pp_NextFrame

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

1.48 PreviousFrame

pp_PreviousFrame

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

1.49 GotoFrame

pp_GotoFrame n

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

1.50 CountFrames

pp_CountFrames

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

1.51 MakeAnim

pp_MakeAnim n

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

1.52 GetApen

pp_GetApen

Returns the number of the current foreground colour

1.53 GetBpen

pp_GetBpen

Returns the number of the current background colour

1.54 Plot

pp_Plot x y

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

1.55 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.

1.56 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
```

1.57 AddPoly

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

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

Example

1.58 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 */
```

1.59 Asay

```
pp_Asay message$
```

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

1.60 AskFile

```
pp_AskFile name$
```

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

1.61 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

1.62 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`

1.63 Circle

```
pp_Circle x y r
```

Draws an empty circle

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

1.64 CircleF

```
pp_CircleF x y r
```

Draws a filled circle

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

1.65 CloseAsay

```
pp_CloseAsay
```

Closes an asynchronous message window

1.66 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.

1.67 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.

1.68 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.

1.69 Cls

```
pp_Cls
```

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

1.70 EffectOn

```
pp_EffectOn
```

Activates 'Effect' mode

1.71 EffectOff

```
pp_EffectOff
```

Disactivates 'Effect' mode

1.72 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

1.73 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

1.74 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

1.75 pp_ClearRange

```
pp_ClearRange r
```

r: Number of the range (1-3)

Clear Range number r

Gradient example

1.76 pp_FishEyesLens

```
pp_FishEyesLens x
```

Activate the Fish eyes lens effect

x: -60,60

1.77 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.

1.78 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

1.79 pp_RestoreBrush

```
pp_RestoreBrush
```

1.80 pp_Ripple

```
pp_Ripple x
```

Activate the Ripple effect

x: 1-50

1.81 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

1.82 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 */
```

1.83 pp_Sin

```
pp_Sin Ay Fx Ax Fy

Activate the Sin effect

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

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

1.84 pp_Straw

```
pp_Straw a b

Activate the Straw effect

a:(1-50) Space between Straw
```

b: (2-50) Size of the Straw

1.85 pp_Twirl

pp_Twirl a

Activate the Twirl effect

a: (-720,720) angle

1.86 pp_Wave

pp_Wave a n

Activate the Wave effect

a: (-300,300) Amplitude

b: (1-300) number of waves

1.87 EndPoly

pp_EndPoly

Creates an empty polygon

Example

1.88 EndPolyS

pp_EndPolyS

Creates an empty polygon with a bezier algo.

Example

1.89 EndPolySF

pp_EndPolySF

Creates a filled polygon with a bezier algo.

Example

1.90 EndPoint

pp_EndPoint

just draw the points.

Example

1.91 EndPolyF

pp_EndPolyF

Creates a filled polygon

Example

1.92 Fill

pp_Fill x y

Flood fill starting from (x,y)

1.93 GetDepth

pp_GetDepth

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

1.94 GetHeight

pp_GetHeight

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

1.95 GetWidth

pp_GetWidth

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

1.96 GetDepthB

pp_GetDepthB

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

1.97 GetHeightB

pp_GetHeightB

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

1.98 GetWidthB

pp_GetWidthB

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

1.99 Light

pp_Light x

Activates the light effect
x: 0-100%

1.100 Line

pp_Line x y x1 y1

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

1.101 Load

pp_Load name\$

Loads a picture

name\$: name of the picture, with path

1.102 Bload

pp_Bload name\$

Loads a brush

name\$: name of the brush, with path

1.103 MaskOn

pp_MaskOn

Activates 'Mask' mode

1.104 MaskOff

pp_MaskOff

Disactivates 'Mask' mode

1.105 PenSize

pp_PenSize n

Returns the current pen size

1.106 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

1.107 Refresh

pp_Refresh

Refreshes the window containing the current image

1.108 Request

pp_Request message\$

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

1.109 Save

pp_Save name\$ a

Saves the current picture

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

1.110 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)

1.111 Psave

pp_Psave name\$ a

Saves the current palette

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

1.112 SetApen

pp_SetApen a

Sets the foreground colour to a

1.113 SetBpen

pp_SetBpen a

Sets the background colour to a

1.114 Shade

pp_Shade x

Activates the shade effect

x: 0-100%

1.115 ShowScreen

pp_ShowScreen

Brings the PerfectPaint screen to the front

1.116 SpareOnOff

pp_SpareOnOff

Activates or disactivates the spare page

1.117 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)

1.118 SetBrush

pp_SetBrush n

Activates brush number n

n: (0-9)

1.119 Pload

```
pp_Pload name$
```

Loads the palette name\$

1.120 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)

1.121 StartPoly

```
pp_StartPoly
```

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

Example

1.122 UpdateUndo

```
pp_UpdateUndo
```

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

1.123 Undo

```
pp_Undo
```

Same as pressing 'undo'

1.124 Warn

```
pp_Warn message$
```

Opens an alert window with an 'OK' button

1.125 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

1.126 PicttoSpare

```
pp_PicttoSpare
```

Copy current picture over the spare page

1.127 SparetoPict

```
pp_SparetoPict
```

Copy page spare over the current picture.

1.128 Spare

```
pp_Spare
```

Activates the Spare effect.

1.129 TextAntialiasing

```
pp_TextAntialiasing x
```

x: 0 -> No antialiasing

1 -> Antialiasing low

2 -> Antialiasing High

Activates Antialiasing.

Text example

1.130 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

1.131 TextSize

```
pp_TextSize x
```

```
x: font size
```

Text example

1.132 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

1.133 TextMake

```
pp_TextMake a$
```

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

Make a text brush with a bitmap or outline font.

Text example

1.134 TextDraw

```
pp_TextDraw x y
```

draw text brush on the current picture.

Text example

1.135 TextEffect

```
pp_TextEffect Effect x
```

Effect: 0 No effect
1 Outline
2 Bevel

x: Amplitude

Text example

1.136 TextFontName

```
pp_TextFontName fonte$
```

fonte\$: font name

Text example

1.137 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 */

.....

```

1.138 TextRotate

```
pp_TextRotate x
```

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

Text example

1.139 TextItalic

```
pp_TextItalic x
```

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

Text example

1.140 TextXDPI

```
pp_TextXDPI x
```

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

Text example

1.141 TextYDPI

```
pp_TextYDPI x
```

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

Text example

1.142 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
```

1.143 BrushOpacity

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


1.144 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.

1.145 ComposeReqOff

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

1.146 ComposeReqOn

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

1.147 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 concolution
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
```

1.148 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
```

1.149 FreeBrush

```
pp_FreeBrush
```

1.150 GetWidthT

```
pp_GetWidthT
Return the current width of the text
```

1.151 GetHeightT

```
pp_GetHeightT
Return the current height of the text
```

1.152 PickBrush

```
pp_PickBrush x y width height
```

1.153 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

1.154 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

1.155 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

1.156 ScaleBrush

```
pp_ScaleBrush Width Height
```

This command will scale the current brush.

1.157 UpdateUndoBox

```
pp_UpdateUndoBox x y x1 y1
```

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

1.158 LineArt

```
pp_LineArt Density
```

Activates the LineArt effect

Density: 1-3

1.159 GaussianBlur

```
pp_GaussianBlur Power
```

Activates the GaussianBlur effect

Power: 1-100

1.160 BumpMapping

```
pp_BumpMapping Lsize Lx Ly Depth Glossiness Fg
```

Activates the BumpMapping effect

Lsize: Size of the Light 1-255

Lx: Position x of the Light

Ly: Position y of the Light

Depth: 1-32

Glossiness: 1-255

Fg: 0 or 1

(set it to 1 if you want to use Foreground color
for Diffuse color)

Also you have to set Ambient color, Diffuse color and Specular color.

```
pp_Diffuse R G B
```

```
pp_Specular R G B
```

```
pp_Ambient R G B
```

1.161 InnerBevel

```
pp_InnerBevel Lsize Lx Ly Depth Smooth Glossiness Fg
```

Activates the InnerBevel effect

Lsize: Size of the Light 1-255

Lx: Position x of the Light

Ly: Position y of the Light

Depth: 1-32

Smooth: 1-32

Glossiness: 1-255

Fg: 0 or 1

(set it to 1 if you want to use Foreground color
for Diffuse color)

Also you have to set Ambient color, Diffuse color and Specular color.

```
pp_Diffuse R G B
```

```
pp_Specular R G B
```

```
pp_Ambient R G B
```

1.162 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
