

# **Arexx\_PerfectPaint**

Georges HALVADJIAN"

Copyright © CopyrightÂ©1998-1999 Georges HALVADJIAN"

COLLABORATORS

	TITLE : Arexx_PerfectPaint		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Georges HALVADJIAN"	January 19, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Arexx_PerfectPaint</b>	<b>1</b>
1.1	Arexx Commands . . . . .	1
1.2	Arexx Commands by Category . . . . .	4
1.3	Inverse . . . . .	6
1.4	New . . . . .	7
1.5	Rot90 . . . . .	7
1.6	pp_Rotate . . . . .	7
1.7	FlipBrushX . . . . .	7
1.8	FlipBrushY . . . . .	7
1.9	FlipX . . . . .	7
1.10	pp_AliasOff . . . . .	7
1.11	pp_AliasOn . . . . .	8
1.12	pp_GetAlias . . . . .	8
1.13	pp_GetPenSize . . . . .	8
1.14	GetSizeText . . . . .	8
1.15	pp_GetPenType . . . . .	8
1.16	FlipY . . . . .	8
1.17	Cycle . . . . .	8
1.18	Integer . . . . .	9
1.19	CheckBox . . . . .	10
1.20	Button . . . . .	10
1.21	Slider . . . . .	11
1.22	String . . . . .	11
1.23	Dialog Example . . . . .	12
1.24	Getfile . . . . .	13
1.25	Text . . . . .	13
1.26	DialogInit . . . . .	14
1.27	Dialog . . . . .	14
1.28	GetDialog . . . . .	15
1.29	False Colours . . . . .	15

---

1.30 Transparency . . . . .	15
1.31 Displace . . . . .	16
1.32 Greyscale . . . . .	16
1.33 Threshold . . . . .	16
1.34 Spray . . . . .	16
1.35 Airbrush . . . . .	16
1.36 NextFrame . . . . .	17
1.37 PreviousFrame . . . . .	17
1.38 GotoFrame . . . . .	17
1.39 CountFrames . . . . .	17
1.40 MakeAnim . . . . .	17
1.41 GetApen . . . . .	17
1.42 GetBpen . . . . .	18
1.43 Plot . . . . .	18
1.44 MultiRequest . . . . .	18
1.45 LongRequest . . . . .	18
1.46 AddPoly . . . . .	18
1.47 Example . . . . .	19
1.48 Asay . . . . .	19
1.49 AskFile . . . . .	19
1.50 Box . . . . .	19
1.51 BoxF . . . . .	19
1.52 Circle . . . . .	20
1.53 CircleF . . . . .	20
1.54 CloseAsay . . . . .	20
1.55 ClosestColor . . . . .	20
1.56 ClosestColorL . . . . .	20
1.57 ClosestColorD . . . . .	21
1.58 Cls . . . . .	21
1.59 EffectOn . . . . .	21
1.60 EffectOff . . . . .	21
1.61 Ellipse . . . . .	21
1.62 EllipseF . . . . .	22
1.63 EndPoly . . . . .	22
1.64 EndPolyS . . . . .	22
1.65 EndPolySF . . . . .	22
1.66 EndPoint . . . . .	22
1.67 EndPolyF . . . . .	23
1.68 Fill . . . . .	23

1.69	GetDepth	23
1.70	GetHeight	23
1.71	GetWidth	23
1.72	GetDepthB	23
1.73	GetHeightB	24
1.74	GetWidthB	24
1.75	Light	24
1.76	Line	24
1.77	Load	24
1.78	Bload	24
1.79	MaskOn	25
1.80	MaskOff	25
1.81	PenSize	25
1.82	PenType	25
1.83	Refresh	25
1.84	Request	25
1.85	Save	26
1.86	Bsave	26
1.87	Psave	26
1.88	SetApen	26
1.89	SetBpen	26
1.90	Shade	26
1.91	ShowScreen	27
1.92	SareOnOff	27
1.93	Spline	27
1.94	SetBrush	27
1.95	Pload	27
1.96	SplineF	27
1.97	StartPoly	27
1.98	UpdateUndo	28
1.99	Undo	28
1.100	Warn	28
1.101	Warp	28
1.102	PicttoSpare	28
1.103	SparetoPict	28
1.104	Spare	29
1.105	TextAntialiasing	29
1.106	TextColorMake	29
1.107	TextSize	29

---

1.108TextStyle . . . . .	29
1.109TextMake . . . . .	30
1.110TextDraw . . . . .	30
1.111TextEffect . . . . .	30
1.112TextFontName . . . . .	30
1.113Text example . . . . .	31
1.114TextRotate . . . . .	31
1.115TextItalic . . . . .	31
1.116TextXDPI . . . . .	32
1.117TextYDPI . . . . .	32
1.118TextForceType . . . . .	32

## 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`

B

`pp_Box`  
`pp_BoxF`  
`pp_Bload`  
`pp_Bsave`  
`pp_Button`           V1.1

C

`pp_CheckBox`  
`pp_Circle`  
`pp_CircleF`  
`pp_CloseAsay`  
`pp_ClosestColor`  
`pp_ClosestColorD`   V1.1  
`pp_ClosestColorL`   V1.1  
`pp_Cls`  
`pp_CountFrames`

---



pp\_Cycle

## D

pp\_Dialog  
pp\_DialogInit  
pp\_Displace V1.1

## E

pp\_EffectOff  
pp\_EffectOn  
pp\_Ellipse  
pp\_EllipseF  
pp\_EndPoint V1.1  
pp\_EndPoly  
pp\_EndPolyF  
pp\_EndPolyS V1.1  
pp\_EndPolySF V1.1

## F

pp\_False  
pp\_Fill  
pp\_FlipX  
pp\_FlipY  
pp\_FlipBrushX  
pp\_FlipBrushY

## G

pp\_GetAlias  
pp\_GetApen  
pp\_GetBpen  
pp\_GetDepth  
pp\_GetDepthB  
pp\_GetDialog  
pp\_GetFile  
pp\_GetHeight  
pp\_GetHeightB  
pp\_GetJiffies  
pp\_GetPenSize  
pp\_GetPenType  
pp\_GetSizeText V1.1  
pp\_GetWidth  
pp\_GetWidthB  
pp\_GotoFrame  
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 V1.1  
pp\_Pload  
pp\_Plot  
pp\_PreviousFrame  
pp\_Psave

## R

pp\_Refresh  
pp\_Request  
pp\_Rot90  
pp\_Rotate

## S

pp\_Save  
pp\_SetApen  
pp\_SetBpen  
pp\_SetBrush  
pp\_SetJiffies  
pp\_Shade  
pp\_ShowScreen  
pp\_Slider  
pp\_Spare V1.1  
pp\_SpareOnOff  
pp\_SparetoPict V1.1  
pp\_Spline  
pp\_SplineF  
pp\_Spray  
pp\_StartPoly  
pp\_String

## T

pp\_Text  
pp\_TextAntialiasing V1.1  
pp\_TextColorMake V1.1  
pp\_TextDraw V1.1  
pp\_TextEffect V1.1  
pp\_TextFontName V1.1  
pp\_TextForceType V1.1  
pp\_TextItalic V1.1  
pp\_TextMake V1.1  
pp\_TextRotate V1.1  
pp\_TextSize V1.1  
pp\_TextStyle V1.1  
pp\_TextXDPI V1.1  
pp\_TextYDPI V1.1  
pp\_Threshold

---

```

pp_Trans

U
pp_Undo
pp_UpdateUndo

W
pp_Warn
pp_Warp

```

## 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      V1.1
pp_EndPoly
pp_EndPolyF
pp_EndPolyS      V1.1
pp_EndPolySF     V1.1
pp_Fill
pp_GetApen
pp_GetBpen
pp_GetDepth
pp_GetDepthB
pp_GetHeight
pp_GetHeightB
pp_GetPenSize
pp_GetPenType
pp_GetSizeText   V1.1
pp_GetWidth
pp_GetWidthB
pp_Line
pp_PenSize

```

---

pp\_PenType  
pp\_Plot  
pp\_SetApen  
pp\_SetBpen  
pp\_SetBrush  
pp\_Spline  
pp\_SplineF  
pp\_StartPoly

#### Text

pp_TextAntialiasing	V1.1
pp_TextColorMake	V1.1
pp_TextDraw	V1.1
pp_TextEffect	V1.1
pp_TextFontName	V1.1
pp_TextForceType	V1.1
pp_TextItalic	V1.1
pp_TextMake	V1.1
pp_TextRotate	V1.1
pp_GetSizeText	V1.1
pp_TextSize	V1.1
pp_TextStyle	V1.1
pp_TextXDPI	V1.1
pp_TextYDPI	V1.1

#### GUI

pp_Asay	
pp_AskFile	
pp_Button	V1.1
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\_Save

#### Misc

pp\_AliasOff  
pp\_AliasOn

---

```
pp_ClosestColor
pp_ClosestColorD  V1.1
pp_ClosestColorL  V1.1
pp_FlipX
pp_FlipY
pp_FlipBrushX
pp_FlipBrushY
pp_GetAlias
pp_MaskOff
pp_MaskOn
pp_PicttoSpare    V1.1
pp_Refresh
pp_Rot90
pp_ShowScreen
pp_SpareOnOff
pp_SparetoPict    V1.1
pp_Undo
pp_UpdateUndo
```

#### Animation

```
pp_CountFrames
pp_GetJiffies
pp_GotoFrame
pp_MakeAnim
pp_NextFrame
pp_PreviousFrame
pp_SetJiffies
```

#### Effects

```
pp_Airbrush
pp_Displace      V1.1
pp_EffectOff
pp_EffectOn
pp_False
pp_Grey
pp_Inv
pp_Light
pp_Rotate
pp_Shade
pp_Spare         V1.1
pp_Spray
pp_Threshold
pp_Trans
pp_Warp
```

## 1.3 Inverse

```
pp_Inv
```

Activates the colour inversion effect

## 1.4 New

```
pp_New width height
```

Creates a new picture of size width\*height

## 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=colour range 1  
2=colour range 2  
3=colour range 3  
4=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 EndPoly

```
pp_EndPoly
```

Creates an empty polygon

Example

## 1.64 EndPolyS

```
pp_EndPolyS
```

Creates an empty polygon with a bezier algo.

Example

## 1.65 EndPolySF

```
pp_EndPolySF
```

Creates a filled polygon with a bezier algo.

Example

## 1.66 EndPoint

```
pp_EndPoint
```

just draw the points.

Example

---

## 1.67 EndPolyF

pp\_EndPolyF

Creates a filled polygon

Example

## 1.68 Fill

pp\_Fill x y

Flood fill starting from (x,y)

## 1.69 GetDepth

pp\_GetDepth

Returns the depth of the current picture

0: No picture in memory

## 1.70 GetHeight

pp\_GetHeight

Returns the height of the current picture

0: No picture in memory

## 1.71 GetWidth

pp\_GetWidth

Returns the width of the current picture

0: No picture in memory

## 1.72 GetDepthB

pp\_GetDepthB

Returns the depth of the current brush

0: No picture in memory

---

### 1.73 GetHeightB

```
pp_GetHeightB
```

Returns the height of the current brush

0: No picture in memory

### 1.74 GetWidthB

```
pp_GetWidthB
```

Returns the width of the current brush

0: No picture in memory

### 1.75 Light

```
pp_Light x
```

Activates the light effect

x: 0-100%

### 1.76 Line

```
pp_Line x y x1 y1
```

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

### 1.77 Load

```
pp_Load name$
```

Loads a picture

name\$: name of the picture, with path

### 1.78 Bload

```
pp_Bload name$
```

Loads a brush

name\$: name of the brush, with path

---

## 1.79 MaskOn

pp\_MaskOn

Activates 'Mask' mode

## 1.80 MaskOff

pp\_MaskOff

Disactivates 'Mask' mode

## 1.81 PenSize

pp\_PenSize n

Returns the current pen size

## 1.82 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

## 1.83 Refresh

pp\_Refresh

Refreshes the window containing the current image

## 1.84 Request

pp\_Request      message\$

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

---



## 1.85 Save

```
pp_Save name$ a
```

Saves the current picture

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

## 1.86 Bsave

```
pp_Bsave name$ a
```

Saves the current brush

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

## 1.87 Psave

```
pp_Psave name$ a
```

Saves the current palette

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

## 1.88 SetApen

```
pp_SetApen a
```

Sets the foreground colour to a

## 1.89 SetBpen

```
pp_SetBpen a
```

Sets the background colour to a

## 1.90 Shade

```
pp_Shade x
```

Activates the shade effect

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

---

## 1.91 ShowScreen

```
pp_ShowScreen
```

Brings the PerfectPaint screen to the front

## 1.92 SareOnOff

```
pp_SpareOnOff
```

Activates or disactivates the spare page

## 1.93 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)

## 1.94 SetBrush

```
pp_SetBrush n
```

Activates brush number n  
n: (1-2)

## 1.95 Pload

```
pp_Pload name$
```

Loads the palette name\$

## 1.96 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)

## 1.97 StartPoly

```
pp_StartPoly
```

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

Example

---

## 1.98 UpdateUndo

pp\_UpdateUndo

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

## 1.99 Undo

pp\_Undo

Same as pressing 'undo'

## 1.100 Warn

pp\_Warn message\$

Opens an alert window with an 'OK' button

## 1.101 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

## 1.102 PicttoSpare

pp\_PicttoSpare

Copy current picture over the spare page

## 1.103 SparetoPict

pp\_SparetoPict

Copy page spare over the current picture.

---

## 1.104 Spare

```
pp_Spare
```

Activates the Spare effect.

## 1.105 TextAntialiasing

```
pp_TextAntialiasing x
```

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

Activates Antialiasing.

Text example

## 1.106 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

## 1.107 TextSize

```
pp_TextSize x
```

```
x: font size
```

Text example

## 1.108 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

## 1.109 TextMake

```
pp_TextMake a$
```

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

Make a text brush with a bitmap or outline font.

Text example

## 1.110 TextDraw

```
pp_TextDraw x y
```

draw text brush on the current picture.

Text example

## 1.111 TextEffect

```
pp_TextEffect Effect x
```

Effect: 0 No effect  
1 Outline  
2 Bevel

x: Amplitude

Text example

## 1.112 TextFontName

```
pp_TextFontName fonte$
```

fonte\$: font name

Text example

---

### 1.113 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 */

.....

```

### 1.114 TextRotate

```

pp_TextRotate x

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

Text example

```

### 1.115 TextItalic

```

pp_TextItalic x

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

Text example

```

## 1.116 TextXDPI

```
pp_TextXDPI x
```

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

```
Text example
```

## 1.117 TextYDPI

```
pp_TextYDPI x
```

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

```
Text example
```

## 1.118 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
```

---