

Introduction

Print Previewer is a powerful Activex® Control for including Previewing capabilities to your users before getting a printed page out of your application.

Now you do not have to work around using previewing features of office applications.

Just drop in this control on your form and issue Commands to the PP Control, instead of the printer object.

That's all. Give a powerful and fully functional feature to your users.

You can also use it as a previewer while designing a printed output from your applications, saving lot of time and paper for taking trial printouts.

Related Topics

[Purpose of PP](#)

[How to use PP](#)

[Properties](#)

[Methods](#)

[Ordering Information](#)

Purpose of PP Control

Visual Basic is a great tool for creating applications and there is no second opinion needed to establish this fact.

However, there are quite a few capabilities that you would want to incorporate in your applications so that your users can enjoy the features that you imagine to provide.

Printing out documents or graphics or charts through your applications is just one of this features. But, it is also amongst the most widely required and demanded feature in almost all commercial applications.

Visual Basic has great printing manipulation capabilities and you must have used them widely till now. The problem arises, when you or your users want to see what's going to be printed and how would it appear on paper. Visual Basic, sadly lets you down on this aspect.

Now, with PP Control, you can add this functionality to all your applications without much hassle or need to rewrite entire routines.

Just drop the control on a Form, and redirect the printing commands to the control instead of the printer. Your users and of course you, can now preview the page before it is printed. The preview includes zooming also.

Related Topics

[Introduction](#)

[How to use PP](#)

[Properties](#)

[Methods](#)

Ordering Information

How to use PP

Using PP is simple.

Just follow these steps

- Add the PP Printing and Previewing Control to your Project. (Click Project / Components, select PP and click OK)
- Now in the toolbox, a new PP control will be added. Just drop this control on the Form which calls your printing subroutine.
- Replace your printing commands like follows:
 instead of writting **Printer.Print "This is Some Text"**
 write the same as **PP1.PrintText "This is Some Text"**

 instead of writting **Printer.Line (120,120)-(1000,1000),,B**
 write the same as **PP1.DrawBox (120,120,1000,1000)**
- You can use the Edit / Replace command for doing these replacement actions.
- Go through all the [Properties](#) and Methods of PP Control before you do so.

That's all it takes to add a useful feature to your application.

Related Topics

[Purpose of PP](#)

[Introduction](#)

[Properties](#)

[Methods](#)

Ordering Information

Properties

CurrentX

DrawMode

FillColor

FontBold

FontUnderline

ForeColor

PageSize

ScaleLeft

ScaleMode

CurrentY

DrawStyle

FillStyle

FontItalic

FontStrikethru

LeftMargin

PageWidth

ScaleTop

TopMargin

Destination

DrawWidth

Font

FontSize

FontTransparent

PageHeight

ScaleHeight

ScaleWidth

CurrentX,CurrentY Property

The CurrentX and CurrentY properties set or return the current position of the Printing Cursor.

Syntax:

```
Object.PP1.CurrentX = 100  
Object.PP1.CurrentY = 100
```

where

- Object is the name of the object on which PP control is dropped.
- PP1 is the default name of the PP control. You must replace with the name you give to this control.
- The distance of Offset will depend on the current [ScaleMode](#) property of PP Control.

Data Type : Single

Availability : Runtime and Designtime.

Destination Property

The Destination Property sets or returns the name of the destination for output.

Syntax:

```
object.PP1.Destination = 0
```

where

- object is the name of the object on which the PP Control is dropped.
- PP1 is the default name of the PP control. You must replace with the name you give to this control.
- The value of Destination can be
 ppWindow (Default) with value = 0 for previewing
 or ppPrinter with value = 1 for printing without previewing

Data Type : Integer

Availability : Runtime and Designtime.

DrawMode Property

This property is similar to the DrawMode property of the Printer Object.

See Visual Basic's [Help](#) for further information on DrawMode.

DrawStyle Property

This property is similar to the DrawStyle property of the Printer Object.

See Visual Basic's [Help](#) for further information on DrawStyle.

DrawWidth Property

This property is similar to the DrawWidth property of the Printer Object.

See Visual Basic's [Help](#) for further information on DrawWidth.

FillColor Property

This property is similar to the FillColor property of the Printer Object.

See Visual Basic's [Help](#) for further information on FillColor.

FillStyle Property

This property is similar to the FillStyle property of the Printer Object.

See Visual Basic's [Help](#) for further information on FillStyle.

Font Property

This property is similar to the Font property of the Printer Object.

See Visual Basic's [Help](#) for further information on Font.

FontBold Property

This property is similar to the FontBold property of the Printer Object.

See Visual Basic's [Help](#) for further information on FontBold.

FontItalic Property

This property is similar to the FontItalic property of the Printer Object.

See Visual Basic's [Help](#) for further information on FontItalic.

FontSize Property

This property is similar to the FontSize property of the Printer Object.

See Visual Basic's [Help](#) for further information on FontSize.

FontUnderline Property

This property is similar to the FontUnderline property of the Printer Object.

See Visual Basic's [Help](#) for further information on FontUnderline.

FontStrikethru Property

This property is similar to the FontStrikethru property of the Printer Object.

See Visual Basic's [Help](#) for further information on FontStrikethru.

FontTransparent Property

This property is similar to the FontTransparent property of the Printer Object.

See Visual Basic's [Help](#) for further information on FontTransparent.

ForeColor Property

This property is similar to the ForeColor property of the Printer Object.

See Visual Basic's [Help](#) for further information on ForeColor.

ScaleHeight Property

This property is similar to the ScaleHeight property of the Printer Object.

See Visual Basic's [Help](#) for further information on ScaleHeight.

ScaleLeft Property

This property is similar to the ScaleLeft property of the Printer Object.

See Visual Basic's [Help](#) for further information on ScaleLeft.

ScaleTop Property

This property is similar to the ScaleTop property of the Printer Object.

See Visual Basic's [Help](#) for further information on ScaleTop.

ScaleWidth Property

This property is similar to the ScaleWidth property of the Printer Object.

See Visual Basic's [Help](#) for further information on ScaleWidth.

LeftMargin Property

The LeftMargin property sets or returns the Horizontal Offset of the printing start column. This distance is determined by the [ScaleMode](#) property set currently.

Syntax :

`object.PP1.LeftMargin = number`

where

- Object is the name of the object that hosts the PP Control.
- PP1 is the default name given to the PP control. You must replace it with the name you have given to the control.
- *number* should be greater than the [ScaleLeft](#) and lesser than the [ScaleWidth](#) properties for proper printing management.

DataType : Single

Availability : Runtime and Designtime.

ScaleMode Property

This property is similar to the ScaleMode property of the Printer Object.

See Visual Basic's [Help](#) for further information on ScaleMode.

TopMargin Property

The TopMargin property sets or returns the Vertical Offset of the printing start row. This distance is determined by the [ScaleMode](#) property set currently.

Syntax :

`object.PP1.TopMargin = number`

where

- Object is the name of the object that hosts the PP Control.
- PP1 is the default name given to the PP control. You must replace it with the name you have given to the control.
- *number* should be greater than the [ScaleTop](#) and lesser than the [ScaleHeight](#) properties for proper printing management.

DataType : Single

Availability : Runtime and Designtime.

PageHeight, PageWidth, PageSize Properties

These properties set or return the dimensions of the Printer Page.

Syntax :

```
object.PP1.PageHeight = number  
object.PP1.PageWidth = number  
object.PP1.PageSize = number
```

where

- Object is the name of the object that hosts the PP Control.
- PP1 is the default name given to the PP control. You must replace it with the name you have given to the control.
- number is the size of printout you desire.
- In case you want to automatically set the pagesize, you can select from the given constant pagesizes. This will set the PageWidth and PageHeight automatically.
- In case you select custom pagesize, you must then set the pagewidth and pageheight properties yourself.

DataType : Single

Availability : Runtime and Designtime.

Methods

[ChangeSize](#)

[DrawBox](#)

[DrawLine](#)

[DrawCircle](#)

[PrintText](#)

[SetScale](#)

[ShowPreview](#)

[TextHeight](#)

[TextWidth](#)

ChangeSize Method

The ChangeSize method resets the pagesize according to the dimensions specified.

Syntax :

Call object.PP1.ChangeSize(*number*)

where

- object is the name of the object hosting the PP Control.
- PP1 is the default name of the PP Control. You must replace it with the name you have given to the PP Control.
- *number* is as specified in the constants list that appears. If the number is set to ppCustom then you must specify the PageWidth and PageHeight in immediate commands before the results appear.

Ensure that this is the first method applied before any printing commands are issued. The results can be unexpected if this is not done.

DrawBox Method

The DrawBox method draws a Rectangle.

Syntax :

Call Object.PP1.DrawBox (x1,y1,x2,y2,[True/False])

where

- object is the name of the object hosting the PP Control.
- PP1 is the default name of the PP Control. You must replace it with the name you have given to the PP Control.
- x1,y1 are single values indicating the horizontal and vertical offsets respectively of the top left corner of the rectangle.
- x2,y2 are single values indicating the horizontal and vertical offsets respectively of the bottom right corner of the rectangle.
- [True/False] is **Optional**. (**False** by default) If set to true will draw a filled rectangle using the [FillColor](#) and the [FillStyle](#) properties. Else an empty rectangle will be drawn.
- The thickness and style of the Borders will be determined by the [DrawWidth](#) and the [DrawStyle](#) properties respectively.

DrawLine Method

The DrawLine method draws a Line.

Syntax :

Call Object.PP1.DrawLine (x1,y1,x2,y2)

where

- object is the name of the object hosting the PP Control.
- PP1 is the default name of the PP Control. You must replace it with the name you have given to the PP Control.
- x1,y1 are single values indicating the horizontal and vertical offsets respectively of the starting point of the line.
- x2,y2 are single values indicating the horizontal and vertical offsets respectively of the ending point of the line.
- The thickness of the Line and its style will be determined by the [DrawWidth](#) and the [DrawStyle](#) properties respectively.

DrawCircle Method

The DrawCircle method draws a Circle.

Syntax :

Call Object.PP1.DrawCircle (x1,y1,r,[True/False])

where

- object is the name of the object hosting the PP Control.
- PP1 is the default name of the PP Control. You must replace it with the name you have given to the PP Control.
- x1,y1 are single values indicating the horizontal and vertical offsets respectively of the center of the circle.
- r is the single value determining the radius of the circle.
- [True/False] is **Optional**. (**False** by default) If set to true will draw a filled circle using the [FillColor](#) and the [FillStyle](#) properties. Else an empty circle will be drawn.
- The thickness and style of the Borders will be determined by the [DrawWidth](#) and the [DrawStyle](#) properties respectively.

PrintText Method

The PrintText method prints a line of Text.

Syntax :

Call Object.PP1.PrintText ("This is Sample Text")

where

- object is the name of the object hosting the PP Control.
- PP1 is the default name of the PP Control. You must replace it with the name you have given to the PP Control.
- You must specify the complete string as you would have passed to a printer object's Print method.
- To print an empty line , pass a "" value.
- The text is printed starting at the CurrentX , CurrentY location using the Font property in the color set by the ForeColor .

SetScale Method

Scales the Printer object to given values.

Syntax :

Call Object.PP1.SetScale ([x1,y1,x2,y2])

where

- object is the name of the object hosting the PP Control.
- PP1 is the default name of the PP Control. You must replace it with the name you have given to the PP Control.
- x1,x2,y1 and y2 become the new values of ScaleLeft, ScaleTop,ScaleWidth and ScaleHeight.
- You must specify either all the values or no values at all.
- Specifying no values is not supported as yet.

ShowPreview Method

This is the method which invokes either Previewing or Printing depending on the [Destination](#) Property

Syntax :

Call object.PP1.ShowPreview()

where

- object is the name of the object hosting the PP Control.
- PP1 is the default name of the PP Control. You must replace it with the name you have given to the PP Control.
- Ensure to issue this as the last command in your printing routine.

TextHeight Method

Returns the height of the Text about to be printed.

Syntax :

Call Object.PP1.TextHeight("Sample Text")

where

- object is the name of the object hosting the PP Control.
- PP1 is the default name of the PP Control. You must replace it with the name you have given to the PP Control.
- The TextHeight is represented in the ScaleMode units.

TextWidth Method

Returns the width of the Text about to be printed.

Syntax :

Call Object.PP1.TextWidth("Sample Text")

where

- object is the name of the object hosting the PP Control.
- PP1 is the default name of the PP Control. You must replace it with the name you have given to the PP Control.
- The TextWidth is represented in the ScaleMode units.

Ordering Information

For orders, enquiries, suggestions, problems and queries contact

Deshpande's Designs

7, Yogeshwar Appartments

Vastrapur,

Ahmedabad - 380 015.

Gujarat

India

tel : 91 - 79 - 673 0448

e-Mail. n_deshpande@hotmail.com

Dirrerence between Trial and Licensed Verions

The Trial version supports only single page output and only two levels of zooming i.e. 100% and 50 %.

The Licensed version supports multiple page ourput and includes Thumbnail previews.

