



TPrintPreview Component

[Properties](#)

[Methods](#)

[Events](#)

[Tasks](#)

[Overview](#)

[About](#)

Unit

PrntPrev

Description

The PrintPreview component is intended to give an alternative method of handling printing from within Delphi. Originally designed to supply a print preview facility, sadly lacking in the VCL, to Delphi the component grew to attempt to supply all printing facilities.

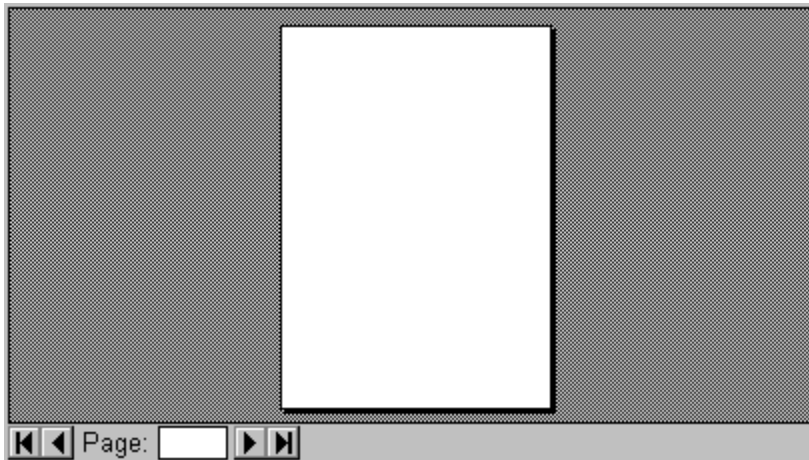
The need for this component grew from a graphical application, and it is for those which it is most suitable. There is no facility for assigning streams to a TPrintPreview component, just a canvas for you to draw upon.

Using the TPrintPreview component

To create a minimal application for using TPrintPreview do the following :-

1. Place the component on the form. There are two alternatives for doing this, you can either dedicate a form to the preview, or place the component over all the other components on the form. The former method allows you to fake the standard MDI print preview look (as used by Microsoft Word for Windows™). The latter method has an effect more like that used in Microsoft Access, which I prefer. In both cases you will probably want to set the Align property to alClient. Do not try to make the component visible, you cant. The visibility of the component is controlled by the Previewing property.
2. Set the following properties:
 - MinPage: The lowest page number that can be selected in the Print dialogue box.
 - MaxPage: The largest page number that can be selected in the Print dialogue box.
 - Options: Set appropriately for your required aspects of the Print dialogue box.
 - UseAbortDlgMode: Set to either dmGauge or dmText.
 - ZoomCursor: Set to the value of DefCursorNum.
 - ZoomFactor: The default TPrintPreview allows only one zoom factor, a factor of three is reasonable.
3. Set handlers for the following events:
 - OnRenderPage: The event that contains the page rendering details.
 - OnSelectionPages: If the user elects to print selected pages in the Print dialogue box this event is fired. In this event you should set the FromPage and ToPage properties.
4. Call the following methods in your application:
 - PrintJobDlg: The lowest page number that can be selected in the Print dialogue box.
 - SetupDlg: The largest page number that can be selected in the Print dialogue box.
 - StartPrintJob: Set appropriately for your required aspects of the Print dialogue box.
5. Write code to change the value of the Previewing property.

Overview



The TPrintPreview component is intended to be used for all printing functions. The visible aspect of the component is a ScrollBox containing a preview of the print job, in addition to this the component has methods for showing the Windows Print and Print Setup common dialogue boxes and the ability to send a print job to either the preview or the printer. The component is intended to make the addition of graphical printing, complete with preview, to any application a simple a painless job.

Because TPrintPreview is so different to the VCLs TPrinter it does not function in a manner anything like the original. TPrintPreview has more in common with the MFCs print preview class than TPrinter. The main difference is that all rendering is done during an event, OnRenderPage, when the printer is ready, rather than when the application thinks it is.

During the OnRenderPage event the application may perform any operation on the canvas that it wishes. There are exceptions to this. As the preview is controlled by mapping modes the only Windows API functions that will work as expected on the preview are those which work with logical co-ordinates. There are two circumstances in which workarounds have been supplied for known problems with this :-

1. Changing the mapping mode. As the print preview component is implemented using mapping modes you cannot change the mapping mode using Windows API calls during the rendering of a page. To avoid this problem the ChangeMapMode method has been supplied. This method sets the mapping mode in a manner which enables the preview to remain true to the printed image.
2. Setting clipping regions. The Windows API functions for managing clipping areas work with device, rather than logical co-ordinates. To allow for clipping the two methods SetClipArea and UnSetClipArea have been supplied. Both methods work with logical co-ordinates (the only co-ordinates in TPrintPreview).

The component has a wide range of read only, run-time only, properties containing details of the printer. These include, the size of the paper, the size of the printable area of a page and the distance from the edge of the paper to the start of the printable area. All these dimensions are available in both pixels and HiMetric (1/100 mm) units. It is advised that you use HiMetric units wherever possible, though for drawing purposes the mapping mode can be changed as you see fit. It is hoped that a future version of TPrintPreview will allow these properties to be writeable.

To make the component as flexible as possible the control panel for the preview mode can be hidden and replaced by your own controls. Also there is an abort dialogue built into the component that can either display a gauge the increments as each page is rendered, or a label indicating which page is being printed. This abort dialogue may be turned off and replaced by one of your own design, for a more accurate gauge, if you require. The final aspect of customisation is the cursor. TPrintPreview has built in support for a magnifying glass cursor for zoom purposes. All aspects of this can be

customised, from the design of the cursor, through the cursor number it uses, to complete control of the OnMouseDown event.

Oh, the advantages of SpudSofts TPrintPreview:

1. Requires minimal input from the application to add printing and preview.
2. Self contained. If you dont want to do anything, you dont have to.
3. Not intended for report processing (only an advantage if you dont want to print reports!).
4. Flexible. I hope it can do anything with a printer that is possible.
5. Supported. For the foreseeable future I will be doing everything I can to make TPrintPreview what you want. This will be reduced when other projects take over.

SpudSoft

SpudSoft has been in existence for about five years now and is the name under which all the software I produce goes. SpudSoft is a partnership between myself and Mr Spud Murphy, the two foot tall furry potato that overseas all my work.

The TPrintPreview component was created because I got really annoyed with the printing facilities that come with Delphi. My current project is a fully featured MDI application that needed some form of print preview and writing TPrintPreview seemed to be the most satisfactory method of solving my problems. Look out for SpudStitch, coming soon to an ftp site near you.

TPrintPreview is not free. Its, err, postcardware. If you find it useful send me a postcard telling me something about your project. If you want the source the please send me at least £20, either sterling or the equivalent in Belgian Francs. I would also appreciate a copy of any finished apps you create using TPrintPreview. Once you have the source code you are free to do anything you wish with it, though I request that you dont make it publicly available and do feed any modifications through me.

As I only own one computer (well, only one PC and a ZX81, a QL an Amiga...) I cannot verify that TPrintPreview will work on all computers. If you do find a problem please let me know about it in as much detail as possible. If you want a bug fixed you will have to include some way for me to return the fix to you, either an email address or enough money for postage.

Oh, I guess if I want things from you Im going to have to tell you how to find me:
















email:	jt@spudsoft.demon.co.uk
Forwarding snail:	JT SpudSoft 64 Brooks Lane Whitwick Leicester LE67 5DE England

Im moving country in two months time, so both the above will be slow in contacting me. My email address will be checked whenever I feel like paying for an international modem call, the other is just a forwarding address, so dont expect too quick a response from these. If you give me an email address Ill let you know my new email address as soon as Ive sorted one out.

Properties

▶ Run-time only properties

 Key properties

▶	<u>Abort</u>	
	<u>MinPage</u>	
	<u>Previewing</u>	
	<u>Align</u> ▶	<u>NumBins</u>
▶	<u>PrintHiMet</u>	
	▶ <u>Colour</u>	
▶	<u>NumColours</u>	
▶	<u>PrintPixels</u>	
	<u>Copies</u> ▶	<u>NumSizes</u>
▶	<u>PrtFileName</u>	
	▶ <u>CurrentPage</u>	
	<u>Options</u>	
	<u>ToPage</u>	
	▶ <u>CurZoom</u>	
▶	<u>Orientation</u>	
	<u>UseAbortDlgMode</u>	
	<u>DefCursorNum</u> ▶	<u>PageHiMet</u>
	<u>UsePageCtrls</u>	
	▶ <u>DeviceName</u>	
▶	<u>PagePixels</u>	
	<u>UseZoom</u>	
	▶ <u>Duplex</u>	
▶	<u>PaperBinsArr</u>	
▶	<u>Xdpi</u>	
	 <u>FromPage</u>	
▶	<u>PaperSize</u>	
▶	<u>Ydpi</u>	
	▶ <u>IndentHiMet</u>	
▶	<u>PaperSizeArr</u>	<u>ZoomCursor</u>
	 <u>IndentPixels</u>	
	<u>PaperSizeName</u>	<u>ZoomFactor</u>
	 <u>MapMode</u>	
	<u>PaperSource</u>	
	<u>ZoomPos</u>	
	 <u>MaxPage</u>	
	<u>PaperSourceName</u>	

Abort Property

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

property Abort : Boolean;

Description

Run-time only.

The abort property is set to False before any print job destined for the printer. Setting Abort to true causes the current job to abort and, if possible, be removed from the print queue.

Setting Abort is only logical during the [OnRenderPage](#) or [OnChkAbort](#) events, or during the event handler of an Abort dialogue box supplied by the application.

The intended use of Abort is for the creation of application controlled abort dialogue boxes.

Example

The following code causes the current print job to abort if the user has asked to print any page after the fifth. I cant think why you would want to do this as a standalone user would not receive any output from the program

```
procedure TForm1.PrintPreview1ChkAbort(Page: Word);  
begin  
    Desc.Caption := 'Page: ' + IntToStr(Page);  
    if Page > 6 then  
        PriuntPeview1.Abort  
end;
```


See also

[OnRenderPage](#) event.

[OnChkAbort](#) event.

Align Property

Applies to
TPrintPreview

Declaration
property Align : TAlign;

Description
The Align property determines how the TPrintPreview control aligns within its container (or parent control).

For more details see the Delphi help.

Example

The following code

See also

See also for Align

Colour Property

See also

Applies to

TPrintPreview

Declaration

property Colour : Boolean;

Description

Run-time only and read only.

The colour property is set to True if the printer is capable of printing in colour.

Example

The following code

See also

NumColours property

Copies Property

[See also](#)

Applies to

TPrintPreview

Declaration

property Copies : Word;

Description

The Copies property is the number of copies that will be requested when the print job is sent to the printer. It can be set programatically (or during the design stage) but is always reset when the Print dialogue box is displayed.

Example

The following code

See also

[PrintJobDlg](#) method

CurrentPage Property

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

property CurrentPage : Word;

Description

Run-time only.

The CurrentPage property contains the number of the page that is currently being displayed by the preview.

If you are creating your own control buttons for the preview you will need to use the CurrentPage property to control the page being displayed.

The property serves is not intended for use during the [OnRenderPage](#) event (the page number for rendering is passed as an argument).

Example

The following code will change the page being displayed to page 7. If the preview is not currently showing (Previewing is False) then page 7 will be the first page displayed when the preview shows.

```
PrintPreview1.CurrentPage := 7;
```

See also

[OnRenderPage](#) event

[Previewing](#) property

[UsePageCtrls](#) property

CurZoom Property

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

property CurZoom : Double;

Description

Run-time only.

The CurZoom property contains the current zoom factor for the preview.

If the standard page controls are being used this property will either be 1.0, the value of the [ZoomFactor](#) property or a value set programatically.

Example

Example code shared with MouseToHiMet et al.

See also

[OnMouseDown](#) event

[UsePageCtrls](#) property

[ZoomFactor](#) property

[ZoomPos](#) property

DefCursorNum Property

[See also](#)

Applies to

TPrintPreview

Declaration

property DefCursorNum : Integer;

Description

The DefCursorNum property may only be set at design time. Attempts to set it at any other time will achieve nothing.

DefCursorNum is the cursor number that will be used for the magnify cursor. You cannot prevent the magnify cursor from being loaded, but the DefCursorNum property allows you to ensure that it does not interfere with any other custom cursors being used in your application.

Example

The following code

See also

The Delphi help system, custom cursors.

[ZoomCursor](#)

DeviceName Property

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

property DeviceName : String;

Description

Run-time only and read only.

The DeviceName property holds the name of the device in use for the current printer. This is used by Windows API functions but is also readable by users - for example, "PCL/HP LaserJet" in the case of the Hewlett-Packard LaserJet.

It is not currently possible to set this string programatically, only through the user actions with the Print and Setup dialogue boxes. A future version will hopefully allow the device to be selected.

Example

The following code puts up a dialogue box reporting the name of the current device.

```
MessageDlg('Printer device: ' + PrintPreview1.DeviceName, mtConfirmation, [mbOK], 0);
```

See also

[PrintJobDlg](#)

[SetupDlg](#)

Duplex Property

Applies to
TPrintPreview

Declaration
property Duplex : Boolean;

Description
Run-time only and read only.

The Duplex property is True if the currently selected printer supports duplex printing.

Example

The following code

See also

See also for Duplex

FromPage Property

See also

Applies to

TPrintPreview

Declaration

property FromPage : Word;

Description

The FromPage property contains the number of the page that the print job is to start from. If the property is set before the PrintJobDlg method is used the Print dialogue box will have its Erom text box filled with this value.

After the PrintJobDlg method has been used the FromPage property will be set to the value entered into the Erom text box.

FromPage can never be set to a value lower than MinPage, any attempt to do so will set it to the value of MinPage.

Example

The following code

See also

[MaxPage](#) property

[MinPage](#) property

[PrintJobDlg](#) method

[ToPage](#) property

IndentHiMet Property

See also

Applies to
TPrintPreview

Declaration
property IndentHiMet : Tpoint;

Description
Run-time only and read only.

The value of IndentHiMet.X is the distance from the left of a sheet of paper to the left hand edge of the printable area in units of 1/100 mm.

The value of IndentHiMet.Y is the distance from the top of a sheet of paper to the top of the printable area in units of 1/100 mm.

Example

The following code

See also

[IndentPixels](#) property

[PageHiMet](#) property

[PrinHiMet](#) property

IndentPixels Property

[See also](#)

Applies to
[TPrintPreview](#)

Declaration
property IndentPixels : Tpoint;

Description
Run-time only and read only.

The value of IndentPixels.X is the distance from the left of a sheet of paper to the left hand edge of the printable area in pixels.

The value of IndentPixels.Y is the distance from the top of a sheet of paper to the top of the printable area in pixels.

Example

The following code

See also

[IndentHiMet](#) property

[PagePixels](#) property

[PrintPixels](#) property

MapMode Property

[See also](#)

Applies to

[TPrintPreview](#)

Declaration

property MapMode : [TMapMode](#);

Description

Run-time only and read only.

The MapMode property contains the current mapping mode in operation.

Example

The following code

See also

[ChangeMapMode](#) method

[TMapMode](#) type

TMapMode Type

Declaration

TMapMode = (mmText, mmLoMetric, mmHiMetric, mmLoEnglish, mmHiEnglish, mmTWIPs, mmlsotropic, mmAnisotropic);

Description

The TMapMode type is used to define the mapping modes used by the [TPrintPreview](#) component.

TMapMode is the type of the [MapMode](#) property and is also the type of the Mode argument to the [ChangeMapMode](#) method.

TMapMode can take the following values:

Value	Meaning
mmText	Units of one pixel.
mmLoMetric	Units of 1/10 mm.
mmHiMetric	Units of 1/100 mm.
mmLoEnglish	Units of 1/10 inch.
mmHiEnglish	Units of 1/100 inch.
mmTWIPs	Units of 1/1440 inch.
mmlsotropic	Arbitrary units, horizontal units equal to vertical units.
mmAnisotropic	Arbitrary units.

MaxPage Property

See also

Applies to
TPrintPreview

Declaration
property MaxPage : Word;

Description

The MaxPage property contains the maximum number that ToPage can be set to. This limit applies both at all times and also limits the maximum value that can be entered into the To box of the Print dialogue box.

Setting MaxPage to a value lower than the current value of ToPage will cause ToPage to be set to the value of MaxPage.

Example

The following code

See also

[FromPage](#) property

[MinPage](#) property

[PrintJobDlg](#) method

[ToPage](#) property

MinPage Property

See also

Applies to

TPrintPreview

Declaration

property MinPage : Word;

Description

The MinPage property contains the minimum number that FromPage can be set to. This limit applies both at all times and also limits the maximum value that can be entered into the From box of the Print dialogue box.

Setting MinPage to a value higher than the current value of FromPage will cause FromPage to be set to the value of MinPage.

Example

The following code

See also

[FromPage](#) property

[MaxPage](#) property

[PrintJobDlg](#) method

[ToPage](#) property

NumBins Property

Applies to
TPrintPreview

Declaration
property NumBins : Longint;

Description
Run-time only and read only.

The NumBins property contains the number of bins (paper sources) present in the current printer.

Example

The following code

See also

NumColours Property

Applies to
TPrintPreview

Declaration
property NumColours : Integer;

Description
Run-time only and read only.

The NumColours property contains the number of colours reported as being present in the printer by the GetDeviceCaps Windows API function.

If the printer is monochrome (Colour = False) NumColors will be set to two.

As I haven't yet tried to use a colour printer with TPrintPreview I cannot vouch for the accuracy of this figure. I do know that with an HP Deskjet 500C printer driver on Windows 95 (release 490) NumColours is set to 8.

Example

The following code

See also

See also for NumColours

NumSizes Property

[See also](#)

Applies to

[TPrintPreview](#)

Declaration

property NumSizes : Longint;

Description

Run-time only and read only.

The NumSizes property contains the number of different paper sizes supported by the current printer.

Information regarding alternative paper sizes is stored in the [PaperSize](#), [PaperSizeArr](#) and [PaperSizeName](#) properties.

Example

The following code

See also

[PaperSize](#) property

[PaperSizeArr](#) property

[PaperSizeName](#) property

Options Property

See also

Applies to

TPrintPreview

Declaration

property Options : TPrintDlgOptions;

Description

The Options property is used to control aspects of the Print dialogue box and to return information about the dialogue box after it's use.

The possible values that can be included in the Options set for the TPrintPreview component are as follows:

Value	Meaning
pdAllPages	Causes the <u>A</u> ll radio button to be selected when the Print dialogue box opens. When the dialogue box has been closed indicates that the <u>A</u> ll radio button was selected.
pdCollateCopies	The Collate Copies check box is checked in the Print dialogue box.
pdDisablePrintToFile	Disables the Print to <u>F</u> ile check box in the Print dialogue box.
pdHidePrintToFile	Hides and disables the Print to <u>F</u> ile check box in the Print dialogue box.
pdNoPageNums	Disables the <u>P</u> ages radio button and the <u>F</u> rom and <u>T</u> o edit controls.
pdNoSelection	Disables the <u>S</u> election radio button.
pdPageNums	Causes the <u>P</u> ages radio button to be selected when the Print dialogue box opens. When the dialogue box has been closed indicates that the <u>P</u> ages radio button was selected.
pdPrintToFile	Causes the Print to <u>F</u> ile check box to be checked when the Print dialogue box opens. When the dialogue box has been closed indicates that the Print to <u>F</u> ile check box was checked.
pdSelection	Causes the <u>S</u> election radio button to be selected when the Print dialogue box opens. When the dialogue box has been closed indicates that the <u>S</u> election radio button was selected.
pdShowHelp	Causes the <u>H</u> elp button to be displayed in the Print dialogue box.

Example

The following code

See also

[PrintJobDlg](#) method

[TPrintDlgOptions](#) type

TPrintDlgOptions Type

Declaration

```
TPrintDlgOption = (pdAllPages, pdCollateCopies, pdDisablePrintToFile, pdHidePrintToFile,  
                  pdNoPageNums, pdNoSelection, pdPageNums, pdPrintToFile, pdSelection,  
                  pdShowHelp);  
TPrintDlgOptions = set of TPrintDlgOption;
```

Description

The TPrintDlgOptions type defines the set of values the Options property of the TPrintPreview component can have.

Orientation Property

See also

Applies to
TPrintPreview

Declaration
property Orientation : TPrinterOrientation;

Description
Run-time only.

The orientation property controls the current orientation of the page. It's value may be set either programatically or by user interaction with the Setup dialogue box.

Example

The following code

See also

TPrinterOrientation type

SetupDlg method

TPrinterOrientation Type

Declaration

TPrinterOrientation = (poPortrait, poLandscape)

Description

The TPrinterOrientations type defines the values that the Orientation property can have.

PageHiMet Property

See also

Applies to

TPrintPreview

Declaration

property PageHiMet : Tpoint;

Description

Run-time only and read only.

The value of PageHiMet.X is the width of the currently selected sheet of paper in units of 1/100 mm.

The value of PageHiMet.Y is the height of the currently selected sheet of paper in units of 1/100 mm.

Example

The following code

See also

[IndentHiMet](#)

[PagePixels](#)

[PrinHiMet](#)

PagePixels Property

[See also](#)

Applies to

TPrintPreview

Declaration

property PagePixels : Tpoint;

Description

Run-time only and read only.

The value of PagePixels.X is the width of the currently selected sheet of paper in pixels.

The value of PagePixels.Y is the height of the currently selected sheet of paper in pixels.

Example

The following code

See also

[IndentPixels](#)

[PageHiMet](#)

[PrintPixels](#)

PaperBinsArr Property

[See also](#)

Applies to

[TPrintPreview](#)

Declaration

property PaperBinsArr[Index : ShortInt] : [TPaperBins](#);

Description

Run-time only and read only.

The PaperBinsArr property is used to obtain information regarding the alternative paper sources available to the current printer.

Example

The following code

See also

PaperSource property

PaperSourceName property

TPaperBins Type

Declaration

```
TPaperBins = record  
    Bin: Word;  
    Name: array[0..PaperBinsNames] of Char;  
end;
```

Description

The TPaperBins type is used to contain details of paper sources available to a printer.

Bin is the value used by Windows API functions to refer to the paper source.

Name is a null terminated string containing the name given to the paper source by the printer driver.

PaperSize Property

[See also](#)

Applies to

[TPrintPreview](#)

Declaration

property PaperSize : Word;

Description

Run-time only.

The PaperSize property contains the current paper size in use by the printer according to Windows API functions.

This value may be set to any value of Size from the [PaperSizeArr](#) property. Setting it to any other value will have an undefined effect.

Example

The following code

See also

[IndentHiMet](#) property

[IndentPixels](#) property

[PageHiMet](#) property

[PagePixels](#) property

[PaperSizeArr](#) property

[PaperSizeName](#) property

[PrintHiMet](#) property

[PrintPixels](#) property

PaperSizeArr Property

[See also](#)

Applies to

[TPrintPreview](#)

Declaration

property PaperSizeArr[Index : ShortInt] : TPaperSize;

Description

Run-time only and read only.

The PaperSizeArr property is used to obtain information regarding the alternative paper sizes available to the current printer.

Example

The following code

See also

[IndentHiMet](#) property

[IndentPixels](#) property

[PageHiMet](#) property

[PagePixels](#) property

[PaperSize](#) property

[PaperSizeName](#) property

[PrintHiMet](#) property

[PrintPixels](#) property

TPaperSize Type

Declaration

```
TPaperSize = record  
    Size : Word;  
    Dimensions : TPOINT;  
    Name : array[0..PaperSizeNames] of Char;  
end;
```

Description

The TPaperSize type is used to contain details of paper sources available to a printer.

Size is the value used by Windows API functions to refer to the paper size.

Dimensions is the size of each sheet of paper in HiMetric (1/100mm) units.

Name is a null terminated string containing the name given to the paper size by the printer driver.

PaperSizeName Property

[See also](#)

Applies to

TPrintPreview

Declaration

property PaperSizeName : String;

Description

Run-time only and read only.

The PaperSizeName property contains the name of the currently selected paper size in human readable terms.

Example

The following code

See also

[IndentHiMet](#) property

[IndentPixels](#) property

[PageHiMet](#) property

[PagePixels](#) property

[PaperSize](#) property

[PaperSizeArr](#) property

[PrintHiMet](#) property

[PrintPixels](#) property

PaperSource Property

[See also](#)

Applies to
[TPrintPreview](#)

Declaration
property PaperSource : Word;

Description
Run-time only.

The PaperSource property contains the current paper source in use by the printer according to Windows API functions.

This value may be set to any value of Bin from the [PaperBinsArr](#) property. Setting it to any other value will have an undefined effect.

Example

The following code

See also

PaperBinsArr property

PaperSourceName property

PaperSourceName Property

[See also](#)

Applies to

[TPrintPreview](#)

Declaration

property PaperSourceName : String;

Description

Run-time only and read only.

The PaperSourceName property contains the name of the currently selected paper source in human readable terms.

Example

The following code

See also

PaperBinsArr property

PaperSource property

Previewing Property

Applies to
TPrintPreview

Declaration
property Previewing : Boolean;

Description

The Previewing property is used to initiate and terminate the previewing functionality. It may be used instead of the Visible property, which does not feature in TPrintPreview, to control whether the component is visible or not. If Previewing is False there are no visible features of TPrintPreview.

The methods and properties of TPrintPreview are useable regardless of the state of Previewing, printing is possible without using the preview.

Example

The following code

See also

See also for Previewing

PrintHiMet Property

[See also](#)

Applies to

TPrintPreview

Declaration

property PrintHiMet : Tpoint;

Description

Run-time only and read only.

The value of PrintHiMet.X is the width of the printable area on the currently selected sheet of paper in units of 1/100 mm.

The value of PrintHiMet.Y is the height of the printable area on the currently selected sheet of paper in units of 1/100 mm.

Example

The following code

See also

[IndentHiMet](#)

[PageHiMet](#)

[PrintPixels](#)

PrintPixels Property

[See also](#)

Applies to

TPrintPreview

Declaration

property PrintPixels: TPoint read FPrintPixels;RTO

Description

Run-time only and read only.

The value of PrintPixels.X is the width of the printable area on the currently selected sheet of paper in pixels.

The value of PrintPixels.Y is the height of the printable area on the currently selected sheet of paper in pixels.

Example

The following code

See also

[IndentPixels](#)

[PagePixels](#)

[PrintHiMet](#)

PrtFileName Property

See also

Applies to

TPrintPreview

Declaration

property PrtFileName : String;

Description

Run-time only.

If the Print to file check box is checked in the Print dialogue box TPrintPreview will try to print a file of the name given by the PrtFileName property.

If there is no string assigned to PrtFileName a Save As dialogue box will open and the user will be given the opportunity to name the file to be created.

Example

The following code

See also

[Options](#) property

[PrintJobDlg](#) method

ToPage Property

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

property ToPage: Word read FToPage write SetToPage default 1;Pub

Description

The ToPage property contains the number of the page that the print job is to stop at. If the property is set before the [PrintJobDlg](#) method is used the Print dialogue box will have its [To](#) text box filled with this value.

After the [PrintJobDlg](#) method has been used the ToPage property will be set to the value entered into the [To](#) text box.

ToPage can never be set to a value higher than [MaxPage](#), any attempt to do so will set it to the value of [MaxPage](#).

Example

The following code

See also

[FromPage](#) property

[MaxPage](#) property

[MinPage](#) property

[PrintJobDlg](#) method

UseAbortDlgMode Property

See also

Applies to

TPrintPreview

Declaration

property UseAbortDlgMode : TAbortDlgMode;

Description

The UseAbortDlgMode property controls the display of the built in abort dialogue box.

The abort dialogue box can be displayed either with a gauge that increments as each page is printed, or with a text display of the number of the page currently being printed.

Neither is optimal and two future improvements will be to allow for the creation of a custom dialogue box and to allow the range and progression of the dialogue box gauge to be controlled from the application.

UseAbortDlgMode can take the following values:

Value	Meaning
dmGauge	An abort dialogue box containing a gauge that increments with each printed page will be displayed when a job is sent to the printer.
dmNone	No abort dialogue box is displayed.
dmText	An abort dialogue box containing a line of text stating the page being printed will be displayed when a job is sent to the printer.

Example

The following code

See also

[TAbortDlgMode](#) type
[StartPrintJob](#) method.

TAbortDlgMode Type

Declaration

TAbortDlgMode = (dmNone, dmText, dmGauge);

Description

The TAbortDlgMode type defines the values that the UseAbortDlgMode property can have.

UsePageCtrls Property

See also

Applies to

TPrintPreview

Declaration

property UsePageCtrls : Boolean

Description

The UsePageCtrls property controls the display of controls at the bottom of the TPrintPreview area. If UsePageCtrls is False the visible property of the panel containing the controls will remain set to False.

If the page controls are not used the application will have to supply it's own method for the user to control the page being previewed.

Example

The following code

See also

CurrentPage property

UseZoom Property

See also

Applies to
TPrintPreview

Declaration
property UseZoom : Boolean;

Description

The UseZoom property controls the zooming of the preview picture. If set to True the default handler of the MouseDown event will zoom (by ZoomFactor) over the area under the cursor, or return to a full page view if the view is already zoomed.

Use of the OnMouseDown event, MouseToHiMet method and CurZoom and ZoomPos properties can be used to control the zooming in a custom manner.

Regardless of the state of the UseZoom property the cursor will always be changed to the cursor specified by the ZoomCursor property when over the printable area of the preview.

Example

The following code

See also

[CurZoom](#) property

[MouseToHiMet](#) method

[OnMouseDown](#) event

[ZoomCursor](#) property

[ZoomPos](#) property

Xdpi Property

See also

Applies to

TPrintPreview

Declaration

property Xdpi : Integer;

Description

Run-time only and read only.

The number of dots per horizontal inch on the printed page of the currently selected printer.

You are advised not to use this as it should be unnecessary. Please inform me of any tasks which can only be accomplished by the use of this property.

Example

The following code

See also

Ydpi property

Ydpi Property

See also

Applies to
TPrintPreview

Declaration
property Ydpi : Integer;

Description
Run-time only and read only.

The number of dots per vertical inch on the printed page of the currently selected printer.

You are advised not to use this as it should be unnecessary. Please inform me of any tasks which can only be accomplished by the use of this property.

Example

The following code

See also

[Xdpi](#) property

ZoomCursor Property

See also

Applies to
TPrintPreview

Declaration
property ZoomCursor : Tcursor;

Description
When the cursor is over the printable area of the preview page the cursor will be changed to ZoomCursor.

In order to use the built in magnifier cursor this property should be set to the value of the DefCursorNum property.

Example

The following code

See also

DefCursorNum property

ZoomFactor Property

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

property ZoomFactor : Double;

Description

The default MouseDown event handler zooms the preview by ZoomFactor centred on the point under the cursor if the [UseZoom](#) property is True.

Example

Example shared with MouseToHiMet et al.

See also

UseZoom property

ZoomPos Property

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

property ZoomPos : Tpoint;

Description

Run-time only.

If the [CurZoom](#) property is non-zero setting the ZoomPos property to a point on the page (in HiMetric units) will cause the zoom to be centred on that point.

ZoomPos does not track the location of the cursor or serve any purpose after the initial positioning of the zoom focus.

Example


Example shared with MouseToHiMet et al.


See also


CurZoom property

Methods

 Key Methods

 ChangeMapMode

 PrintJobDlg

 UnSetClipArea

 Create

 SetClipArea

 Destroy

 SetupDlg

 MouseToHiMet

 StartPrintJob

ChangeMapMode Method

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

function ChangeMapMode(Canvas: TCanvas; Mode: [TMapMode](#); X, Y: Integer): [TMapMode](#);

Description

ChangeMapMode is the only method of changing the mapping mode that will work with [TPrintPreview](#). Attempts to use Windows API functions will cause the preview to display incorrectly. If the canvas is destined for the printer this function simply passes the call to a correctly formatted call to the Windows API function. If the canvas is for a preview the function maps the required mapping mode to either mmIsotropic or mmAnisotropic to provide the correct look.

ChangeMapMode should only be used during an [OnRenderPage](#) event.

The function returns the current mapping mode. In most cases this will be the new mapping mode, in the case of an error occurring no change to the mapping mode will take place and the old mapping mode will be returned. It is recommended that the return value is checked.

Example

The following code is shared with OnRenderPage

See also

[MapMode](#) property

[OnRenderPage](#) event

[TMapMode](#) property

Create Method

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

constructor Create(AOwner: TComponent); override;

Description

The Create component for [TPrintPreview](#) is the same as that for any TPanel component (the class from which [TPrintPreview](#) is descended)

Example

The following code has yet to be written.

See also

The Delphi Help file

Destroy Method

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

destructor Destroy; override;

Description

The Create component for [TPrintPreview](#) is the same as that for any TPanel component (the class from which [TPrintPreview](#) is descended)

Example

The following code has yet to be written.

See also

The Delphi Help file.

MouseToHiMet Method

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

function MouseToHiMet(MousePos: TPoint): TPoint;

Description

MouseToHiMet is used to convert the co-ordinates of a mouse event into HiMetric units that can be used to set ZoomPos.

Example

The following code implements a three stage zoom process.

It is assumed that UseZoom has been set to False.

```
procedure TForm1.PrintPreview1MouseDown(Sender: TObject; Button: TMouseButton; Shift:
    TShiftState; X, Y: Integer);
var
    MousePosScreen: TPoint;
    MousePosHiMet: TPoint;
begin
    MousePosScreen := Point(X, Y);
    MousePosHiMet := PrintPreview1.MouseToHiMet(MousePosScreen);
    with PrintPreview1 do
        begin
            if CurZoom = 3.0 then
                CurZoom := 6.0
            else if CurZoom = 1.0 then
                CurZoom := 3.0
            else
                CurZoom := 1.0;
        end;
    PrintPreview1.ZoomPos := MousePosHiMet;
end;
```

See also

[UseZoom](#) property

[ZoomPos](#) property

PrintJobDlg Method

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

procedure PrintJobDlg;

Description

The PrintJobDlg method displays the Print common dialogue box.

The Print dialogue box is initialised according to the values of the [Options](#), [FromPage](#), [ToPage](#), [MinPage](#), [MaxPage](#) and [Copies](#) properties.

When the Print dialogue box is closed it updates the values of the [Options](#), [FromPage](#), [ToPage](#) and [Copies](#) properties. The print job is *not* started automatically, a call to the [StartPrintJob](#) method is required.

If the user selected the Setup button within the Print dialogue button the effect would be exactly the same as calling the [SetupDlg](#) method. In this case all properties detailing the capabilities of the printer are liable to have changed. Caching values prior to a call to the PrintJobDlg is likely to result in incorrect behaviour.

Example

The following code is an event handler for a print button.

```
procedure TForm1.PrtDlgBtnClick(Sender: TObject);  
begin  
    PrintPreview1.PrintJobDlg;  
    PrintPreview1.StartPrintJob(Document 1);  
end;
```

See also

[Copies](#) property

[FromPage](#) property

[MaxPage](#) property

[MinPage](#) property

[Options](#) property

[StartPrintJob](#) method

[ToPage](#) property

SetClipArea Method

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

procedure SetClipArea(Canvas: TCanvas; Rect: TRect);

Description

The SetClipArea method is the only way to set clipping rectangles with [TPrintPreview](#). This routine is still in need of investigation as it does not function with text on HP LaserJets under Windows 95.

Anyway, try it out, its fairly obvious. Anything outside of Rect is clipped, Rect is measured in the current units. Please let me know the results of your experiments.

Example

The following code is shared with the OnRenderPage event.

See also

[UnSetClipArea](#) method

SetupDlg Method

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

procedure SetupDlg;

Description

Displays the Print Setup dialogue box.

Following the call to SetupDlg all properties detailing the capabilities of the printer are likely to have changed. Caching values prior to a call to the SetupDlg is likely to result in incorrect behaviour.

Example

The following code is an example handler for a Print Setup button. The code is possibly deceptively simple. Remember that no property values used before the call can be assumed to be the same afterwards.

```
procedure TForm1.SetupBtnClick(Sender: TObject);  
begin  
    PrintPreview1.SetupDlg;  
end;
```

See also

[Colour](#) property

[DeviceName](#) property

[Duplex](#) property

[IndentHiMet](#) property and [IndentPixels](#) property

[NumBins](#) property

[NumColours](#) property

[NumSizes](#) property

[Options](#) property

[Orientation](#) property

[PageHiMet](#) property and [PagePixels](#) property

[PaperBinsArr](#) property

[PaperSize](#) property

[PaperSizeArr](#) property

[PaperSizeName](#) property

[PaperSource](#) property

[PaperSourceName](#) property

[PrintHiMet](#) property and [PrintPixels](#) property

[Xdpi](#) property and [Ydpi](#) property

StartPrintJob Method

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

procedure StartPrintJob(JobName: String);

Description

The StartPrintJob method is used to place a job called JobName in the print queue and start the process that will result in an Abort dialogue box (assuming [UseAbortDlgMode](#) is not set to dmNone) being displayed and one [OnRenderPage](#) event for each number between [FromPage](#) and [ToPage](#), inclusive.

Your application cannot rely on all [OnRenderPage](#) events after a StartPrintJob call being for the printer. There is no built in method for determining whether a job is intended for the screen or not. If you would like the [OnRenderPage](#) event to have that information let me know.

Example

The following code is shared with PrintJobDlg.

See also

OnRenderPage event

UnSetClipArea Method

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

procedure UnSetClipArea(Canvas: TCanvas);

Description

The UnSetClipArea method removes the clipping rectangle created by a call to [SetClipArea](#), if any.

If no clipping rectangles exist UnSetClipArea achieves nothing.


Example

The following code is shared with OnRenderPage et al.


See also


[SetClipArea](#) method


Events

 Key Events

 OnChkAbort

 OnRenderPage

 OnSelectionPages

 OnMouseDown

 OnResize

OnChkAbort Event

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

property OnChkAbort : TAbortProcEvent;

Description

The OnChkAbort event occurs a large number of times during the processing of a print job. It occurs as a result of Windows interrupting the printing at frequent intervals to check that everything is alright. Assuming that nothing has gone wrong with your printing process the event can be ignored. If the print job need to be abort the [Abort](#) property should be set during this event.

This event will be of vital import during the processing of custom Abort dialogue boxes, but I dont think theyll work yet. The only way to create a custom Abort dialogue box would be just before the call to the [StartPrintJob](#) method. Unfortunately I think that the processing during the StartPrintJob will prevent any input from the app reaching the dialogue box. If you have the time to check whether this is true or not please do. Just create a custom dialogue box before the call to StartPrintJob and destroy it right after. The handler for the Abort button can either try to set the Abort property directly or set a global variable during the button event handler and set Abort during OnChkAbort if the global variable is True. Its all a question of which memory space things have access too and Im afraid I dont have time now.

Example

The following code is lacking at the moment.

See also

Abort property

UseAbortDlgMode property

TAabortProcEvent Type

Declaration

TAabortProcEvent = **procedure**(Page: Word) **of object**;

Description

The TAabortProcEvent is used for the OnChkAbort event of the TPrintPreview component.

The value of the Page argument is the page that is currently being processed.

OnMouseDown Event

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

property OnMouseDown : TMouseEvent;

Description

The OnMouseDown event occurs when the user presses a mouse button with the mouse pointer over the printable area of the preview. Use the OnMouseDown event handler when you want some processing to occur as a result of pressing a mouse button in addition to, or instead of, the standard zoom function.

The Button parameter of the OnMouseDown event identifies which mouse button was pressed. By using the Shift parameter of the OnMouseDown event handler, you can respond to the state of the mouse buttons and shift keys. Shift keys are the Shift, Ctrl, and Alt keys.

Example

The following code is shared with MouseToHiMet

See also

[MouseToHiMet](#) method

[UseZoom](#) property

OnRenderPage Event

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

property OnRenderPage : TRenderPageEvent;

Description

The OnRenderPage event occurs whenever you are required to draw on a canvas. This may occur as a result of new page being printed or as a result of the preview area being resized or zoomed. The event handler has no way of knowing why it has been called.

The page in need of rendering and a Canvas to draw upon are both passed as arguments to the event handler. The dimensions, and other details, of the Canvas can be obtained from other properties of the TPrintPreview component (specifically [PrintHiMet](#)).

When the OnRenderPage event occurs the Canvas is in the mmHiMetric mapping mode. If you wish to work in another mapping mode use the [ChangeMapMode](#) method.

Example

The following code is an OnRenderPage event handler that performs a number of simple operations on the canvas.

```
procedure TForm1.PrintPreview1RenderPage(Page: Word; Canvas: TCanvas);  
var  
    Current: Integer;  
    X, Y: Integer;  
begin  
    PrintPreview1.ChangeMapMode(Canvas, mmAnisotropic, 1000, 1000);  
    Canvas.Pen.Style := psSolid;  
    Canvas.Pen.Width := 2;  
    Canvas.Pen.Color := clBlack;  
    Canvas.Pen.Width := 1;  
    Canvas.Pen.Color := clBlack;  
    Canvas.Font.Name := 'Times New Roman';  
    Canvas.Font.Height := 500;  
    Canvas.Font.Color := clBlack;  
    X := 500 - (Canvas.TextWidth(IntToStr(Page)) div 2);  
    Y := 500 - (Canvas.TextHeight(IntToStr(Page)) div 2);  
    Canvas.TextOut(X, Y, IntToStr(Page));  
    PrintPreview1.SetClipArea(Canvas, Rect(0, 400, 1000, 600));  
    for Current := 0 to 500 do  
        begin  
            Canvas.Rectangle(500 + Current, 500 + Current, 500 + Current * 2, 500 + Current * 2);  
        end;  
    Canvas.MoveTo(0,0);  
    Canvas.LineTo(1000, 1000);  
    Canvas.MoveTo(1000, 0);  
    Canvas.LineTo(0, 1000);  
    PrintPreview1.UnSetClipArea(Canvas);  
end;
```

See also

[ChangeMapMode](#) method

TRenderPageEvent Type

Declaration

TRenderPageEvent = **procedure**(Page: Word; Canvas: TCanvas) **of object**;

Description

The TRenderPageEvent is used for the OnRenderPage event of the TPrintPreview component.

The value of Page is the number of the page in need of rendering and Canvas is a TCanvas to draw upon.

OnResize Event

Applies to
TPrintPreview

Declaration
property OnResize : TNotifyEvent;

Description
The OnResize event occurs whenever the component is resized while an application is running.

If the preview area is not zoomed it is rescaled and redrawn whenever an OnResize event occurs. This handler has been provided for you to add your own additional processing.

Example

The following code

See also

See also for OnResize

OnSelectionPages Event

[See also](#)

[Example](#)

Applies to

[TPrintPreview](#)

Declaration

property OnSelectionPages : TNotifyEvent;

Description

The OnSelectionPages event occurs whenever the user closes a Print dialogue box after having selected the Selection radio button. The event occurs to allow the application to set the [FromPage](#) and [ToPage](#) properties to values encompassing the current selection.

Example

The following code is a very simple OnSelectionPages event handler. In a more useful application the pages would be set to the currently selected pages.

```
procedure TForm1.PrintPreview1SelectionPages(Sender: TObject);  
begin  
    PrintPreview1.FromPage := 12;  
    PrintPreview1.ToPage := 36;  
end;
```

See also

FromPage property

ToPage property

