TPagePrinter

Hierarchy Properties Methods Events Example
Types Constants Non-Members

TPagePrinter gives you complete page layout and printing control and does print preview.

Unit

PagePrnt

Registration

TPagePrinter is shareware. The registered version is $50 (US) and comes with full source code. Please check my web page (http://www.public.usit.net/bmenees) or e-mail me (bmenees@usit.net) for the latest ordering information.

Description

The key methods of TPagePrinter are BeginDoc, WriteLine, WriteLines, EndDoc, and Print. The key properties of TPagePrinter are the Lines property and the MarginLeft, MarginRight, MarginTop, and MarginBottom properties.

By right-clicking the TPagePrinter component at design-time you can preview the current property settings (including the Lines property).

TPagePrinter will raise an EPagePrinter exception if there is a problem with one of its properties. If there are any problems with the underlying physical printer (e.g. if there are no printers installed), TPagePrinter will raise an EPrinter exception. If you are using PrintToFile (and have I/O checking enabled), an EInOutError exception can also be raised.

Notes

It **requires** long strings, enhanced metafiles, the Win32 common controls, and it makes use of several Win32 specific API calls. This means it can't be used with Delphi 1.0, so please don't ask, beg, threaten, etc. It has been tested with and seems to work fine with Delphi 2.0, Delphi 3.0, and C++Builder 1.0.

This component has its origins in TLinePrinter. I started off calling this component TLinePrinter Version 2.0, but I decided a new class name was more appropriate for several reasons. The main reason was that TLinePrinter is a non-visual component, and the new component is a visual component. I didn't want the new visual component to start showing up on forms where the V.1.0 component hadn't shown! A new class name also gave me the chance to redefine the interface entirely. I added, edited, renamed, and deleted many properties, methods, events, and units. I think you'll agree the changes are for the better.

Standard Disclaimer

TPagePrinter Example

TPagePrinter

A typical example of using TPagePrinter in Delphi is:

```
procedure TMainForm.btnPrintClick(Sender: TObject);
begin
     PagePrinter.BeginDoc;
     //Put something in the Lines property.
     PagePrinter.Lines.Assign(Memo.Lines);
     PagePrinter.WriteLines(False);
     PagePrinter.EndDoc;
     //Preview would be an app-specific property.
     if not Preview then PagePrinter.Print;
end;
```

In C++Builder this would be:

```
void __fastcall TMainForm::btnPrintClick(TObject *Sender)
{
    PagePrinter->BeginDoc();
    //Put something in the Lines property.
    PagePrinter->Lines->Assign(Memo->Lines);
    PagePrinter->WriteLines(false);
    PagePrinter->EndDoc();
    //Preview would be an app-specific property.
    if (!Preview) PagePrinter->Print();
}
```

Hierarchy

```
TObject
    |
TPersistent
    |
TComponent
    |
TControl
    |
TWinControl
    |
TScrollingWinControl
    |
TScrollBox
```

TPagePrinter Properties

TPagePrinter   Legend

All properites in TScrollBox plus:

- AbortOnCancel
- Align
- Alignment
- AutoFooterFont
- AutoHeaderFont
- AvailablePageHeight
- AvailablePageWidth
- Canvas
- CanvasPosition
- Collate
- Color
- Copies
- DefaultColWidth
- DragCursor
- DragMode
- Enabled
- FileName
- Footer
- FooterFont
- FooterFormat
- FriendlyFooter
- FriendlyHeader
- GradientBackground
- GutterBottom
- GutterLeft
- GutterRight
- GutterTop
- Header
- HeaderFont
- HeaderFormat
- LineNumber
- Lines
- LineSpacing
- MarginBottom
- MarginLeft
- MarginRight
- MarginTop
- MeasureUnit
- Orientation
- PageBorderOffset
- PageBorders
- PageCount
- PageNumber
- Pages
- ParentColor
- PhysicalPageHeight

PhysicalPageWidth

PrintableHeight

PrintableWidth
PrintFromPage

Printing
PrintToFile
PrintToPage
ProgressMessage
ShadowColor
ShadowOffset
ShowCancel
ShowHint
ShowMargins
ShowProgress
TableFormat
TableGrid
TabSize
Title
TokenSeparator
Visible
WordWrap
ZoomLocation
ZoomPercent

TPagePrinter Methods

TPagePrinter Legend

All methods in TScrollBox plus:

- BeginDoc
- BeginUpdate
- Clear
- Create
- Destroy
- EndDoc
- EndUpdate
- ▫ ExpandFriendlyFormat
- ▫ ExpandLogicalFields
- ▫ GetClippedLine
- ▫ GetPreviewPagePixels
- ▫ GetPrinterHandle
- ▫ GetScaleFactor
- ▫ Invalidate
- ▫ Loaded
- MeasureUnitsToPixelsH
- MeasureUnitsToPixelsV
- ▫ MeasureUnitsToScreenPixels
- NewLine
- NewPage
- ▫ PaintPreview
- ▫ ParseFormatToken
- PixelsToMeasureUnitsH
- PixelsToMeasureUnitsV
- PrevLine
- Print
- RefreshProperties
- ▫ ScaleValue
- ▫ SetPixelsPerInch
- ▫ SplitLine
- ▫ SplitLineAndPrint
- ▫ UpdateDesigner
- ▫ UpdatePagePreviewSize
- ▫ UpdateProgressDlg
- ▫ ValidateFormatString
- Write
- WriteLine
- WriteLineAligned
- WriteLines
- ▫ WriteTableGrid
- WriteTableLine
- ZoomToFit
- ZoomToHeight
- ZoomToWidth

TPagePrinter Events

TPagePrinter Legend

All events in TScrollBox plus:

- OnNewLine
- OnNewPage
- OnStartDrag

TPagePrinter Constants

TPagePrinter

In Pixels

DefaultDPI = 300
DefaultBorderWidth = 2

In Inches

DefaultPhysicalPageHeightIn = 11.0
DefaultPhysicalPageWidthIn = 8.5
DefaultAvailablePageHeightIn = 10.5
DefaultAvailablePageWidthIn = 8.0
DefaultGutterLeftIn = 0.25
DefaultGutterTopIn = 0.25

In Millimeters

DefaultPhysicalPageHeightMm = 297.0
DefaultPhysicalPageWidthMm = 210.0
DefaultAvailablePageHeightMm = 284.0
DefaultAvailablePageWidthMm = 198.0
DefaultGutterLeftMm = 6.0
DefaultGutterTopMm = 6.0

Expanded in Headers, Footers, and Tables.

LineField = '{$LINE}'
PageField = '{$PAGE}'
DateField = '{$DATE}'
TimeField = '{$TIME}'
TitleField = '{$TITLE}'

Progress Dialog Messages

ProgressFinishMsg = ‘<FINISH>’
SendingPagesMsg = 'Sending Pages To Printer'

TPagePrinter Types

TPagePrinter

EPagePrinter
TGradientOrientation
TLineSpacing
TMeasurement
TMeasureUnit
TPageBorder
TPageBorders
TPageList
TPixels
TPrintCanvas
TPrintPage
TZoomLocation

TPagePrinter Non-Member Functions

TPagePrinter

ExpandTabsAsSpaces
FillGradient
GenSpace
ReplaceSubString
StripBackToWhiteSpace
TokenizeString

Scope

- 🟩 Protected
- 🟩 Published

Accessibility

- 🟩 Read Only
- 🟩 Run-Time Only

TPagePrinter.AbortOnCancel

TPagePrinter

Determines what happens when the user cancels printing.

```
property AbortOnCancel: Boolean default False;
```

Description

If the user presses the Cancel button while printing and AbortOnCancel is True, then TPrinter.Abort is called.   If AbortOnCancel is False, then TPrinter.EndDoc is called.

In an ideal world you would probably want to set AbortOnCancel to True.   However, I've found that calling TPrinter.Abort multiple times can crash your program without even throwing an exception.   This doesn't happen on every system, every time, but I've seen it on several occasions when TPrinter.Abort was called more than three times within one session.   So to be as safe as possible, you should leave AbortOnCancel set to False.

See Also

ShowCancel, ShowProgress

TPagePrinter.Alignment

TPagePrinter

Alignment determines how text is aligned on the page.

```
property Alignment: TAlignment default taLeftJustify;
```

Description

Use Alignment to change the way text is printed with WriteLine or WriteLines.   Alignment can be one of the following values:

Value	Meaning
taLeftJustify	Align text to the left side of the page
taCenter	Center text horizontally on the page.
taRightJustify	Align text to the right side of the page.

TPagePrinter.AutoFooterFont

TPagePrinter

AutoFooterFont makes sure the FooterFont property gets changed any time the Font property gets changed.

```
property AutoFooterFont: Boolean default True;
```

Description

If AutoFooterFont is True, then FooterFont will be set equal to Font anytime the Font property changes. However, you *can* change FooterFont without the Font property being affected.   If AutoFooterFont is False, then FooterFont is completely independent of changes to the Font property.

See Also

AutoHeaderFont

TPagePrinter.AutoHeaderFont

TPagePrinter

AutoHeaderFont makes sure the HeaderFont property gets changed any time the Font property gets changed.

```
property AutoHeaderFont: Boolean default True;
```

Description

If AutoHeaderFont is True, then HeaderFont will be set equal to Font anytime the Font property changes.   However, you *can* change HeaderFont without the Font property being affected.   If AutoHeaderFont is False, then HeaderFont is completely independent of changes to the Font property.

See Also

AutoFooterFont

TPagePrinter.AvailablePageHeight

TPagePrinter

AvailablePageHeight gives the largest printable height of the page for the current printer.

```
property AvailablePageHeight: TMeasurement;
```

Description

Run-time and Read only.   AvailablePageHeight returns TPrinter.PageHeight converted from pixels into the current measure units.   This is essentially the physical height of the page minus the top and bottom gutters.

See Also

AvailablePageWidth

TPagePrinter.AvailablePageWidth

TPagePrinter

AvailablePageWidth gives the largest printable width of the page for the current printer.

```
property AvailablePageWidth: TMeasurement;
```

Description

Run-time and Read only. AvailablePageWidth returns TPrinter.PageWidth converted from pixels into the current measure units. This is essentially the physical width of the page minus the left and right gutters.

See Also

AvailablePageHeight

TPagePrinter.Canvas

TPagePrinter

Returns a reference to the current canvas so you can do custom drawing.

```
property Canvas: TPrintCanvas;
```

Description

Run-time and Read only.   Because the Canvas property is read-only you can’t assign another Canvas to it.   However, you can still use all properties and methods of the Canvas as you would expect to.

TPagePrinter.CanvasPosition

TPagePrinter

Returns the point on the Canvas where the next line will be printed.

```
property CanvasPosition: TPoint;
```

Description

Run-time and Read only.   The point returned is relative to the printable area as defined by the *margins*.   Thus (0,0) is the point just inside the left and top margins as defined by the MarginLeft and MarginTop properties.

You will probably never need to use this property.   It is only useful if you need to do custom drawing on the Canvas and don’t want to interfere with existing text.

TPagePrinter.Collate

TPagePrinter

Controls whether multiple copies are collated when physically printed.

```
property Collate: Boolean default True;
```

Description

If Collate is True and multiple copies are being printed, then pages are printed in the order: 1,2,3,…,1,2,3,…,1,2,3,.... If Collate is False and multiple copies are being printed, then pages are printed in the order 1,1,1,...,2,2,2,…,3,3,3,…. Collate has no effect when only one copy is being printed. Collate also has no effect on the preview window.

See Also

Copies

TPagePrinter.Copies

TPagePrinter

Controls how many copies of a document will be physically printed.

```
property Copies: Cardinal default 1;
```

Description

Copies has no effect on the preview window.

See Also

Collate

TPagePrinter.DefaultColWidth

TPagePrinter

Default column width to use when printing table lines.

```
property DefaultColWidth: TMeasurement;
```

Description

The column width that will be used if the HeaderFormat, FooterFormat, or TableFormat contain invalid or missing column widths.

TPagePrinter.FileName

TPagePrinter

The name of the file to print to when PrintToFile is True.

```
property FileName: String;
```

Description

When PrintToFile is True, all text will be output to the file specified in FileName.

All or part of FileName is printed if the FriendlyHeader or FriendlyFooter properties contain the Friendly Format Specifiers &f or &F.

TPagePrinter.Footer

TPagePrinter

The text to be printed at the bottom of each page.

```
property Footer: String;
```

Description

The Footer property is a *table string* that is laid out as specified by the FooterFormat property. For more information see: Table String.

Note: There are limitations on the Footer when PrintToFile is True.

See Also

FooterFont, FriendlyFooter, Header

TPagePrinter.FooterFont

TPagePrinter

The font used when printing the Footer.

```
property FooterFont: TFont;
```

Description

The FooterFont is only used when printing the Footer text.   If you want the FooterFont to change everytime the Font property changes, set AutoFooterFont to True.

See Also

HeaderFont

TPagePrinter.FooterFormat

TPagePrinter

The format specifier for the Footer property.

```
property FooterFormat: String;
```

Description

FooterFormat is a *table format string* that specifies the column widths and alignments to be used for each field in the Footer string. For more information see: Table String.

See Also

FooterFont, FriendlyFooter, HeaderFormat

TPagePrinter.FriendlyFooter

TPagePrinter

FriendlyFooter is an alternative way to specify both the Footer and FooterFormat strings.

```
property FriendlyFooter: String;
```

Description

FriendlyFooter can contain multiple Friendly Format Specifiers which control alignment and field content.   Setting FriendlyFooter will automatically rebuild the Footer and FooterFormat properties.   If FriendlyFooter is a non-empty string, then calling BeginDoc also causes the Footer and FooterFormat properties to be rebuilt.

See Also

FriendlyHeader

TPagePrinter.FriendlyHeader

TPagePrinter

FriendlyHeader is an alternative way to specify both the Header and HeaderFormat strings.

```
property FriendlyHeader: String;
```

Description

FriendlyHeader can contain multiple Friendly Format Specifiers which control alignment and field content.   Setting FriendlyHeader will automatically rebuild the Header and HeaderFormat properties. If FriendlyHeader is a non-empty string, then calling BeginDoc also causes the Header and HeaderFormat properties to be rebuilt.

See Also

FriendlyFooter

TPagePrinter.GradientBackground

TPagePrinter

Toggles the gradient background effect on and off.

```
property GradientBackground: Boolean default True;
```

Description

When GradientBackground is True, the background of the preview window makes a gradient from Black to the color specified in the Color property. When GradientBackground is False, the background of the preview window is solid-filled with Color.

TPagePrinter.GutterBottom

TPagePrinter

The height of the unprintable area at the bottom of a page in the printer.

```
property GutterBottom: TMeasurement;
```

Description

Run-time and Read only.   The gutter sizes are read from the printer driver and vary with each type of printer.   However, they are typically about ¼” (6.35mm).

See Also

GutterLeft, GutterRight, GutterTop

TPagePrinter.GutterLeft

TPagePrinter

The width of the unprintable area at the left of a page in the printer.

```
property GutterLeft: TMeasurement;
```

Description

Run-time and Read only. The gutter sizes are read from the printer driver and vary with each type of printer.   However, they are typically about ¼” (6.35mm).

See Also

GutterBottom, GutterRight, GutterTop

TPagePrinter.GutterRight

TPagePrinter

The width of the unprintable area at the right of a page in the printer.

```
property GutterRight: TMeasurement;
```

Description

Run-time and Read only. The gutter sizes are read from the printer driver and vary with each type of printer.   However, they are typically about ¼” (6.35mm).

See Also

GutterBottom, GutterLeft, GutterTop

TPagePrinter.GutterTop

TPagePrinter

The height of the unprintable area at the top of a page in the printer.

```
property GutterTop: TMeasurement;
```

Description

Run-time and Read only. The gutter sizes are read from the printer driver and vary with each type of printer.   However, they are typically about ¼” (6.35mm).

See Also

GutterBottom, GutterLeft, GutterRight

TPagePrinter.Header

TPagePrinter

The text to be printed at the top of each page.

```
property Header: String;
```

Description

The Header property is a *table string* that is laid out as specified by the HeaderFormat property. For more information see: Table String.

See Also

HeaderFont, FriendlyHeader, Footer

TPagePrinter.HeaderFont

TPagePrinter

The font used when printing the Header.

```
property HeaderFont: TFont;
```

Description

The HeaderFont is only used when printing the Header text.   If you want the HeaderFont to change everytime the Font property changes, set AutoHeaderFont to True.

See Also

FooterFont

TPagePrinter.HeaderFormat

TPagePrinter

The format specifier for the Header property.

```
property HeaderFormat: String;
```

Description

HeaderFormat is a *table format string* that specifies the column widths and alignments to be used for each field in the Header string. For more information see: Table String.

See Also

HeaderFont, FriendlyHeader, FooterFormat

TPagePrinter.LineNumber

TPagePrinter

The number of the last line printed on the current page.

```
property LineNumber: Cardinal;
```

Description

Run-time and Read only.   LineNumber is 0-based and can be thought of as returning the number of lines printed on the current page.

See Also

NewLine, PrevLine

TPagePrinter.Lines

TPagePrinter

A text buffer to store a list of strings to be printed.

```
property Lines: TStrings;
```

Description

The Lines property is typically used to store an existing list of strings (e.g. the lines from a Memo control) for printing.   The WriteLines procedure can then be used to print the contents of the Lines property with one procedure call.

TPagePrinter.LineSpacing

TPagePrinter

Controls the amount of space between each line when printed.

```
property LineSpacing: TLineSpacing default lsSingleSpace;
```

Description

LineSpacing determines how much space is printed between each line of text. LineSpacing can be one of the following values:

Value	Meaning
lsHalfSpace	A half-line width overlapping the previous line.
lsSingleSpace	Immediately after the previous line.
lsSingleAndAHalf	A half-line width between it and the previous line.
lsDoubleSpace	A full-line width between it and the previous line.

TPagePrinter.MarginBottom

TPagePrinter

The distance from the bottom of the page to stop printing.

```
property MarginBottom: TMeasurement;
```

Description

The margins define the printable area used by TPagePrinter. Because of their importance to almost every property and method in TPagePrinter, the margins can't be changed while printing.

See Also

MarginLeft, MarginRight, MarginTop, RefreshProperties

TPagePrinter.MarginLeft

TPagePrinter

The distance from the left of the page to start printing.

```
property MarginLeft: TMeasurement;
```

Description

The margins define the printable area used by TPagePrinter. Because of their importance to almost every property and method in TPagePrinter, the margins can’t be changed while printing.

See Also

MarginBottom, MarginRight, MarginTop, RefreshProperties

TPagePrinter.MarginRight

TPagePrinter

The distance from the right of the page to stop printing.

```
property MarginRight: TMeasurement;
```

Description

The margins define the printable area used by TPagePrinter.   Because of their importance to almost every property and method in TPagePrinter, the margins can’t be changed while printing.

See Also

MarginBottom, MarginLeft, MarginTop, RefreshProperties

TPagePrinter.MarginTop

TPagePrinter

The distance from the top of the page to start printing.

```
property MarginTop: TMeasurement;
```

Description

The margins define the printable area used by TPagePrinter.   Because of their importance to almost every property and method in TPagePrinter, the margins can’t be changed while printing.

See Also

MarginBottom, MarginLeft, MarginRight, RefreshProperties

TPagePrinter.MeasureUnit

TPagePrinter

Determines whether english or metric units are used for TMeasurement properties.

```
property MeasureUnit: TMeasureUnit default muInches;
```

Description

If MeasureUnit is muInches then all TMeasurement properties return their results in Inches.   If MeasureUnit is muMillimeters then all TMeasurement properties return their results in Millimeters.

MeasureUnit also determines the default paper size.   When set to english units, the default paper size is 8.5” x 11”.   When set to metric units, the default paper size is 210mm x 297mm (A4).

When switching MeasureUnits, the margins and other TMeasurement properties should automatically update to display in the selected units.   For example, a 1” margin gets translated to a 25.4mm margin and vice versa.

TPagePrinter.Orientation

TPagePrinter

Orientation determines if the print job prints vertically or horizontally on a page.

```
property Orientation: TPrinterOrientation;
```

Description

Use Orientation to determine if a print job prints in landscape or portrait mode.   Orientation can be one of the following values:

Value	Meaning
poPortrait	The print job prints vertically on the page.
poLandscape	The print job prints horizontally on the page.

TPagePrinter.PageBorderOffset

TPagePrinter

Determine how far above or below the margins the page border lines are printed.

```
property PageBorderOffset: TMeasurement;
```

Description

When PageBorderOffset is 0, the PageBorders are printed exactly on the margins.   However, any text written to the page also begins exactly on the margin, so with no PageBorderOffset, the text will overlap the page borders by a pixel or two.   This tends to look bad.

By setting PageBorderOffset to a small value of about 1/16" (1.5mm), you can prevent the page borders and the text from overlapping each other.

TPagePrinter.PageBorders

TPagePrinter

Determines which edges should have page border lines printed.

```
property PageBorders: TPageBorders default [];
```

Description

By default no page borders are printed.   However, printing top and bottom borders tends to look very nice.

See Also

PageBorderOffset

TPagePrinter.PageCount

TPagePrinter

Returns the number of pages in the document.

```
property PageCount: Cardinal;
```

Description

Run-time and Read only.

See also

PageNumber, Pages

TPagePrinter.PageNumber

TPagePrinter

Allows you to get and set the page number of the document.

```
property PageNumber: Cardinal;
```

Description

Run-time only.   PageNumber is always 0 when there are no pages in the document.   When there are pages, PageNumber begins at 1.   By setting PageNumber to a value between 1 and PageCount, you can control which page is displayed in the preview window.   However, you can’t set PageNumber while printing.

TPagePrinter.Pages

TPagePrinter

Returns a reference to a print page.

```
property Pages[Index: Cardinal]: TPrintPage ;
```

Description

Run-time and Read only.   Because a TPrintPage is really just a TMetafile, you can use any of the properties and methods of TMetafile with a value returned from Pages.   For example, you could use TMetafile.SaveToFile if you wanted to save a copy of a page in native format.

Note: Unlike most indexed properties, Pages is **1-based**.   That is, Pages[1] is the first page in the list, and Pages[PageCount] is the last page in the list.   This way a page's PageNumber will be its index in Pages.

TPagePrinter.PhysicalPageHeight

TPagePrinter

Returns the height of the physical sheet of paper in the printer.

```
property PhysicalPageHeight: TMeasurement;
```

Description

Run-time and Read only.

See Also

AvailablePageHeight, PhysicalPageWidth, PrintableHeight

TPagePrinter.PhysicalPageWidth

TPagePrinter

Returns the width of the physical sheet of paper in the printer.

```
property PhysicalPageWidth: TMeasurement;
```

Description

Run-time and Read only.

See Also

AvailablePageWidth, PhysicalPageHeight, PrintableWidth

TPagePrinter.PrintableHeight

TPagePrinter

Returns the printable height of the page as defined by the margins.

```
property PrintableHeight: TMeasurement;
```

Description

Run-time and Read only. The PhysicalPageHeight, MarginTop, and MarginBottom properties determine PrintableHeight according to the formula:

```
   PrintableHeight:=PhysicalPageHeight-MarginTop-MarginBottom;
```

The largest that PrintableHeight can be is AvailablePageHeight.

See Also

PrintableWidth

TPagePrinter.PrintableWidth

TPagePrinter

Returns the printable width of the page as defined by the margins.

```
property PrintableWidth: TMeasurement;
```

Description

Run-time and Read only. The PhysicalPageWidth, MarginLeft, and MarginRight properties determine PrintableWidth according to the formula:

```
   PrintableWidth:=PhysicalPageWidth-MarginLeft-MarginRight;
```

The largest that PrintableWidth can be is AvailablePageWidth.

See Also

PrintableHeight

TPagePrinter.PrintFromPage

TPagePrinter

Sets the page to start printing from.

```
property PrintFromPage: Cardinal default 0;
```

Description

Run-time only.   When PrintFromPage is 0 (or 1), printing starts with the first page in the document. When PrintFromPage is non-zero, it specifies the page to start printing from when physically printing. PrintFromPage has no effect on the preview window.   PrintFromPage must be less than or equal PrintToPage.

TPagePrinter.Printing

TPagePrinter

Determines whether you are currently printing.

```
property Printing: Boolean;
```

Description

Run-time and Read only.   Printing tells you whether you are currently printing to TPagePrinter's page buffers.   That is, it tells you whether you are between calls to BeginDoc (which turns Printing on) and EndDoc (which turns Printing off).

Printing does not tell you whether you are physically printing because there is no need for it.   The only time TPagePrinter does any physical printing is within the Print method.

TPagePrinter.PrintToFile

TPagePrinter

Determines whether all text printing goes to a file.

```
property PrintToFile: Boolean default False;
```

Description

When PrintToFile is True all text is output to the file specified in the FileName property. When PrintToFile is False all text is output to the print Canvas.

When PrintToFile is True, TPagePrinter behaves as if each page were infinitely long. One implication of this is that if you want page breaks in the output file, you must manually call NewPage. Another implication is that the Footer doesn’t get printed at the bottom of the page. The Footer is printed at the top of the page directly under the Header.

TPagePrinter.PrintToPage

TPagePrinter

Sets the page to stop printing at.

```
property PrintToPage: Cardinal default 0;
```

Description

Run-time only.   When PrintToPage is 0, printing stops with the last page in the document.   When PrintToPage is non-zero, it specifies the page to stop printing at when physically printing.   PrintToPage has no effect on the preview window.   PrintToPage must be greater than or equal PrintFromPage.

TPagePrinter.ProgressMessage

TPagePrinter

The status message to display on the print progress dialog.

```
property ProgressMessage: String;
```

Description

Lets you specify the status message the user will see in the progress dialog when physically printing pages.   If ProgressMessage is an empty string, then the text of the SendingPagesMsg constant is used as the status string.

If ShowProgress is False, ProgressMessage is unused.

TPagePrinter.ShadowColor

TPagePrinter

Determines the shadow color for the preview page.

```
property ShadowColor: TColor default clBtnShadow;
```

Description

The color used to draw the shadow effect in the preview window.   ShadowColor has no effect on physically printed output.

See Also

ShadowOffset

TPagePrinter.ShadowOffset

TPagePrinter

Determines how large the shadow should be.

```
property ShadowOffset: TPixels default 5;
```

Description

ShadowOffset determines how many pixels to the right and bottom of the preview page the shadow effect should extend to.   By setting ShadowOffset to 0, you can disable the painting of any shadow. ShadowOffset has no effect on physically printed output.

See Also

ShadowColor

TPagePrinter.ShowCancel

TPagePrinter

Determines whether the progress dialog has a Cancel button.

```
property ShowCancel: Boolean default True;
```

Description

ShowCancel has no effect if ShowProgress is False. If ShowProgress is True and ShowCancel is True, then progress dialog has a Cancel button that can be used to cancel printing to the printer. If ShowCancel is False the progress dialog has no Cancel button.

Note: If ShowCancel is True and ShowProgress is True, then the Print function will call Application.ProcessMessages before printing each page to determine if the Cancel button has been pressed. ProcessMessages is not called if either of these properties are False.

TPagePrinter.ShowMargins

TPagePrinter

Determines whether the margins are shown in the preview window.

```
property ShowMargins: Boolean default True;
```

Description

If ShowMargins is True, a dashed gray rectangle is drawn on the preview page where the margins would be.   ShowMargins has no effect on physically printed output.

See Also

MarginBottom, MarginLeft, MarginRight, MarginTop

TPagePrinter.ShowProgress

TPagePrinter

Determines whether a progress dialog is shown while physically printing.

```
property ShowProgress: Boolean default True;
```

Description

If ShowProgress is True, a progress dialog is displayed while pages are being sent to the printer. That is, a progress dialog is displayed during a call to Print.

Note: Regardless of the ShowProgress setting, no progress dialog is displayed while pages are being buffered in TPagePrinter (i.e. between BeginDoc and EndDoc). TPagePrinter has no way of knowing how many pages you are going to send to it, so it can't display a progress indicator before EndDoc is called. If you need a progress dialog during this phase of printing, you need to create one specific to your application. Display your custom dialog before calling BeginDoc, update it while sending data to TPagePrinter, and then close it after the call to EndDoc or Print.

See Also

ProgressMessage, ShowCancel

TPagePrinter.TableFormat

TPagePrinter

Determines the layout of columns in table lines.

```
property TableFormat: String;
```

Description

Table Format is a *table format string* that specifies the column widths and alignments to be used for each field in lines printed with WriteTableLine. For more information see: Table String.

See Also

DefaultColWidth, TokenSeparator

TPagePrinter.TableGrid

TPagePrinter

Determines whether grid lines are printed on table lines.

```
property TableGrid: Boolean default False;
```

Description

If TableGrid is True, each column in a line printed with WriteTableLine is surrounded by grid lines.   If TableGrid is False, no grid lines are printed.

TPagePrinter.TabSize

TPagePrinter

Determines the width of a tab character in spaces.

```
property TabSize: Cardinal default 8;
```

Description

Before tab characters are printed they must be expanded into spaces.   TabSize controls how many spaces are substituted for each tab character.

TPagePrinter.Title

TPagePrinter

The title to use when a job is sent to the printer.

```
property Title: String;
```

Description

The Title property determines the description that will be displayed for your print job when it is physically sent to the printer.   It also determines the caption that is displayed on the print progress dialog (if ShowProgress is True).   Internally, TPagePrinter.Title gets and sets TPrinter.Title.

TPagePrinter.TokenSeparator

TPagePrinter

The separator character to use between columns in table strings.

```
property TokenSeparator: Char default ‘|’;
```

Description

TokenSeparator defaults to the vertical bar character ‘**|**’, but you can set it to another character if you need to use the vertical bar character in a column’s text.

TPagePrinter.WordWrap

TPagePrinter

Determines whether long lines wrap when printed.

```
property WordWrap: Boolean default True;
```

Description

If WordWrap is True, then lines longer than the page width are wrapped and printed on multiple lines.
If WordWrap is False, then lines longer than the page width are clipped at the right margin.

TPagePrinter.ZoomLocation

TPagePrinter

Determines where the preview window focuses when ZoomPercent is changed.

```
property ZoomLocation: TZoomLocation default zlTopLeft;
```

Description

ZoomLocation determines which part of the preview page is focused on when ZoomPercent changes. ZoomLocation can be one of the following values:

Value	Meaning
zlTopLeft	The top left corner of the page is zoomed in on.
zlTopCenter	The top center of the page is zoomed in on.
zlCenter	The center of the page is zoomed in on.

TPagePrinter.ZoomPercent

TPagePrinter

Determines the scale that the preview page is displayed at.

```
property ZoomPercent: Cardinal default 25;
```

Description

ZoomPercent determines how much the preview page is zoomed in on.   When ZoomPercent is 100, the preview page is shown actual size.

See Also

ZoomLocation

TPagePrinter.BeginDoc

TPagePrinter

Begins a new document for printing.

```
procedure BeginDoc;
```

Description

BeginDoc will create a new, blank canvas for printing on.   It does not send the print job to a physical printer though.   To do that you must call Print.

See Also

EndDoc

TPagePrinter.BeginUpdate

TPagePrinter

Turns of screen repainting for the preview window.

```
procedure BeginUpdate;
```

Description

BeginUpdate prevents the screen from being repainted until the EndUpdate method is called. Use BeginUpdate to prevent screen repaints and to speed processing time while you are changing multiple properties of a preview.

TPagePrinter.Clear

TPagePrinter

Clears any current page list.

```
procedure Clear;
```

Description

Clear resets any buffered page list.   Use this method to free up memory when you are done with a printed document (i.e. after you finish previewing it or after you finish physically printing it).

TPagePrinter.EndDoc

TPagePrinter

Finishes printing for a document.

```
procedure EndDoc;
```

Description

The EndDoc method ends the current job being sent to TPagePrinter. After the application calls EndDoc, the preview window displays the first page printed.

See Also

BeginDoc

TPagePrinter.EndUpdate

TPagePrinter

Re-enables screen repainting.

```
procedure EndUpdate;
```

Description

The EndUpdate method re-enables screen repainting that was turned off with the BeginUpdate method.

TPagePrinter.ExpandFriendlyFormat

TPagePrinter

Expands a friendly format string.

```
procedure ExpandFriendlyFormat(const UserFmt: String; AsHeader: Boolean);
```

Description

Protected.   ExpandFriendlyFormat parses a friendly format string and converts it into Header (or Footer) and HeaderFormat (or FooterFormat) strings.

TPagePrinter.ExpandLogicalFields

TPagePrinter

Expands logical fields in headers and footers.

```
function ExpandLogicalFields(S: String): String;
```

Description

Protected.   ExpandLogicalFields expands the constants {$LINE}, {$PAGE}, {$DATE}, {$TIME}, and {$TITLE} in the Header, Footer, and table lines.

TPagePrinter.GetClippedLine

TPagePrinter

Clips a line to the specified width.

```
function GetClippedLine(const Line: String; const Width: TPixels): String;
```

Description

Protected. GetClippedLine returns the part of Line that will fit in the specified Width using the current Font.

See Also

SplitLine

TPagePrinter.GetPreviewPagePixels

TPagePrinter

Determines the size of the preview page in pixels.

```
function GetPreviewPagePixels(Horz: Boolean): TPixels;
```

Description

Protected. Determines the size of the preview page when drawn on the screen based on ZoomPercent.

TPagePrinter.GetPrinterHandle

TPagePrinter

Returns the printer handle.

```
function GetPrinterHandle: HDC;
```

Description

Protected.   Used to get the printer handle while protected by an exception handler.   This way if no printers are installed, printer exceptions won't get thrown and prevent your application from running.

TPagePrinter.GetScaleFactor

TPagePrinter

Returns the printer to screen scale factor.

```
function GetScaleFactor(Horz: Boolean): Double;
```

Description

Protected. GetScaleFactor returns the ratio of printer canvas pixels per inch to screen canvas pixels per inch.

TPagePrinter.MeasureUnitsToPixelsH

TPagePrinter

Converts measure units to horizontal pixels.

```
function MeasureUnitsToPixelsH(const M: TMeasurement): TPixels;
```

Description

MeasureUnitsToPixelsH converts measure units to pixels based on the horizontal resolution of the current printer.

See Also

MeasureUnitsToPixelsV

TPagePrinter.MeasureUnitsToPixelsV

TPagePrinter

Converts measure units to vertical pixels.

```
function MeasureUnitsToPixelsV(const M: TMeasurement): TPixels;
```

Description

MeasureUnitsToPixelsV converts measure units to pixels based on the vertical resolution of the current printer.

See Also

MeasureUnitsToPixelsH

TPagePrinter.MeasureUnitsToScreenPixels

TPagePrinter

Converts measure units to screen pixels

```
function MeasureUnitsToScreenPixels(const Value: TMeasurement; Horz:
Boolean): TPixels;
```

Description

Protected.   Converts measure units to screen pixels based on the current screen resolution.

TPagePrinter.NewLine

TPagePrinter

Creates a new line in the document.

```
function NewLine: Cardinal;
```

Description

NewLine will advance the CanvasPosition to a new line.   If there is no more room on the current page, then NewPage is called implicitly.

NewLine returns the current LineNumber.

See Also

OnNewLine

TPagePrinter.NewPage

TPagePrinter

Creates a new page in the document.

```
function NewPage: Cardinal;
```

Description

NewPage will create a new page in the current document.   NewPage returns the current PageNumber.

See Also

OnNewPage

TPagePrinter.PaintPreview

TPagePrinter

Paints the preview window.

```
procedure PaintPreview(Sender: TObject); virtual;
```

Description

Protected.   PaintPreview is the paint procedure for the preview window.

TPagePrinter.ParseFormatToken

TPagePrinter

Parses format tokens.

```
procedure ParseFormatToken(var CurToken: String; var CurAlignment:
TAlignment; var CurWidth: TMeasurement);
```

Description

Protected.   ParseFormatToken splits a format token from a table format string into its alignment and width.   If the token is invalid, left alignment and DefaultColWidth are returned.

TPagePrinter.PixelsToMeasureUnitsH

TPagePrinter

Converts pixels to horizontal measure units.

```
function PixelsToMeasureUnitsH(const P: TPixels): TMeasurement;
```

Description

PixelsToMeasureUnitsH converts pixels to measure units based on the horizontal resolution of the current printer.

See Also

PixelsToMeasureUnitsV

TPagePrinter.PixelsToMeasureUnitsV

TPagePrinter

Converts pixels to vertical measure units.

```
function PixelsToMeasureUnitsV(const P: TPixels): TMeasurement;
```

Description

PixelsToMeasureUnitsV converts pixels to measure units based on the vertical resolution of the current printer.

See Also

PixelsToMeasureUnitsH

TPagePrinter.PrevLine

TPagePrinter

Moves to the previous line in the document.

```
function PrevLine: Boolean;
```

Description

PrevLine tries to move the CanvasPosition to the previous line in the document. If successful, PrevLine returns True; otherwise, it returns False. PrevLine should only fail at the top of a page.

TPagePrinter.Print

TPagePrinter

Sends the current document to the printer.

```
function Print: Boolean;
```

Description

Print sends a page list to the current printer. The exact behavior of Print is based upon several settings: AbortOnCancel, Copies, Collate, PrintFromPage, PrintToPage, ProgressMessage, ShowCancel, ShowProgress, and Title.

Print returns False if printing was cancelled; it returns True otherwise. Note that printing can only be cancelled if ShowCancel is True and ShowProgress is True. If both ShowCancel and ShowProgress are True, then Print uses Application.ProcessMessages to determine whether the Cancel button has been pressed.

TPagePrinter.RefreshProperties

TPagePrinter

Refreshes important properties of TPagePrinter.

```
procedure RefreshProperties;
```

Description

RefreshProperties makes sure margins, headers, footers, linespacing, and other values have valid values based on the current printer settings.   You will typically want to call it after something external (e.g. a Print Setup dialog) may have changed properties of the current printer.

TPagePrinter.ScaleValue

TPagePrinter

Scales a value for the preview window.

```
function ScaleValue(Value: TMeasurement; Horz: Boolean): TPixels;
```

Description

ScaleValue is used by PaintPreview to scale a measure unit to screen pixels and then scale it again based on ZoomPercent.

TPagePrinter.SetPixelsPerInch

TPagePrinter

Makes sure the printer font's PixelsPerInch property is set right.

```
procedure SetPixelsPerInch;
```

Description

Protected. SetPixelsPerInch is used to get around the VCL's tiny font bug. For fonts to render correctly on a printer canvas, their PixelsPerInch property must be explicitly set *after* printing begins.

TPagePrinter.SplitLine

TPagePrinter

Splits a line of text intelligently.

```
procedure SplitLine(var CurLine: String; var Buffer: String; const
ClipWidth: TPixels; const TrimLastWhiteSpace: Boolean);
```

Description

Protected.   SplitLine is used to clip a line to a specified width and then clip back even farther so that words aren't split in the middle.   If necessary, it can also trim off a dangling whitespace character, so it won't be printed on the next line.

See Also

SplitLineAndPrint

TPagePrinter.SplitLineAndPrint

TPagePrinter
Splits a line of text repeatedly until it is all printed.

```
procedure SplitLineAndPrint(const Line: String; UseWrite: Boolean);
```

Description

Protected.   SplitLineAndPrint splits a line of text and prints each part until the entire line is printed. The UseWrite parameter determines whether Write is used to print the line segments instead of WriteLine.

See Also

SplitLine

TPagePrinter.UpdateDesigner

TPagePrinter

Tells the form designer that some properties have been modified.

```
procedure UpdateDesigner;
```

Description

Protected.   UpdateDesigner is used to tell the IDE's form designer that one or more properties have been modified, and that the object inspector and form may need to be updated to reflect these changes.

TPagePrinter.UpdatePagePreviewSize

TPagePrinter

Updates the preview window when the zoom percent changes.

```
procedure UpdatePagePreviewSize;
```

Description

Protected.   UpdatePagePreviewSize updates the preview window’s scrollbars and ZoomLocation when ZoomPercent changes.

TPagePrinter.UpdateProgressDlg

TPagePrinter

Updates the progress dialog when some print settings change.

```
procedure UpdateProgressDlg(const Status: String);
```

Description

Protected.   UpdateProgressDlg updates the progress bar, progress message, current page, and visibility of the progress dialog during physical printing.

TPagePrinter.ValidateFormatString

TPagePrinter

Makes sure the format string contains valid tokens.

```
function ValidateFormatString(const Fmt: String; const ConvertUnits:
Boolean): String;
```

Description

Protected.   ValidateFormatString ensures that each format token in a format string contains an alignment character and width.

TPagePrinter.Write

TPagePrinter

Writes a line of text and leaves the position at the end of the line.

```
procedure Write(const Line: String);
```

Description

Write prints a line of text and leaves the CanvasPosition at the end of the line.   Thus, subsequent calls to Write will begin printing where the previous line left off.

Write ignores the Alignment property.   It always prints left justified relative to the CanvasPosition.

See Also

WriteLine

TPagePrinter.WriteLine

TPagePrinter

Writes a line of text and advances the position to a new line.

```
procedure WriteLine(const Line: String);
```

Description

WriteLine prints a line of text and calls NewLine.

See Also

Write

TPagePrinter.WriteLineAligned

TPagePrinter

Writes a line of text with a specific alignment.

```
procedure WriteLineAligned(const AAlignment: TAlignment; const Line:
String);
```

Description

WriteLineAligned behaves exactly like WriteLine except you can explicitly specify the alignment to print with instead of having to implicitly use the Alignment property.

TPagePrinter.WriteLines

TPagePrinter

Writes the contents of the Lines property.

```
procedure WriteLines(const LinesAsTable: Boolean);
```

Description

WriteLines prints each line of text in the Lines property.   If the LinesAsTable parameter is false, then the lines are printed with WriteLine.   If LinesAsTable is True, then WriteTableLine is used to print each line as a table string.

TPagePrinter.WriteTableGrid

TPagePrinter

Draws the grid around each cells.

```
procedure WriteTableGrid(const CurWidth: TPixels; const TopGrid, BottomGrid:
Boolean);
```

Description

Protected. WriteTableGrid draws the grid lines around a cell in a table line. It is called implicitly by WriteTableLine.

TPagePrinter.WriteTableLine

TPagePrinter

Writes a line as a table string.

```
procedure WriteTableLine(const Line: String);
```

Description

WriteTableLine prints a formatted line of text as a table string.

See Also

TableFormat

TPagePrinter.ZoomToFit

TPagePrinter

Zooms the preview window to fit an entire page.

```
procedure ZoomToFit;
```

Description

ZoomToFit will recalculate ZoomPercent to fit an entire page into the preview window.

See Also

ZoomToHeight, ZoomToWidth

TPagePrinter.ZoomToHeight

TPagePrinter

Zooms the preview window to fit a page’s height.

```
procedure ZoomToHeight;
```

Description

ZoomToHeight will recalculate ZoomPercent to fit a page’s entire height into the preview window.

See Also

ZoomToFit, ZoomToWidth

TPagePrinter.ZoomToWidth

TPagePrinter

Zooms the preview window to fit a page's width.

```
procedure ZoomToWidth;
```

Description

ZoomToWidth will recalculate ZoomPercent to fit a page's entire width into the preview window.

See Also

ZoomToFit, ZoomToHeight

TPagePrinter.OnNewLine

TPagePrinter

An event that fires before each new line.

```
property OnNewLine: TNotifyEvent;
```

Description

OnNewLine fires before each new line and at the top of each new page.   When it fires at the top of a new page, it fires after the Header, Footer, and PageBorders are printed.

See Also

OnNewPage

TPagePrinter.OnNewPage

TPagePrinter

An event that fires before each new page.

```
property OnNewPage: TNotifyEvent;
```

Description

OnNewLine fires before anything is printed on the page.   Thus, you can use it to change the Header, Footer, and PageBorders before they are printed on each page.

See Also

OnNewLine

EPagePrinter

TPagePrinter

The exception class used by TPagePrinter.

```
EPagePrinter = class(EPrinter);
```

Description

EPagePrinter is a decendent of EPrinter and is thrown by TPagePrinter whenever there is an exception not directly related to the physical printer (e.g. trying to change margins while printing). When there is a physical printing error, an EPrinter exception is still thrown.

TGradientOrientation

TPagePrinter

Defines values for the orientation of the gradient.

```
TGradientOrientation = (goHorizontal, goVertical);
```

Description

TGradientOrientation is the type of the Orientation parameter for the FillGradient procedure.   It determines the direction the gradient is drawn.

TLineSpacing

TPagePrinter

Defines values for the LineSpacing property.

```
TLineSpacing = (lsHalfSpace, lsSingleSpace, lsSingleAndAHalf,
lsDoubleSpace);
```

Description

TLineSpacing is the type of the LineSpacing property.

TMeasurement

TPagePrinter

The type for any unit measurements.

```
TMeasurement = Double;
```

Description

TMeasurement is the type used for any measurements using english or metric units.

TMeasureUnit

TPagePrinter

Defines values for the MeasureUnit property.

```
TMeasureUnit = (muInches, muMillimeters);
```

Description

TMeasureUnit is the type for the MeasureUnit property.

TPageBorder

TPagePrinter

Defines values for the PageBorders property.

```
TPageBorder = (pbTop, pbBottom, pbLeft, pbRight);
```

Description

TPageBorder is the type of the elements in the PageBorders property.

See Also

TPageBorders

TPageBorders

TPagePrinter

The set type for the PageBorders property.

```
TPageBorders = set of TPageBorder;
```

Description

TPageBorders is the type of the PageBorders set.

See Also

PageBorder

TPageList

TPagePrinter

The type of the Pages property.

```
TPageList = class(TList);
```

Description

TPageList is the type of the Pages property.

TPixels

TPagePrinter

The type for any pixel measurements.

```
TPixels = Cardinal;
```

Description

TPixels is the type used for any measurements involving pixels.

TPrintCanvas

TPagePrinter

The type of the Canvas property.

```
TPrintCanvas = TMetafileCanvas;
```

Description

TPrintCanvas is the type of the Canvas property used by TPagePrinter.   Because it is a descendent of TMetafileCanvas, it has all of the methods and properties available in TCanvas and TMetafileCanvas.

TPrintPage

TPagePrinter

The type of the pages in the Pages property.

```
TPrintPage = TMetafile;
```

Description

TPrintPage is the type of the pages in the Pages property.   Because it is a descendent of TMetafile, it has all the methods available of that type.

TZoomLocation

TPagePrinter

Defines values for the ZoomLocation property.

```
TZoomLocation = (zlTopLeft, zlTopCenter, zlCenter);
```

Description

TZoomLocation is the type of the ZoomLocation property.

ExpandTabsAsSpaces

TPagePrinter

Expands tabs as spaces in a string.

```
function ExpandTabsAsSpaces(const S: String; const TabSize: Cardinal):
String;
```

Description

ExpandTabsAsSpaces will expand every tab in S into the number of spaces indicated by TabSize.

FillGradient

TPagePrinter

Draws a gradient on a canvas.

```
procedure FillGradient(Canvas: TCanvas; Rc: TRect; LeftTopColor,
RightBottomColor: TColor; Orientation: TGradientOrientation);
```

Description

FillGradient will fill the rectangle given by Rc with a gradient in the specified direction.

GenSpace

TPagePrinter

Generates a string of spaces.

```
function GenSpace(const Size: Cardinal): String;
```

Description

GenSpace returns a string with the number of spaces indicated by Size.

ReplaceSubString

TPagePrinter

Replaces a substring with another string.

```
function ReplaceSubString(const OldSubStr, NewSubStr: String; S: String):
String;
```

Description

ReplaceSubString replaces every instance of OldSubStr in S with NewSubStr.

StripBackToWhiteSpace

TPagePrinter

Strips any trailing non-whitespaces characters.

```
function StripBackToWhiteSpace(const S: String): String;
```

Description

StripBackToWhiteSpace strips off any trailing non-whitespace characters from S.   It is typically used after GetClippedLine to strip off any partial words at the end of a clipped string.

TokenizeString

TPagePrinter

Splits a string into separate tokens.

```
procedure TokenizeString(const S: String; TokenSeparator: Char; Tokens: TStringList);
```

Description

TokenizeString splits a string S containing TokenSeparator characters into separate tokens which are returned in the Tokens parameter.

Table String

TPagePrinter

A formatted string which defines a table line.

Examples

Table Format String:	`<2.5|^-2.5|>2.5`
Text Table String:	`Title: {$TITLE}\|{$DATE} {$TIME}\|Page {$PAGE}`

Description

There are two types of table strings: table format strings and text table strings.

A table format string is made up of zero or more column tokens. Each column token must contain an alignment character and a column width. An alignment character is either a ‘<’ for left alignment, ‘^’ for center alignment, or ‘>’ for right alignment. A column width must be a TMeasurement. TokenSeparator determines the character which must be between each distinct column token.

Text table strings are just strings concatenated together separated by TokenSeparators. When a text table string is printed using WriteTableLine, each token in the text table string is printed in a column whose alignment and width are determined by the corresponding token in the TableFormat property.

If TableGrid is True then negative column widths can be used to force the grid to be turned off (i.e. not printed) for a specific column. If WordWrap is True then table strings will wrap to multiple lines as necessary. You can force a line break in a table string by embedding a linefeed character (ASCII 10) in a token.

In table strings, there are five logical constants which are expanded. {$LINE} is expanded to the current line number, {$PAGE} is expanded to the current page number, {$DATE} is expanded to the current date (using ShortDateFormat), {$TIME} is expanded to the current time (using LongTimeFormat), and {$TITLE} is expanded to the contents of the Title property.

See Also

DefaultColWidth, ValidateFormatString

Friendly Format Specifiers

TPagePrinter

Specifies alignment and content for headers and footers using simple codes.

Alignment	
&l	Left Align
&c	Center Align
&r	Right align

Content	
&f	File name only
&F	File name and path
&d	Short date
&t	Short time
&D	Long date
&T	Long time
&l	Title
&p	Page number
&&	Ampersand (&)