

## ReportWorks

Properties

Methods

Events

The ReportWorks component is designed to provide fast, simple, hasslefree reporting functionality. Reports can be previewed, printed to the printer or printed to file. Columns can be established in the form of tabs with left, right and center justification. Lines, or rows, can be shaded, if so desired.

Output can be printed with or without linefeed, left, right or center justified. Large strings, longer than 255 characters, can be output with automatic word-wrap within the established page margins.

Printer page units can be set to Inches, Millimeters or Points. Once margins are set, all printing occurs within these boundaries.

Graphics functionality is available, and bitmaps, icons and metafiles can be imported to the page.

Copyright

## Properties

In addition to the properties inherited from the **TGraphicsControl**, the ReportWorks component introduces the following:

### > RunTime Only

|                     |                     |                    |
|---------------------|---------------------|--------------------|
| <u>BottomMargin</u> | <u>LinesPerInch</u> | <u>RightMargin</u> |
| <u>CurrentPage</u>  | <u>LineWidth</u>    | <u>TextStyle</u>   |
| <u>Destination</u>  | <u>Orientation</u>  | <u>Title</u>       |
| <u>FileName</u>     | <u>PageBorder</u>   | <u>TopMargin</u>   |
| <u>LeftMargin</u>   | > <u>PageNum</u>    | <u>Units</u>       |
| <u>LineMeasure</u>  | > <u>PageTotal</u>  | <u>ZoomPercent</u> |

## Methods

In addition to the methods inherited from the **TGraphicsControl**, the ReportWorks component introduces the following:

|                                       |  |  |
|---------------------------------------|--|--|
| <u><a href="#">ClearTabs</a></u>      | <u><a href="#">LineCenter</a></u>      | <u><a href="#">PrintRight</a></u>        |
| <u><a href="#">DrawGraphic</a></u>    | <u><a href="#">LineEnd</a></u>         | <u><a href="#">PrintTab</a></u>          |
| <u><a href="#">DrawLine</a></u>       | <u><a href="#">LineStart</a></u>       | <u><a href="#">Rectangle</a></u>         |
| <u><a href="#">Ellipse</a></u>        | <u><a href="#">LineTo</a></u>          | <u><a href="#">ResetPagePosition</a></u> |
| <u><a href="#">EndDoc</a></u>         | <u><a href="#">NewPage</a></u>         | <u><a href="#">SetBrush</a></u>          |
| <u><a href="#">FirstPage</a></u>      | <u><a href="#">NextLine</a></u>        | <u><a href="#">SetBrushDefault</a></u>   |
| <u><a href="#">FontBold</a></u>       | <u><a href="#">NextPage</a></u>        | <u><a href="#">SetFont</a></u>           |
| <u><a href="#">FontItalic</a></u>     | <u><a href="#">Paint</a></u>           | <u><a href="#">SetPagePosition</a></u>   |
| <u><a href="#">FontRegular</a></u>    | <u><a href="#">PositionGraphic</a></u> | <u><a href="#">SetPen</a></u>            |
| <u><a href="#">FontStrikeout</a></u>  | <u><a href="#">PreviousLine</a></u>    | <u><a href="#">SetPenDefault</a></u>     |
| <u><a href="#">FontUnderline</a></u>  | <u><a href="#">PreviousPage</a></u>    | <u><a href="#">SetTab</a></u>            |
| <u><a href="#">GoToPageBottom</a></u> | <u><a href="#">Print</a></u>           | <u><a href="#">ShadeLine</a></u>         |
| <u><a href="#">GoToPageTop</a></u>    | <u><a href="#">PrintCenter</a></u>     | <u><a href="#">StartDoc</a></u>          |
| <u><a href="#">GoToXY</a></u>         | <u><a href="#">PrintLeft</a></u>       | <u><a href="#">StretchGraphic</a></u>    |
| <u><a href="#">IsPageEnd</a></u>      | <u><a href="#">PrintLine</a></u>       |  |
| <u><a href="#">LastPage</a></u>       | <u><a href="#">PrintParagraph</a></u>  |  |

## Events

ReportWorks has only one event.

OnPrint

## Copyright

### **ReportWorks Version 1.00**

Copyright (c) 1995

All rights reserved

Howard S. Walsh

The unregistered version of ReportWorks will work only while Delphi is running.

Registered users of ReportWorks will receive a royaltyfree version that will run without Delphi. Registered users will also receive the full source code. The source code may also be altered to suit the registered users need. However, I request that I be made aware of any improvements or additions to the source code.

The only restriction to registered users is that they cannot resell the component as their own, whether it has been altered or not.

## BottomMargin Property

**Applies to**  
ReportWorks

**Declaration**  
**property** BottomMargin : Single;

**Description**  
The BottomMargin property allows the user to set the printer page bottom margin relative to the actual page length, not printing area. The BottomMargin may be set in Inches, Millimeters or Points.

**Example**  
The following example sets the page bottom margin to one inch.

```
Units := ulnch;  
BottomMargin := 1;
```

## CurrentPage Property

**Applies to**  
ReportWorks

**Declaration**  
**property** CurrentPage : Integer;

### Description

The CurrentPage property allows the user to set which report page will show in the preview screen. Normally, this is initially set to 1 in the Object Inspector. Once the preview screen is shown, maneuvering among the pages should be accomplished by using [FirstPage](#), [PreviousPage](#), [NextPage](#) and [LastPage](#) instead of by setting **CurrentPage**.

Setting CurrentPage to a value greater than the total number of pages in the report will result in a blank preview page.

### Example

The following example sets the current preview page to page 2.

```
Procedure Example.SampleReport;
begin
    with ReportWorks1 do
    begin
        StartDoc;
        .
        .
        .
        EndDoc;
    end;
end;

procedure Example.PageTwo
begin
    ReportWorks1.Destination := dPreview;
    ReportWorks1.CurrentPage := 2;
    SampleReport;
end;
```

## Destination Property

**Applies to**  
ReportWorks

**Declaration**  
**property** Destination : TDestination;

**Description**  
The Destination property directs the report output to the preview screen, printer or file. The property takes a value of TDestination which is either **dPreview**, **dPrinter** or **dFile**.

| Value    | Description                               |
|----------|---|
| dPreview | Output is directed to the preview screen. |

dPrinterOutput is directed to the printer. ReportWorks encapsulates the Printer object.

|       |   |
|-------|---|
| dFile | Output is directed to a file. If <u>FileName</u> is not specified, ReportWorks uses quickrpt.prn. |
|-------|---|

**Example**  
The following example directs output to the printer object.

**Destination := dPrinter;**



## FileName Property

**Applies to**  
ReportWorks

**Declaration**  
**property** FileName : String;

**Description**  
The FileName property allows the user to specify a file name to use when the Destination is set to **dFile**.

**Example**  
The following example directs output to a file.

```
FileName := Report1.prn  
Destination := dFile;  
StartDoc;
```

## LeftMargin Property

**Applies to**  
ReportWorks

**Declaration**  
**property** LeftMargin : Single;

**Description**  
The LeftMargin property allows the user to set the printer page left-hand margin relative to the actual page length, not printing area. The LeftMargin may be set in Inches, Millimeters or Points.

**Example**  
The following example sets the page left margin to one inch.

```
Units := ulInch;  
LeftMargin := 1;
```

## LineMeasure Property

**Applies to**  
ReportWorks

**Declaration**  
**property** LineMeasure : TLineMeasure;

**Description**  
The LineMeasure property determines how the report lines, or rows, are measured. The property takes a value of **TLineMeasure** which is either **ImTextHeight** or **ImLinesPerInch**.

| Value          | Description  |
|----------------|--|
| ImTextHeight   | Lines are measured by the current font height.   |
| ImLinesPerInch | Number of lines per inch is determined by the value of <u><a href="#">LinesPerInch</a></u> property. |

**Example**  
The following example sets the number of lines per inch to 5.

```
LinesPerInch := 5  
LineMeasure := ImLinesPerInch;
```

## LinesPerInch Property

**Applies to**  
ReportWorks

**Declaration**  
**property** LinesPerInch : integer;

**Description**  
The LinesPerInch property sets the number of lines in one vertical inch when LineMeasure is set to **ImLinesPerInch**.

**Example**  
The following example sets the number of lines per inch to 5.

```
LinesPerInch := 5  
LineMeasure := ImLinesPerInch;
```

## LineWidth Property

**Applies to**  
ReportWorks

**Declaration**  
**property** LineWidth : TLineWidth;

**Description**  
The LineWidth property set the thickness of lines drawn by the [DrawLine](#) method. The property takes a value of **TLineWidth**, which is one of the following: **lwDefault**, **lwFine**, **lwNormal**, **lwDouble**, **lwThick** or **lwHeavy**.

| Value     | Description   |
|-----------|---|
| lwDefault | Line thickness is determined by the current page pen width. |
| lwFine    | Line thickness is approximately 1/100 of an inch.           |
| lwNormal  | Line thickness is approximately 1/50 of an inch.            |
| lwDouble  | Line thickness is approximately 1/25 of an inch.            |
| lwThick   | Line thickness is approximately 1/15 of an inch.            |
| lwHeavy   | Line thickness is approximately 1/10 of an inch.            |

**Example**  
The following example sets the line width to lwDouble.

```
LineWidth := lwDouble;  
DrawLine;
```

## Orientation Property

**Applies to**  
ReportWorks

**Declaration**  
**property** Orientation : TPrinterOrientation;

### Description

The LineMeasure property sets the report page orientation to either Portrait or Landscape. The property takes a value of **TPrinterOrientation**, which is a Delphi type of either **poPortrait** or **poLandscape**.

Orientation should not be changed in the middle of a printed page, but can be changed before the document is started or before a new page is started.

### Example

The following example sets the printer orientation to Landscape.

```
Orientation := poLandscape;  
StartDoc;
```

## PageBorder Property

**Applies to**  
ReportWorks

**Declaration**  
**property** PageBorder : Boolean;

**Description**  
The PageBorder takes a boolean value which determines if a black border is drawn around the preview page.

**Example**  
The following example will draw a black border around the preview page.

```
PageBorder := 1;  
Destination := dPreview  
StartDoc;
```

## RightMargin Property

**Applies to**  
ReportWorks

**Declaration**  
**property** RightMargin : Single;

**Description**  
The RightMargin property allows the user to set the printer page right-hand margin relative to the actual page length, not printing area. The RightMargin may be set in Inches, Millimeters or Points.

**Example**  
The following example sets the page right margin to one inch.

```
Units := ulnch;  
RightMargin := 1;
```



## TextStyle Property

**Applies to**  
ReportWorks

**Declaration**  
**property** TextStyle : TTextStyle;

### Description

The TextStyle property determines if text is printed as Transparent or Opaque (default). The property takes a value of **TTextStyle**, which is either **tsDefault** or **tsTransparent**.

The default value, **tsDefault**, will print text either as Opaque or Transparent or other depending on the current setting of the Report Brush Style.

### Example

The following example allows text to print properly over a yellow shaded line.

```
ShadeLine(clYellow);  
TextStyle := tsTransparent;  
PrintLeft(This is a test.);
```

## Title Property

**Applies to**  
ReportWorks

**Declaration**  
**property** Title : String;

**Description**  
The Title property determines the document name to be used by the Print Manager when the report prints to the printer. If Title is not specified, then the Print Manager uses Untitled.

**Example**  
The following example sets the report title to Car Expenses.

```
Title := Car Expenses;  
StartDoc;
```

## TopMargin Property

**Applies to**  
ReportWorks

**Declaration**  
**property** TopMargin : Single;

**Description**  
The TopMargin property allows the user to set the printer page top margin relative to the actual page length, not printing area. The TopMargin may be set in Inches, Millimeters or Points.

**Example**  
The following example sets the page top margin to one inch.

```
Units := uInch;  
TopMargin := 1;
```

## Units Property

**Applies to**  
ReportWorks

**Declaration**  
**property** Units : TUnits;

### Description

The Units property determines if ReportWorks used inches, millimeters, or points as its measurement unit. ReportWorks can also use the Report Canvas pixel as measurement, however, if the report is printed to a preview screen the output may differ from the report printed to a printer. Due to this restriction, the value, **uDefault**, should not be used unless the output is to be directed to only one destination.

The property takes one of the following values:

| Value    | Description  |
|----------|--|
| uInches  | Output is measured in inches.                            |
| uMM      | Output is measured in millimeters.                       |
| uPoints  | Output is measured in points (about 72 points per inch). |
| UDefault | Output is measured in pixels.                            |

### Example

The following example sets the unit measurement to inches.

```
Units := uInch;  
RightMargin := 1;
```

## ZoomPercent Property

**Applies to**  
ReportWorks

**Declaration**  
**property** ZoomPercent : Integer;

**Description**  
The ZoomPercent property determines the size of the preview screen. The default value is 100. Fastest screen redraw occurs when the default value is used. If this value is set to anything other than 100, ReportWorks uses StretchDraw to repaint the screen and this method is slower than Draw.

**Example**  
The following example doubles the preview page size.

```
ZoomPercent := 200;
```

## PageNum Property

**Applies to**  
ReportWorks

**Declaration**  
**property** PageNum : Integer;

**Description**  
The PageNum property returns the current printing page. Use this property to indicate which page is the current preview page.

## PageTotal Property

**Applies to**  
ReportWorks

**Declaration**  
**property** PageTotal : Integer;

### **Description**

The PageTotal property returns the total number of pages in the report. This property is set during the call to [EndDoc](#), otherwise it is set to one. To get an accurate page total, print report to preview screen and then call PageTotal.

## ClearTabs Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** ClearTabs;

### **Description**

The ClearTabs method clears all previously set tab locations. If no tabs have been set, this method does nothing.

ClearTabs is called by the [StartDoc](#) method. Calls to [SetTab](#), therefore, should be called after StartDoc is called.



## DrawGraphic Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** DrawGraphic(x1,y1:single;AGraphic:TGraphic);

**Description**  
The DrawGraphic method places a bitmap, metafile or icon at the location specified by x1 and y1. The left position is set by x1 and the top position is set by y1, relative to the page margins.

**Example**  
The following example places a bitmap in the top-left corner of the report page.

```
Procedure Report1.Header;  
var ABitMap:TBitMap;  
begin  
    ...  
    ABitMap.LoadFromFile(c:\company\logo.bmp);  
    ReportWorks1.DrawGraphic(0,0,ABitMap);  
    ...  
end;
```

## DrawLine Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** DrawLine;

### Description

The DrawLine method draws a line from the left margin to the right margin at the current line, or row, position. The line thickness is determined by the setting of the property [LineWidth](#).

### Example

The following example draws a double line, one thick and one fine.

```
Procedure Example.Report;
begin
  with ReportWorks1 do
    begin
      StartDoc;
      ...
      LineWidth := lwThick;
      DrawLine;
      NextLine;
      LineWidth := lwFine;
      DrawLine;
      ...
      EndDoc;
    end;
end;
```

## Ellipse Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** Ellipse(x1,y1,x2,y2:single);

**Description**  
The Ellipse method draws an ellipse, bound by the four corners determined by x1, y1, x2 and y2, on the report page. The corner parameters are as follows:

|           |               |
|-----------|---------------|
| <b>x1</b> | <b>Left</b>   |
| <b>x2</b> | <b>Right</b>  |
| <b>y1</b> | <b>Top</b>    |
| <b>y2</b> | <b>Bottom</b> |

Corner parameters are determined within the page margins, not within the actual page dimensions.

**Example**  
The following example draws an 3 x 4 inch ellipse starting one inch down from the top margin and a half inch in from the left margin.

```
Procedure Example.DrawEllipse;  
begin  
  with ReportWorks1 do  
    begin  
      TopMargin := 1;  
      LeftMargin := 1;  
      BottomMargin := 1;  
      RightMargin := 1;  
      Ellipse(.5, 1, 3.5, 5);  
    end;  
end;
```

## EndDoc Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** EndDoc;

### Description

The EndDoc method signals the end of the report. The PageTotal property is set in this procedure. If [Destination](#) is set to dPreview, then Refresh is called to update the preview screen, otherwise a call is made to Printer.EndDoc to set output to the printer or file. The Abort / Status window, if open, is closed.

### Example

The following example ends a report.

```
Procedure Example.Report;  
begin  
  with ReportWorks1 do  
    begin  
      StartDoc;  
      ...  
      EndDoc;  
    end;  
end;
```

## FirstPage Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** FirstPage;

### **Description**

The FirstPage method sets the CurrentPage property to 1. The next time the users report procedure is called, the first page of the report will appear on the preview screen if Destination is set to **dPreview**.

## FontBold Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** FontBold;

**Description**  
The FontBold method sets the current report font style to **Bold**. Other styles ( ie: Italic) are not affected.

## FontItalic Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** FontItalic;

**Description**  
The FontItalic method sets the current report font style to *Italic*. Other styles ( ie: Bold) are not affected.

## FontRegular Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** FontRegular;

**Description**  
The FontRegular method cancels all previously set styles for the current report font.



## FontStrikeout Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** FontStrikeout;

**Description**  
The FontStrikeout method sets the current report font style to ~~Strikeout~~. Other styles ( ie: Italic) are not affected.

## FontUnderline Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** FontUnderline;

**Description**  
The FontUnderline method sets the current report font style to UnderLine. Other styles ( ie: Italic) are not affected.

## GoToPageBottom Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** GoToPageBottom;

**Description**  
The GoToPageBottom moves the current report page position for drawing or text output to the page bottom margin.

**Example**  
The following example draws a line at the bottom margin position.

```
Procedure Example.Report;  
begin  
  with ReportWorks1 do  
    begin  
      StartDoc;  
      ...  
      GoToPageBottom;  
      DrawLine;  
      EndDoc;  
    end;  
end;
```

## GoToPageTop Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** GoToPageTop;

**Description**  
The GoToPageTop moves the current report page position for drawing or text output to the page top margin.

**Example**  
The following example draws a line at the top margin position.

```
Procedure Example.Report;  
begin  
  with ReportWorks1 do  
    begin  
      StartDoc;  
      ...  
      GoToPageTop;  
      DrawLine;  
      ...  
      EndDoc;  
    end;  
end;
```

## GoToXY Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** GoToXY(x,y:single);

**Description**  
The GoToXY method moves the current report page position for drawing or text output to the position determined by x and y. Position is relative to the page margins.

**Example**  
The following example draws a 3 inch line one inch down from the top margin and a half inch in from the left margin.

```
Procedure Example.Report;  
begin  
  with ReportWorks1 do  
    begin  
      StartDoc;  
      ...  
      GoToXY(.5, 1);  
      LineTo(3.5, 1);  
      ...  
      EndDoc;  
    end;  
end;
```

## IsPageEnd Method

**Applies to**  
ReportWorks

**Declaration**  
**function** IsPageEnd : Boolean;

**Description**  
The IsPageEnd function returns 1 if the current report page position is at the bottom margin, otherwise, it returns 0.

**Example**  
The following example prints a column of numbers on more than one page.

```
Procedure Example.Report;  
var x : integer  
begin  
  with ReportWorks1 do  
    begin  
      StartDoc;  
      ...  
      for x := 1 to 200 do  
        begin  
          Print(IntToStr(x));  
          NextLine;  
          if IsPageEnd then NewPage;  
        end;  
        ...  
      EndDoc;  
    end;  
end;
```

## LastPage Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** LastPage;

### Description

The LastPage method sets the CurrentPage property to the value of the PageTotal property. The next time the users report procedure is called, the last page of the report will appear on the preview screen if Destination is set to **dPreview**.

## LineCenter Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** LineCenter;

**Description**  
The LineCenter method moves the report page pen position to the center of the current line relative to the page margins.



## LineEnd Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** LineEnd;

**Description**  
The LineEnd method moves the report page pen position to the end of the current line relative to the page margins, ie: right margin.

## LineStart Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** LineStart;

**Description**  
The LineStart method moves the report page pen position to the beginning of the current line relative to the page margins, ie: left margin.

## LineTo Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** LineTo(x,y:single);

**Description**  
The LineTo method draws a line from the current pen position to the position specified by x and y relative to the page margins.

**Example**  
The following example draws a 3 inch line one inch down from the top margin and a half inch in from the left margin.

```
Procedure Example.Report;  
begin  
  with ReportWorks1 do  
    begin  
      StartDoc;  
      ...  
      GoToXY(.5, 1);  
      LineTo(3.5, 1);  
      ...  
      EndDoc;  
    end;  
end;
```

## NewPage Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** NewPage;

### Description

The NewPage method increments [PageNum](#), sets the report page pen position to the top of the next page, updates the Abort / Status window if open, and calls Printer.NewPage.

### Example

The following example prints a column of numbers on more than one page.

```
Procedure Example.Report;
var x : integer
begin
  with ReportWorks1 do
    begin
      StartDoc;
      ...
      for x := 1 to 200 do
        begin
          Print(IntToStr(x));
          NextLine;
          if IsPageEnd then NewPage;
        end;
        ...
      EndDoc;
    end;
end;
```

## NextLine Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** NextLine;

**Description**  
The NextLine method sets the report page pen position to the next line or row as determined by the setting of the [LineMeasure](#) property.

**Example**  
The following example prints a column of numbers on more than one page.

```
Procedure Example.Report;  
var x : integer  
begin  
  with ReportWorks1 do  
    begin  
      StartDoc;  
      ...  
      for x := 1 to 200 do  
        begin  
          Print(IntToStr(x));  
          NextLine;  
          if IsPageEnd then NewPage;  
        end;  
        ...  
      EndDoc;  
    end;  
end;
```

## NextPage Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** NextPage;

### **Description**

The NextPage method increments the [CurrentPage](#) property. Bounds are checked to prevent CurrentPage exceeding the PageTotal. The next time the users report procedure is called, the next page of the report will appear on the preview screen if [Destination](#) is set to **dPreview**.

## Paint Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** Paint;

### **Description**

The Paint method repaints the preview screen. Under most circumstances this method does not need to be call. Instead, call **Refresh** to update the preview screen, if needed. Normally, ReportWorks handles all necessary repainting.

## PositionGraphic Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** PositionGraphic(AGraphic:TGraphic;AJustify:TJustify);

**Description**  
The PositionGraphic method draws a bitmap, icon or metafile justified left, right or center on the current report page line or row.

**Example**  
The following example draws the company logo in the top left corner of the report page.

```
Procedure Example.Report;  
var ABitMap:TBitMap;  
begin  
    with ReportWorks1 do  
        begin  
            StartDoc;  
            ...  
            GoToPageTop  
            ABitMap.LoadFromFile(c:\company\logo.bmp);  
            PositionGraphic(ABitMap,jLeft);  
            ...  
            EndDoc;  
        end;  
end;
```



## PreviousLine Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** PreviousLine;

### **Description**

The PreviousLine method sets the report page pen position to the previous line or row as determined by the setting of the [LineMeasure](#) property. If LineMeasure is set to **ImTextHeight**, it is assumed that the report page font size has not been changed between the last call to [NextLine](#) and the call to PreviousLine.

## PreviousPage Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** PreviousPage;

### **Description**

The PreviousPage method decrements the [CurrentPage](#) property. Bounds are checked to prevent CurrentPage being set below page 1. The next time the users report procedure is called, the previous page of the report will appear on the preview screen if [Destination](#) is set to **dPreview**.

## Print Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** Print(s:string);

### Description

The Print method prints a text string at the current report page pen position. Word wrapping is not performed, so the user is responsible to make sure the string fits on the line properly within the page margins or tab positions.

If printing on a shaded line, the [TextStyle](#) property should be set to **tsTransparent** to enable proper text appearance.

### Example

The following example prints a column of numbers on more than one page.

```
Procedure Example.Report;
var x : integer
begin
  with ReportWorks1 do
    begin
      StartDoc;
      ...
      for x := 1 to 200 do
        begin
          Print(IntToStr(x));
          NextLine;
          if IsPageEnd then NewPage;
        end;
        ...
      EndDoc;
    end;
end;
```

## PrintCenter Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** PrintCenter(s:string);

**Description**  
The PrintCenter method prints a text string centered on the current report page line or row. Word wrapping is not performed, so the user is responsible to make sure the string fits on the line properly within the page margins or tab positions.

**Example**  
The following example prints the report title centered near the top of the page.

```
Procedure Example.Report;  
var x : integer  
begin  
  with ReportWorks1 do  
    begin  
      StartDoc;  
      DrawLine  
      NextLine;  
      SetFont(Arial,24);  
      PrintCenter(Title);  
      NextLine;  
      SetFont(Arial,12);  
      ...  
      EndDoc;  
    end;  
end;
```

## PrintLeft Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** PrintLeft(s:string);

### **Description**

The PrintLeft method prints a text string left justified on the current report page line or row. Word wrapping is not performed, so the user is responsible to make sure the string fits on the line properly within the page margins or tab positions.

### **Example**

The following example prints a column of numbers left justified on more than one page.

```
Procedure Example.Report;  
var x : integer  
begin  
  with ReportWorks1 do  
    begin  
      StartDoc;  
      ...  
      for x := 1 to 200 do  
        begin  
          PrintLeft(IntToStr(x));  
          NextLine;  
          if IsPageEnd then NewPage;  
        end;  
        ...  
      EndDoc;  
    end;  
end;
```

## PrintLine Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** PrintLine(s:string);

### Description

The PrintLine method prints a text string at the current report page pen position and calls the [NextLine](#) method. Word wrapping is not performed, so the user is responsible to make sure the string fits on the line properly within the page margins or tab positions.

If printing on a shaded line, the [TextStyle](#) property should be set to **tsTransparent** to enable proper text appearance.

### Example

The following example prints a column of numbers, using **PrintLine**, on more than one page.

```
Procedure Example.Report;
var x : integer
begin
  with ReportWorks1 do
    begin
      StartDoc;
      ...
      for x := 1 to 200 do
        begin
          PrintLine(IntToStr(x));
          if IsPageEnd then NewPage;
        end;
        ...
      EndDoc;
    end;
end;
```

## PrintParagraph Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** PrintParagraph(pgraph:pchar);

**Description**  
The PrintParagraph method prints a null-terminated text string starting at the current report page line. Word wrapping is performed within the report page margins. If the bottom margin is reached during the printing of the string, [NewPage](#) is called and printing continues on the next page.

**Example**  
The following example prints a char array using PrintParagraph.

```
Procedure Example.Report;  
var Abuffer : array[1..80] as char;  
begin  
  Abuffer := Thisis a test.;  
  with ReportWorks1 do  
    begin  
      StartDoc;  
      ...  
      PrintParagraph(ABuffer);  
      ...  
      EndDoc;  
    end;  
end;
```

## PrintRight Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** PrintRight(s:string);

### Description

The PrintRight method prints a text string right justified on the current report page line or row. Word wrapping is not performed, so the user is responsible to make sure the string fits on the line properly within the page margins or tab positions.

If printing on a shaded line, the [TextStyle](#) property should be set to **tsTransparent** to enable proper text appearance.

### Example

The following example prints a column of numbers right justified on more than one page.

```
Procedure Example.Report;
var x : integer
begin
  with ReportWorks1 do
    begin
      StartDoc;
      ...
      for x := 1 to 200 do
        begin
          PrintRight(IntToStr(x));
          NextLine;
          if IsPageEnd then NewPage;
        end;
        ...
      EndDoc;
    end;
end;
```



## PrintTab Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** PrintTab(tTab:integer; s:string);

### Description

The PrintTab method prints a string at the tab position indicated by **tTab**. Use [SetTab](#) to establish tab positions before using the PrintTab method. The string is printed left, right or center justified depending on how the tab position was set. Word wrapping is not performed, so the user is responsible to make sure the string fits on the line properly within the page margins or tab positions.

If printing on a shaded line, the [TextStyle](#) property should be set to **tsTransparent** to enable proper text appearance.

### Example

The following example prints a column of right justified numbers using **PrintTab**. Only one tab position is set.

```
Procedure Example.Report;
var x : integer
begin
  with ReportWorks1 do
    begin
      StartDoc;
      ...
      SetTab(6,jRight);
      for x := 1 to 20 do
        begin
          PrintTab(1,IntToStr(x));
          NextLine;
          if IsPageEnd then NewPage;
        end;
      ...
      EndDoc;
    end;
end;
```

## Rectangle Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** Rectangle(x1,y1,x2,y2:single);

**Description**  
The Rectangle method draws a rectangle, bound by the four corners determined by x1, y1, x2 and y2, on the report page. The corner parameters are as follows:

|           |               |
|-----------|---------------|
| <b>x1</b> | <b>Left</b>   |
| <b>x2</b> | <b>Right</b>  |
| <b>y1</b> | <b>Top</b>    |
| <b>y2</b> | <b>Bottom</b> |

Corner parameters are determined within the page margins, not within the actual page dimensions.

**Example**  
The following example draws an 3 x 4 inch rectangle starting one inch down from the top margin and a half inch in from the left margin.

```
Procedure Example.DrawRectangle;  
begin  
  with ReportWorks1 do  
    begin  
      TopMargin := 1;  
      LeftMargin := 1;  
      BottomMargin := 1;  
      RightMargin := 1;  
      Rectangle(.5, 1, 3.5, 5);  
    end;  
end;
```

## ResetPagePosition Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** ResetPagePosition;

**Description**  
The ResetPagePosition method returns the report page pen to the position saved by [SetPagePosition](#).

**Example**  
The following example draws a line, then moves to the bottom margin, draws another line and returns to the previous position.

```
Procedure Example.Report;  
begin  
  with ReportWorks1 do  
    begin  
      StartDoc;  
      ...  
      SetPagePosition;  
      DrawLine  
      GoToPageBottom;  
      DrawLine;  
      ResetPagePosition;  
      ...  
    end;  
end;
```

## SetBrush Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** SetBrush(bColor:TColor;bStyle:TBrushStyle);

**Description**  
The SetBrush method sets the report page brush to the color specified by **bColor** using the style specified by **bStyle**. See Delphi help for description of **TColor** and **TBrushStyle** types.

## SetBrushDefault Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** SetBrushDefault;

**Description**  
The SetBrushDefault method sets the report page brush to default values. Brush color is set to **black** and brush style is set to **solid**.

## SetFont Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** SetFont(fname:string;fsize:integer);

**Description**  
The SetFont method sets the report page font to the font name specified by **fname** and to the size specified by **fsize**.

**Example**  
The following example sets the font to Arial in two different sizes.

```
Procedure Example.Report;  
begin  
  with ReportWorks1 do  
    begin  
      StartDoc;  
      DrawLine  
      NextLine;  
      SetFont(Arial,24);  
      PrintCenter(Title);  
      NextLine;  
      SetFont(Arial,12);  
      ...  
      EndDoc;  
    end;  
end;
```

## SetPagePosition Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** SetPagePosition;

**Description**  
The SetPagePosition method captures the current report page pen position. This position can be reset later by a call to [ResetPagePosition](#).

## SetPen Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** SetPen(pColor:Tcolor; pWidth:integer;  
pStyle:TPenStyle; pMode:TPenMode);

**Description**  
The SetPen method set the report page pen to the color specified by **pColor**, the width specified by **pWidth**, the style specified by **pStyle** and the mode specified by **pMode**. See **TColor**, **TPenStyle** and **TPenMode** for information on these types.



## SetPenDefault Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** SetPenDefault;

**Description**  
The SetPenDefault method sets the report page pen to the following default values. The pen color is set to black, width is set to 1, style is set to solid, and the mode is set to Copy.

## SetTab Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** SetTab(tpos:single;tjus:TJustify) : integer;

**Description**  
The SetTab method creates a tab list relative to the specified tab position in **tpos**. Tab justification is specified in **tjus**. Tab positions are relative to the report page margins.

**Example**  
The following example prints a column of right justified numbers using **SetTab**. Only one tab position is set.

```
Procedure Example.Report;  
var x : integer  
begin  
  with ReportWorks1 do  
    begin  
      StartDoc;  
      ...  
      SetTab(6,jRight);  
      for x := 1 to 20 do  
        begin  
          PrintTab(1,IntToStr(x));  
          NextLine;  
          if IsPageEnd then NewPage;  
        end;  
        ...  
      EndDoc;  
    end;  
end;
```

## ShadeLine Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** ShadeLine(AColor: TColor);

**Description**  
The ShadeLine method colors the current report page line, or row, to the color specified by AColor;

**Example**  
The following example shades alternate lines to the color yellow.

```
Procedure Example.Report;  
var x : integer;  
    shade : boolean;  
begin  
    shade := 0;  
    with ReportWorks1 do  
        begin  
            StartDoc;  
            ...  
            for x := 1 to 100 do  
                begin  
                    shade := not shade;  
                    if shade then ShadeLine(clYellow);  
                    print(IntToStr(x));  
                    NextLine;  
                    if IsPageEnd then NewPage;  
                end;  
            ...  
            EndDoc;  
        end;  
    end;  
end;
```

## StartDoc Method

**Applies to**  
ReportWorks

**Declaration**  
**procedure** StartDoc;

### Description

The StartDoc method performs a number of initializations that are required before the report can be printed to the preview screen, printer or file. First, if the [OnPrint](#) event is assigned then the assigned procedure is run. Then tabs are cleared with [ClearTabs](#), margins are set, the report page pen is set to the top of the page, the default pen and brush are set and [PageNum](#) is initialized.

If [Destination](#) is set to dPreview then the preview page is drawn. If Destination is set to dFile then the proper initializations for printing to a file are performed. The Abort / Status window is created and activated. Finally, the Printer.BeginDoc method is called.

As a number of initializations are performed by [StartDoc](#), this method should be the first method called in a report. Properties can be set before calling StartDoc.

### Example

The following example shows StartDoc being call first.

```
Procedure Example.Report;  
begin  
    with ReportWorks1 do  
        begin  
            StartDoc;  
            DrawLine  
            ...  
            EndDoc;  
        end;  
end;
```

## StretchGraphic Method

**Applies to**  
ReportWorks

### **Declaration**

**procedure** StretchGraphic(X1,Y1,X2,Y2:single;AGraphic:TGraphic);

### **Description**

The StretchGraphic method draws a bitmap, icon or metafile to the area specified by X1, Y1, X2 and Y2. The corner parameters are as follows:

|           |               |
|-----------|---------------|
| <b>X1</b> | <b>Left</b>   |
| <b>X2</b> | <b>Right</b>  |
| <b>Y1</b> | <b>Top</b>    |
| <b>Y2</b> | <b>Bottom</b> |

The graphic to draw is passed in **AGraphic**.

## OnPrint Event

**Applies to**  
ReportWorks

**Declaration**  
**property** OnPrint;

### **Description**

The OnPrint event is called by StartDoc before any initializations are performed and the call to Printer.BeginDoc is made. Use the OnPrint event to perform special initializations or open windows before the report starts to print.

## Abort Form

The Abort form is called by [StartDoc](#) when the report is printing to the printer or to a file. The current printing page, report name and printer name is displayed.

Printing can be aborted by pressing the **Abort** button.





