

# QuickReport

version 1.0b for Delphi

Copyright 1996

qsd

Quick Soft Development AS

All Rights Reserved

web <http://www.qsd.no>

compuserve 101372,3471

swreg #8644

Postal address:

Rådhusgt. 9a  
N-0151 Oslo  
Norway

Fax number: +47 22 41 74 91



## Introduction

7

About this document

7

About QuickReport

7

Licensing and Registration

7

32 bit QuickReport 1.0b for Delphi 2.0

7

32 bit QuickReport Source code and QuickReport for Win 3.1

7

Copyright and Distribution

7

Getting Technical Support

8

## Using QuickReport

8

Getting Started

8

Installation of QuickReport with Delphi 1.0

8

Installing the Report Templates in Delphi 1.0

8

QuickReport Templates and the Delphi 2.0 Object Repository

9

QuickReport Sample Applications

9

QuickReport Basics

10

Creating a very simple report

10

Customizing Report Display

11

Creating a Master Detail Report

12

Printing single record reports

13

Creating reports from non BDE data sources

13

Creating a Custom Preview Form

13

Printing Images

13

Important Issues when using QuickReport

14

Using Small and Large fonts when designing reports

14

Limitations of the design time preview

14

QuickReport and DataSources

14

Answers to some Frequently Asked Questions

14

How do I format output of a data field?

14

How do I combine several reports into one?

14

How do I filter records?

15

How do I make sure several bands stay on the same page?

15

Why are my fonts clipped or displayed at a wrong size during preview?

15

Why are my reports suddenly rescaled, all the fonts have changed size?

15

How do I change the layout of mailing labels to fit any particular form ?

15

How do I eliminate the large black areas on the HP LaserJet 4 reports?

15

Why are some bands printed at the top of the report, even if they shouldn't?

15

How do I print one record per page?

15

Why doesn't my modeless custom preview form work?

15

#### QuickReport Reference

17

#### QRBand

17

property Align : TAlign

17

property BandType : TQRBandType

17

property Color : TColor

18

property Enabled : Boolean

18

property Font : TFont

18

property ForceNewPage : Boolean

18

property Frame : TPen

18

property LinkBand : TQRBand

18

property ParentFont : Boolean

18

property Ruler : TQRruler

19

event AfterPrint : procedure (BandPrinted : Boolean)

19

event BeforePrint : TQRBandBeforePrintEvent

19

#### QRCustomControl

19

property Caption

19

method Paint  
19  
method Print(x,y : integer)  
19  
method Stretch(var Value : integer)  
19  
QRDBCalc  
20  
property AlignToband : Boolean  
20  
property AsInteger : Longint  
20  
property AsReal : real  
20  
property Font : TFont  
20  
property Operation : TQRCalcOperation  
20  
property ParentFont : TFont  
20  
property PrintMask : String  
20  
property ResetBand : TQRBand  
20  
event OnPrint : procedure (sender : TObject; var Value : String)  
21  
QRDetailLink  
21  
property DataSource  
21  
property DetailBand  
21  
property FooterBand  
21  
property HeaderBand  
21  
property Master : TQRController  
21  
property PrintBefore : Boolean  
21  
event OnFilter : procedure (PrintRecord : Boolean) of object  
22  
event OnNeedData : procedure (MoreData : Boolean) of object  
22  
QRDBText  
22  
property Alignment : TAlignment  
22  
property AlignToband : Boolean  
22  
property AutoSize : Boolean  
22  
property Color : TColor  
22  
property DataSource : TDataSource  
22

property DataField : String  
23  
property Font : TFont  
23  
property ParentFont : Boolean  
23  
property Transparent : Boolean  
23  
event OnPrint : procedure (sender : TObject; var Value : String)  
23

#### QRGroup

23

property DataSource : TDataSource  
23  
property DataField : String  
23  
property FooterBand : TQRBand  
23  
property HeaderBand : TQRBand  
24  
property Level : Integer  
24  
Event OnNeedData : procedure (var Value : String)  
24

#### QRLabel

24

property Alignment : TAlignment  
24  
property AlignToband : Boolean  
24  
property AutoSize : Boolean  
24  
property Caption : String  
24  
property Color : TColor  
24  
property Font : TFont  
25  
property ParentFont : Boolean  
25  
property Transparent : Boolean  
25  
event OnPrint : procedure (sender : TObject; var Value : String)  
25

#### QRMemo

25

property Alignment : TAlignment  
25  
property Color : TColor  
25  
property Font : TFont  
25  
property Lines : TStringList  
25  
property ParentFont : Boolean  
26

property WordWrap : Boolean  
26

QRPreview  
26

property PageNumber : integer  
26

property Zoom : integer  
26

method ZoomToFit  
26

method ZoomToWidth  
26

QRPrinter  
26

property Canceled : boolean  
26

property Canvas : TCanvas (Delphi 1.0) TMetafileCanvas (Delphi 2.0)  
27

property Copies : integer  
27

property EnableOpenBtn : boolean  
27

property EnablePrintBtn : boolean  
27

property EnableSaveBtn : boolean  
27

property FromPage : integer  
27

property LeftWaste : integer  
27

property Orientation : TPrinterOrientation  
27

property Page : TMetafile  
27

property PageCount : integer  
27

property PageHeight : integer  
27

property PageNumber : integer  
28

property PageWidth : integer  
28

property PaperBin : integer  
28

property PaperLength  
28

property PaperSize : TQRPaperSize  
28

property PaperWidth  
28

property PreviewCaption : String  
29

property PrinterOK : boolean  
29

property ProgressCaption : String  
29

property ShowProgress : boolean  
29  
property Status : TQRPrinterStatus  
29  
property Thumbs : integer  
29  
property Title : String  
29  
property ToPage : integer  
29  
property TopWaste : integer  
29  
property TotalPageHeight : integer  
30  
property TotalPageWidth : integer  
30  
property ZoomButtons : boolean  
30  
event OnPreview  
30  
method BeginDoc  
30  
method Cancel  
30  
method CleanUp  
30  
method EndDoc  
30  
method Load(Filename : String)  
30  
method NewPage  
30  
method Preview  
31  
method Print  
31  
method Save(Filename : String)  
31  
QRSysData  
31  
property Alignment : TAlignment  
31  
property AlignToband : Boolean  
31  
property AutoSize : Boolean  
31  
property Color : TColor  
31  
property Data : TQRSysDataType  
32  
property Font : TFont  
32  
property ParentFont : Boolean  
32  
event OnPrint : procedure (sender : TObject; var Value : String)  
32

property Text : String

32

property Transparent : Boolean

32

QRShape

32

property Brush : TBrush

32

property Height : Integer

33

property Pen : TPen

33

Property Shape : TQRShapeType

33

property width

33

QuickReport

33

property PageCount : Longint

33

property PageNumber : Longint

33

property RecordNo : longint

34

property ColumnMarginInches : Longint

34

property ColumnMarginMM : longint

34

property Columns : integer

34

property DisplayPrintDialog : Boolean

34

property LeftMarginInches : Longint

34

property LeftMarginMM : Longint

34

property OnStartPage : TNotifyEvent

34

property Orientation : TPrinterOrientation

34

property PageFrame : TQRFrame

34

property PaperLength

35

property PaperSize : TQRPaperSize

35

property PaperWidth

35

property ReportTitle : String

35

property RestartData : Boolean

35

property ShowProgress : Boolean

35

property SQLCompatible : Boolean

35



property TitleBeforeHeader : Boolean  
36  
method NewPage  
36  
method Prepare  
36  
method Preview  
36  
method Print  
36  
event OnEndPage : TNotifyEvent  
36  
event AfterDetail : TNotifyEvent  
36  
event AfterPrint : TNotifyEvent  
36  
event BeforeDetail : TNotifyEvent  
36  
event BeforePrint : procedure (var PrintReport : Boolean)  
36  
event OnFilter : procedure (PrintRecord : Boolean) of object  
37  
event OnNeedData : procedure (MoreData : Boolean) of object  
37

#### QuickReport Registration Form

38

#### QuickReport Mailing List

39

Files you can download from our internet server

39

3<sup>rd</sup> party QuickReport printable components

39

Version history

39

Version 1.0b

39

Version 1.0a

39

Version 1.0

40

Beta versions

40

Credits

40

## **Introduction**

### ***About this document***

This document describes both the 16 bit and 32 bit version of QuickReport for Delphi. You use QuickReport and the features described in this document entirely at your own risk. If you have any questions regarding QuickReport, this document or related topics please contact QSD.

### ***About QuickReport***

Welcome to QuickReport for Delphi. QuickReport is designed to be a tightly integrated visual reporting tool, that also gives you great code control over your reports. Here is a listing of some of QuickReport's features:

- Banded report generator, using the Delphi form designer as the report designing environment
- Event handlers can be attached to virtually any part of the report generation process to give you great control of the finished report
- Powerful on-screen preview, including preview component for easy creation of your own preview forms. Reports can even be previewed from within the Delphi IDE
- Create reports from tables and queries, in addition to arrays, lists, text files or any source you can imagine
- Powerful master/detail functions with unlimited levels and unlimited detail tables at the same level
- Create mailing labels and multi column reports
- Automatic printing of thumbnails
- Print labels, database fields, calculations, memo fields, pictures, icons and shapes
- Use any Windows font, size, style and color
- Finished reports can be saved to file for later viewing or printing
- Portrait or Landscape printing of reports
- Build in support for custom paper sizes
- On line help file describing all methods, properties and events

### ***Licensing and Registration***

#### **32 bit QuickReport 1.0b for Delphi 2.0**

Delphi 2.0 includes the full 32 bit version of QuickReport 1.0. No additional registration is needed to use the version included with Delphi 2.0. The full 32 bit version of QuickReport 1.0b is available for free use by Delphi 2.0 users.

#### **32 bit QuickReport Source code and QuickReport for Win 3.1**

The full source code for QuickReport is available directly from QSD AS. For a single purchase price of \$79.00 US you get the full source code *and* the Win 3.1 version of QuickReport. Full order information is available on the registration sheet at the back of this document.

### ***Copyright and Distribution***

QuickReport is copyright 1996 by QSD AS (QuickSoft Development AS). QSD can be contacted on the web at <http://www.qsd.no>, via fax at +47 22 41 64 91 or postal mail at:  
QSD AS, Rådhusgt. 9, N-0151 Oslo, Norway

The latest version of QuickReport can be downloaded from our ftp server at [ftp.qsd.no](ftp://ftp.qsd.no). Go to the /pub/qsd directory. A pointer to the latest version can also be found on our web page.

Use of QuickReport is entirely at your own risk. Under no events shall QSD AS be responsible for any loss or damage caused by the use or distribution of QuickReport.

### ***Getting Technical Support***

Users of the 32 bit version of QuickReport included in Delphi 2.0 will get QuickReport support through Borland Technical Support .

Users who have purchased the 16 bit version / source code from QSD will receive technical support via e-mail. Please send your questions to *support@qsd.no*. Depending on the amount of registrations we might also set up a technical support phone line but details are not set yet. Announcements will be made through our Web site.

## Using QuickReport

### **Getting Started**

QuickReport is a fast, easy, native component reporting solution for Delphi. To help you understand how to use QuickReport, Delphi includes both templates in the Object Repository and a full sample reporting application in Delphi 2.0/Demos/Quickrpt directory.

### Installation of QuickReport with Delphi 1.0

Please follow these steps to install QuickReport in Delphi:

- Create a directory called QUICKRPT under your Delphi directory (C:\DELPHI\QUICKRPT)
- Unzip the QuickReport zip file in this directory
- Start Delphi
- Choose *Options | Install Components*
- Click *Add*
- Type your QuickReport directory path followed by QUICKREP.DCU (C:\DELPHI\QUICKRPT\QUICKREP.DCU)
- Click *OK*
- Click *OK* again

Delphi will now rebuild the library and add a QuickReport page on your component palette.

If you experience any problems please look up “Adding and Removing Components from the Library” in the Delphi online help.

### Installing the Report Templates in Delphi 1.0

There are three report templates included in QuickReport

<i>QRLabels</i>	A basic mailing label template
<i>QRList</i>	Simple list template with page header and footer
<i>QRMD</i>	A Master/Detail template, using the QRDetailLink component

To install the report template intended to speed up the creation of reports please follow these instructions.

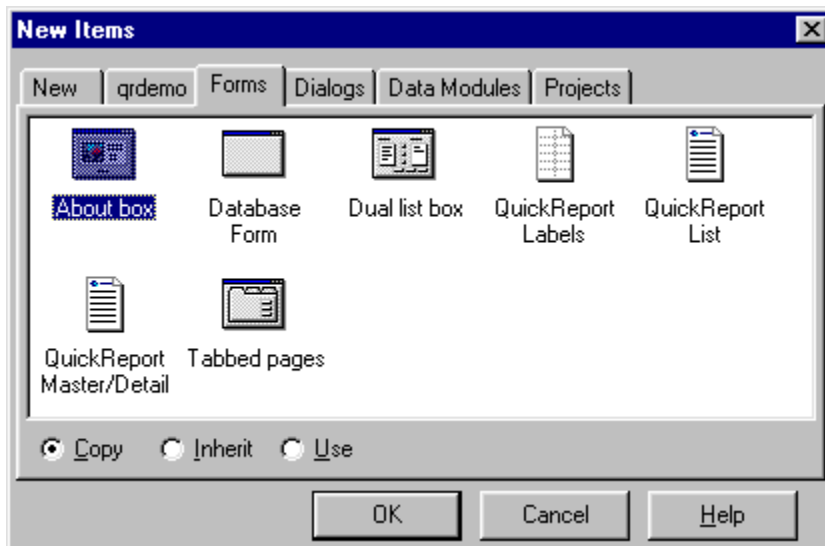
- \* Open the file QRLABELS.PAS in the form designer
- \* Choose Options | Gallery
- \* Choose Form Templates tab
- \* Type “QuickReport Labels”
- \* Click Browse
- \* Locate the QRLABELS.BMP file and click OK
- \* Click OK
- \* Repeat the steps for QRLIST.PAS with text “QuickReport List” and QRMD with text “QuickReport MasterDetail”

In your form Gallery you should now have three QuickReport template, which can be used as starting point for new reports.

### QuickReport Templates and the Delphi 2.0 Object Repository

In Delphi 2.0 you will find three report templates on the Forms page of the Object Repository, including:

*QuickReport Mailing Labels*      A basic mailing label template  
*QuickReport List*                      Simple list template with page header and footer  
*QuickReport Master/detail* A Master/Detail template, using the QRDetailLink component



Instructions on customising the templates are included in the source code automatically created when you use one of the templates.

### ***QuickReport Sample Applications***

QuickReport includes a sample application to get you started with your own reports. To use the demo program:

- Install the Delphi sample tables (DBDEMOS alias)
- Move to the DELPHI 2.0\DEMOS\QUICKRPT directory for Delphi 2.0 and to your DELPHI\QUICKRPT directory for Delphi 1.0
- Open the project DEMO.DPR and run it

There are five reports in this program plus a sample custom preview form. Each of them demonstrates some basic QuickReport features. The main form itself demonstrates how to call reports, both for print and preview. It also demonstrates how to connect a custom preview for general use in your application.

***Simple Report*** Shows you how to create a basic report with a title band, column headers, detail band and a page footer with page number. In addition it shows a simple use of grouping with the QRGroup component and even the use of the QRGroup.OnNeedData event.

***Memos and Pictures*** This report shows how to include images and memo fields in your reports.

***Master Detail Report*** The QRDetailLink component is a powerful helper when you want to create master detail reports. This report includes a total of four tables and also demonstrates the how to avoid orphan bands.

***Mailing Labels*** This report demonstrates how to create a mailing label report, in addition to the use of an overlay band.

***Text File Lister*** One powerful feature of QuickReport is the ability to create reports from datasources other than TTable and TQuery. It is a text file lister, using the QuickReport.OnNeedData event to retrieve lines from a text file. The report can very well be used as a source code lister, since it displays two

columns of small text to fit a lot of information on one page.

**Custom Preview form** The custom preview form shows how you can incorporate the preview totally in your application's look and feel using the QRPreview component. This form is very simple but can act as a starting point for your own preview form.

### QuickReport Basics

When using QuickReport all reports are designed as Delphi forms using the Delphi form designer. Advanced users can also generate forms/reports programmatically on the fly but that is not covered in this document.

QuickReport is a *banded* report generator. This means that you build your reports using *QRBands*, a special type of the Delphi TPanel, as the different parts of your report. A report can have several bands of different types:

- The most important bands are the **Detail Band, Page header and footer band and Title band. Group Headers, Footers and SubDetail bands** work in conjunction with other components.

For a complete description of all band types please look up TQRBand in the Component Reference chapter.

### Creating a very simple report

The simplest report you can create is a form having the following components:

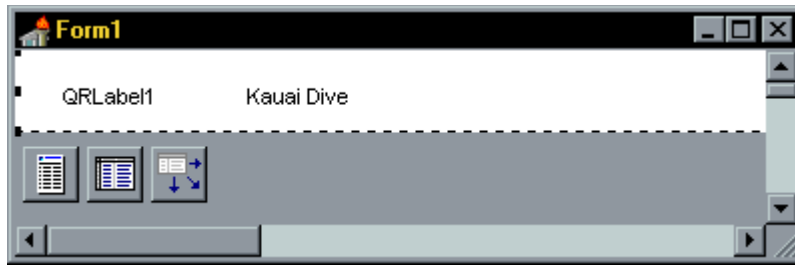
- A **TQRBand** with BandType set to rbTitle
- A **TQuickReport** component
- A **QRLabel** on the QRBand. Put some text in the Caption property
- Set the Form.Font to Arial or another TrueType font



Now you can right click the TQuickReport component and select *Preview* from the menu. A Preview form with a blank page, except for the tiny QRLabel you put on it, should now appear.

To make the report a bit more interesting, you do the following:

- Change the **QRBand.BandType property** to **rbDetail**
- Add a TTable component to the form. Set the DataBase property to DBDemos and the TableName to Customers. Set Active to True
- Add a TDataSource component and set the DataSet property to point to the TTable you just added
- Set the QuickReport.DataSource property to point to the newly added DataSource
- Add a QRDBText component to the QRBand. Set the DataSource property to the correct value and pick a field for the DataField property.

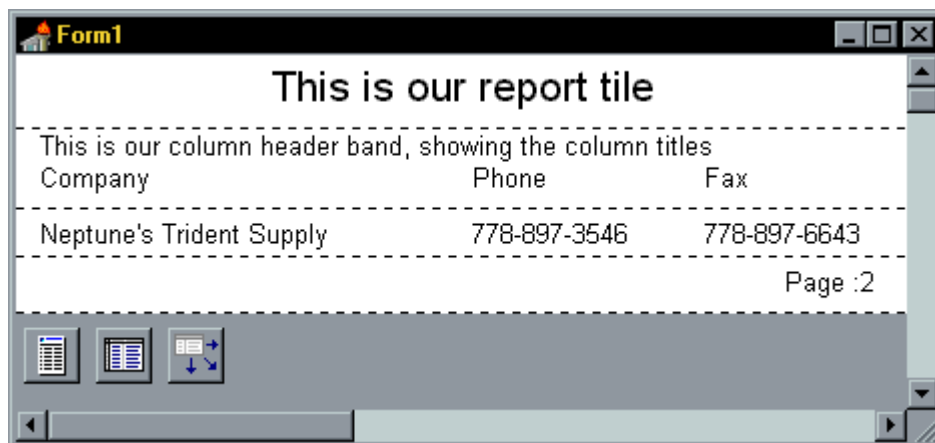


Now you can right click the QuickReport component and select *Preview* again. You have now finished your first data aware report using QuickReport!

## Customizing Report Display

We shall now create a customized report with a title band, column heading, detail band and page footer. Close any existing reports and create a new form.

- Create a new form
- Set the form font to Arial
- Put four bands on the report
- Set their respective band types to: rbTitle, rbColumnHeading, rbDetail and rbPageFooter
- Put a TTable component on the form, set DataBase to DBDemos, TableName to Customer and Active to True
- Put a TDataSource on the form, Set the DataSet property to Table1
- Put a TQuickReport component on the form. Set the DataSource property to DataSource1
- Put TQRLabels on the title and column header band as show below
- Put three QRDBText components on the Detail Band. Set their DataSource property to DataSource1 and the DataField properties to Company, Phone and Fax
- Put a QRSysData on the Page footer band. Set the Data property to qrcPageNumber. Set the Text property to "Page: "
- Your report should now look approximately like the one below



Again you can right click the QuickReport component and select *Preview*.

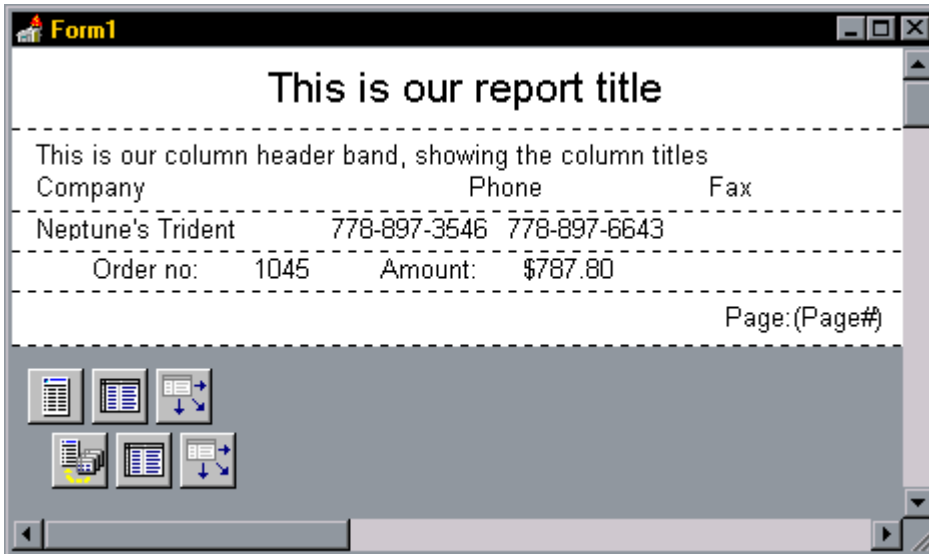
## Creating a Master Detail Report

We will now change this report into a master detail report, showing all the orders for each customer.

- Add a TTable, set the Database property to DBDemos and TableName to Orders, set Active to True. Set

the MasterSource to DataSource1 and click on the MasterField property to bring up the dialog box to create a master/detail relationship. Click on the index combo box and select CustNo, then click CustNo in both the list boxes and click *Add*. Click *OK*.

- Add a TDataSource, set the Dataset property to Table2 (the new table)
- Add a new QRBand to your report
- Add a QRDetailLink component. Set the DataSource property to DataSource2 (The new one). Set the Master property to QuickReport1 and DetailBand to QRBand5 (The new band). The BandType property of this band will now be set to rbSubDetail automatically.
- Put two QRDBText components on the new band, connect them both to DataSource2 and then to OrderNo and AmountPaid. Set the Alignment properties to alRighJustify
- Right click your QuickReport component and select *Preview*



The finished Master Detail form

### Printing single record reports

When your QuickReport component doesn't have a DataSource connected it will print only one set of detail bands, effectively creating a report listing only one record. This makes it very easy to print the current record using a Data Module in Delphi 2.0.

### Creating reports from non BDE data sources

QuickReport can create reports from virtually any type of data. By not specifying a DataSource, but instead creating a OnNeedData event handler you can make QuickReport print whatever data you manage to put into your QRLabel components. Both the QuickReport and the QRDetailLink components have the OnNeedData event, so you might very well create reports where some parts (master information) is taken from a table and other information (like detail information) is taken from some other source using the QRDetailLink.OnNeedData event. For a complete example on how to use the OnNeedData event take a look at the Text File Lister in the demonstration program. It will demonstrate how to print a text file with QuickReport, reading the file line by line and feeding the lines into a report.

### Creating a Custom Preview Form

A print preview is essential to any application. In addition to providing a full featured preview form automatically, QuickReport makes it very easy to create preview forms with your own look and feel. The QRPreview component takes all the hassle out of displaying the actual page, providing you with properties such as PageNumber and Zoom to give you total control.

There are basically two steps needed to create your own preview form. The first is to actually create the

form, using a QRPreview component and probably several buttons for navigating between the pages, zooming, printing or whatever you want include.

The second step is to hook this form into QuickReport. This is done by writing an event handler for the QRPrinter.OnPreview event. At first sight it might look strange that this isn't an event in QuickReport component, but that would require you connect your preview form to all reports in your form. Putting this event in the QRPrinter object will usually mean you have to write one line of code to attach the event handler, then it will be active for *all* reports in your application. A typical event handler for the QRPrinter.OnPreview event might look like this:

```
procedure TMyMainForm.OnPreview;
begin
    PreviewForm.ShowModal;
end;
```

And the code you would use to attach this preview form to the QRPrinter object could be put in the MyMainForm.OnCreate event and would look like this:

```
QRPrinter.OnPreview:=OnPreview;
```

Your preview form would now be used any time you call preview method.

For a complete example on both a preview form and how to connect it please take a look at the QuickReport demonstration program.

## Printing Images

QuickReport has no image component by itself. Instead it is aware of any standard TImage or TDBImage component placed on the report and will print them.

Note that to print a metafile image with the 16 bit version of QuickReport it needs to be a *placeable* metafile. Programs to convert a metafile to a placeable metafile can be found on our web server, <http://www.qsd.no>, and on CompuServe.

## **Important Issues when using QuickReport**

### Using Small and Large fonts when designing reports

Please always have your Windows font size set to small fonts when designing reports with QuickReport. This is to limit problems when your reports are printed on computers with other screen settings. Also remember to set your Form.Scaled property to False.

### Limitations of the design time preview

During the design time preview, QuickReport cannot call any event handlers you have created. This limits the design time preview to reports where the dataset is available at design time, and where events are not critical to the working of the report.

### QuickReport and DataSources

QuickReport will not restore the current record after it has finished creating a report. If you want your datasource to be restored you will have to do so manually.

Also note that a lot of screen flashing might appear if you run QuickReport on a dataset used by a form visible at the time you run the report. If you run reports on the same dataset as you use in your forms you should hide those forms before running the report. DisableControls will not work, since that will also disable the dataaware QuickReport controls, causing all records in the report to print the same data.

A new version of QuickReport which allows disable controls is available from...???

## **Answers to some Frequently Asked Questions**

How do I format output of a data field?



Formatting the output of a data field can be done in two ways. The best is to use the DisplayMask property of the field in question; double click your TTable or TQuery component and make sure the field is added as a component. Select the field component and set the DisplayMask property.

You can also use the OnPrint event of the QRDBText or QRLabel components to format the contents just prior to printing. Note that you should change the Value parameter passed to the event handler, not the actual caption of the component.

### **How do I combine several reports into one?**

Using the TQRDetailLink component you can create reports with many 'master' tables. Do not connect the TQuickReport component to a datasource, but instead create two or more TQRDetailLink components and set the TQuickReport component as the master for all of them. Connect detail bands, headers and footers as you wish. You can even attach any number of subDetail bands to these 'master' tables. When preparing the report, QuickReport will find out that no master table is present, and only the detail tables will be printed.

### **How do I filter records?**

QuickReport has a built in data filtering event which can be used to accept or reject a record. Both TQuickReport and TQRDetailLink has an OnFilter event, and in this event handler you can check the values of the current record. Set the PrintRecord parameter to False if you want to reject the current record.

### **How do I make sure several bands stay on the same page?**

The TQRBand component has a LinkBand property which is used to make several bands stick together. The LinkBand property can be set connected to another QRBand and QuickReport will then make a page break if there is not enough space left on the page for *both* bands. The second band can in turn be linked to a third band, and so on.

### **Why are my fonts clipped or displayed at a wrong size during preview?**

QuickReport relies on Windows to perform any scaling of fonts during preview and thumbnail printing. Windows is not able to scale all fonts, typically TrueType fonts and PostScript fonts work very well, while Bitmap fonts (like MS SYSTEM) have difficulty.. Please make sure you use a TrueType or PostScript font if you want accurately scaled previews and thumbnail printouts.

### **Why are my reports suddenly rescaled, all the fonts have changed size?**

Delphi sometimes rescale controls on a form, typically if you change screen resolution or Windows font size. Make sure the Scaled property of your report form is set to False, as this will minimize the problem. Also see the *Important Issues on using QuickReport* paragraph.

### **How do I change the layout of mailing labels to fit any particular form ?**

Select the number of labels across the page with the QuickReport.Columns property. Put a page header band on the report to change top spacing. Use the QuickReport.LeftMargin properties to change the left spacing and use the QuickReport.ColumnMargin to adjust column spacing. Using these three methods you should be able to create mailing label reports that fit most needs.

### **How do I eliminate the large black areas on the HP LaserJet 4 reports?**

Try selecting an HP LaserJet III driver instead.

### **Why are some bands printed at the top of the report, even if they shouldn't?**

The default BandType of is rbTitle. You have probably forgotten to change this to the correct band type.

### **How do I print one record per page?**

If you want to print a report with one record on each page you set the ForceNewPage property of your Detail band to True.

### **Why doesn't my modeless custom preview form work?**

When you call a QuickReport.Preview QuickReport first prepares the report, then displays it (modally) and

finally free the report from memory. But when you call up the preview modeless QuickReport continues to execute, freeing the report you are trying to view on screen at the same time. So a different approach must be used to show a report modeless. First issue a QuickReport.Prepare instead of Preview. Then call QRPrinter.Preview (make sure you have set the QRPrinter.OnPreview event to an event handler which calls up your preview form, modeless). The report will now stay in memory and your preview form will work correctly. Before you prepare another report you must now call a QRPrinter.Cleanup to free the allocated memory and make the QRPrinter ready to accept a new print job.

## QuickReport Reference



### QRBand



QRBands make up the different parts of your report, such as Detail Band for the data part, Page Header band for page headers and so on. QuickReport then replicates these bands everywhere they are supposed to appear.

Some band types are printed automatically by QuickReport, while others need to be connected to a QRGroup or QRDetailLink component to be printed. For the bands that print automatically you can have multiple bands of the same type. QuickReport will print them all (those with the Enabled property set to True) in their creation order. If you want to rearrange the order that they print, use the Send To Back and Bring To Front features of the Delphi form designer.

#### property Align : TAlign

The Align property of a QRBand is usually set to alTop, making the bands appear after one another in the form designer. If you want to visually rearrange the order of the bands you can do so by changing the properties to alNone, rearrange them, then set Align back to alTop. The order of the bands in the form designer has no impact on printing order.

#### property BandType : TQRBandType

The BandType property is used to tell QuickReport how to handle each band. There are several different band types, and picking the right one for each band is important. When you connect a band to a QRGroup or QRDetailLink component the BandType is automatically set to the right type. The possible values of BandType are:

**rbTitle** Report title band, printed once at the beginning of the report. You can have several bands of this type and they will all be printed one after another. Title bands are printed automatically

**rbPageHeader** Page Header, printed once at the top of each page. You can have several bands of this type and all will be printed after another on top of the page. Page header bands are printed automatically.

**rbColumnHeader** The Column Header band is much like the Page header, except that it is printed on top of every column in a multi column report. The Column Header bands are printed automatically

**rbDetail** Detail bands are replicated for all records in your dataset. You can have many detail bands but for basic reports you usually have only one. The detail bands are printed automatically

**rbPageFooter** The Page Footer band is printed at the bottom of each page. The page footers are printed automatically

**rbOverlay** Overlay bands are printed as an overlay of the page. It is always positioned from the top left corner, independent of all other bands on the page.

**rbGroupHeader** Group header bands are usually used as group headers in master/detail reports. They must be connected to a QRGroup component or a QRDetailLink component to be printed.

***rbSubDetail*** Sub Detail bands are used as detail bands for the detail tables in a master/detail report when you use the QRDetailLink component and must be connected to this component to be printed

***rbGroupFooter*** Group footers are used mostly as footers in master/detail reports. They must be connected to a QRGroup or QRDetailLink component to be printed.

***rbSummary*** Summary bands are automatically printed at the end of the report.

#### property Color : TColor

Sets the background color of the band

#### property Enabled : Boolean

If set to False, the band will not be printed. This property can be changed during report creation to print different bands depending of the data being printed.

#### property Font : TFont

The parent font to use for controls on this band.

#### property ForceNewPage : Boolean

If True, a page break will be made before this band is printed. This can be used if you want a group header to always appear on a new page or any other situations. Please see also the QuickReport.NewPage method.

#### property Frame : TPen

The pen to be used to frame the band. Set Pen width to 0 for no frame (default)

#### property LinkBand : TQRBand

The LinkBand property is used to make sure several bands always appear on the same page. When QuickReport checks if there is enough space left on the page to print the current band, it will include the size of a band connected to this property. This band can again be connected to another band, and so on. Please note that you should always link to the *next* band on the report, never backwards to the previous.

#### property ParentFont : Boolean

The ParentFont property determines where a band looks for its font information. If ParentFont is True, the Band uses the font in its parent form's Font property. If ParentFont is False, the Band uses its own Font property.

By using ParentFont, you can ensure that all the bands on a report have a uniform appearance. For example, if you want all the bands in a form to use 12-point Courier for their font, you can set the form's Font property to that font. By default, all the controls on report form will use the same font.

To specify a different font for a particular control, specify the desired font as the value of the band's Font property, and ParentFont becomes False automatically.

#### property Ruler : TQRRuler

QuickReport can paint a ruler in the background of the band during design time. This will help you position the controls on the bands. One of the following values can be set:

<b><i>qrrNone</i></b>	No ruler is displayed
<b><i>qrrInchesH</i></b>	Horizontal ruler lines every inch
<b><i>qrrInchesV</i></b>	Vertical ruler lines every inch
<b><i>qrrInchesHV</i></b>	Horizontal and vertical ruler lines every inch
<b><i>qrrCmH</i></b>	Horizontal rulers every centimetre
<b><i>qrrCmV</i></b>	Vertical ruler lines every centimetre
<b><i>qrrCmHV</i></b>	Horizontal and vertical ruler lines every centimetre

event AfterPrint : procedure (BandPrinted : Boolean)

This event is called whenever the band has been printed or tried to be printed but stopped in the BeforePrint event. The BandPrinted parameter will tell if the band was actually printed or not.

#### event BeforePrint : TQRBandBeforePrintEvent

The BeforePrint event is called before the band is about to be printed. You can set the PrintBand parameter to false if you want the band to be skipped for this time.

### **QRCustomControl**

The QRCustomControl object is the base class for all printable QuickReport components. It's not found on the component palette, instead you will subclass it if you want to create your own printable controls. There are three methods you need to override to create a printable control

#### property Caption

The QRCustomControl component has the basic capabilities of printing text, so it also has a Caption property. This property is not published by default.

#### method Paint

The Paint method is used to draw the component at *design time*. That is in the Delphi Form designer, not in the design time preview. If you want to print anything but basic text you must override this method.

#### method Print(x,y : integer)

The Print method is used to print the method onto the QRPrinter.Canvas. If you want to print anything but basic text you must override this method.

#### method Stretch(var Value : integer)

If your component has a variable height you must override the Stretch method. The Value parameter shall be set to the *extra* height your component will occupy, that is not including the value of the Height property.

### **QRDBCalc**



The QRDBCalc component eases the job of doing basic calculations during report generation. It can connect to a database field and calculations can be reset after the printing of any particular band.

#### property AlignToBand : Boolean

If True, the band will align horizontally to the entire band with, not only in its given rectangle. So if Alignment is taRightJustify and AlignToBand is True, the text will be right aligned on the band (page) instead of just within the control. Similarly, if Alignment is taCenter and AlignToBand is True, the text will be centred on the band (page). The default is False.

#### property AsInteger : Longint

Returns the current value of the QRDBCalc component as an Integer.

#### property AsReal : real

Returns the current value of the QRDBCalc component as a Real.

#### property Font : TFont

The Font property is a font object that controls the attributes of text printed to the report. To modify a font, you change the value of the Color, Name, Size, or Style properties of the font object.

#### property Operation : TQRCalcOperation

This property selects the type of calculation for the control to perform. It can be one of the following values:

<b>qrcSUM</b>	Calculate the total of the field
<b>qrcCOUNT</b>	Count the number of times the control is printed
<b>qrcMAX</b>	The maximum value of the database field
<b>qrcMIN</b>	The minimum value of the database field

**qrcAVERAGE** Calculates the average of the database field

#### property ParentFont : TFont

The ParentFont property of a printable control determines if the control will use its own Font property for font selection or if it will get font information from its parent QRBand.

#### property PrintMask : String

Use the PrintMask to format the output of the QRDBCalc component. Look up the FormatFloat procedure in Delphi help for documentation on how the PrintMask works.

#### property ResetBand : TQRBand

Setting the ResetBand property causes the calculation to be reset every time the selected band is printed. The reset will be performed after the band is printed.

#### event OnPrint : procedure (sender : TObject; var Value : String)

The OnPrint event is called just before the control is printed. The string to be outputted is in the Value property and can be changed. This property is also useful to suppress printing of repeated values.

### **QRDetailLink**



QRDetailLink is used to create different types of master detail reports with QuickReport. Using the QRDetailLink control you can create virtually any type of such reports, including unlimited levels and unlimited number of detail tables on the same level. You connect a QRDetailLink to a QuickReport component or another QRDetailLink using the Master property. In addition you must set up a master/detail relationship between the tables using the MasterSource/MasterField properties of the TTable/TQuery components. For more information on using the QRDetailLink please see *Creating a Master Detail Report* elsewhere in this document.

The QRDetailLink component is not intended to be used together with the QRGroup component. They are two different ways of creating master/detail reports and mixing them might give a wrong result.

#### property DataSource

The DataSource to connect this QRDetailLink to. A master/detail relationship must be set up for this to work. If you don't specify a DataSource you can use the OnNeedData event to print detail information from other sources, such as arrays or text files.

#### property DetailBand

Selects the DetailBand for this DetailLink. The detail band is printed for each record in the DataSource. The BandType of the selected band is automatically set to rbSubDetail.

#### property FooterBand

Selects the band to be used as this QRDetailLink's footer band. The footer band is printed after any detail records for this DetailLink. It might contain any summary information for the detail table. The BandType property of the selected band is automatically set to rbGroupFooter.

#### property HeaderBand

The HeaderBand is printed before the detail records of the DetailLink component. The BandType property of the selected band is automatically set to rbGroupHeader.

#### property Master : TQRController

The Master property connects to a TQuickReport component or another QRDetailLink. A master detail relationship must be setup between the tables involved.

#### property PrintBefore : Boolean

If PrintBefore is False (Default) the QRDetailLink component and its bands will be printed after any detail band of the Master. If True, this QRDetailLink will be printed before the Master information.

### event OnFilter : procedure (PrintRecord : Boolean) of object

The OnFilter event is used for local filtering of tables. The event handler will be called each time the record pointer is advanced through the dataset, allowing you to reject the record by setting the PrintRecord parameter to False. Please see *How do I filter Records* for more information.

### event OnNeedData : procedure (MoreData : Boolean) of object

The OnNeedData event is used if you want to print detail data from other sources than Delphi DataSources. To print such a report you would leave the QRDetailLink.DataSource property blank and provide an event handler for this event. The event handler would then retrieve the data from whatever source you want and place the result into the Caption property of QRLabels on your report. When there are no more data to print you set the MoreData parameter to False.

For more information on event driven reports please see *Creating Reports from non BDE Datasources*.

## QRDBText



The QRDBText component is a data aware text control. It can print both text fields, numeric fields and memo fields. If you connect it to a memo field it will automatically stretch to make room for the entire text. The band it's put on will also stretch accordingly. If the memo field is too large to fit on a page it will be clipped.

### property Alignment : TAlignment

The Alignment property specifies how text is aligned within the component.

These are the possible values:

- taLeftJustify**     Align text to the left side of the control
- taCenter**         Center text horizontally in the control
- taRightJustify**     Align text to the right side of the control

### property AlignToBand : Boolean

If True, the band will align horizontally to the entire band with, not only in its given rectangle. So if Alignment is taRightJustify and AlignToBand is True, the text will be right aligned on the band (page) instead of just within the control. Similarly, if Alignment is taCenter and AlignToBand is True, the text will be centred on the band (page). The default is False.

### property AutoSize : Boolean

When the AutoSize property is True, the component resizes to the width and length of the current string.

### property Color : TColor

Selects the background color for the control

### property DataSource : TDataSource

Selects the DataSource from which you want to print a data field.

### property DataField : String

Selects the Data field to be printed by the control.

### property Font : TFont

The Font property is a font object that controls the attributes of text printed to the report. To modify a font, you change the value of the Color, Name, Size, or Style properties of the font object.

### property ParentFont : Boolean

The ParentFont property of a printable control determines if the control will use its own Font property for font selection or if it will get font information from its parent QRBand.

### property Transparent : Boolean

The Transparent property determines if the control is transparent. You could place a transparent label or text control on top of a bitmap, and the control won't hide part of the bitmap. If Transparent is True, the Color property (for background color) is ignored.

### event OnPrint : procedure (sender : TObject; var Value : String)

The OnPrint event is called just before the control is printed. The string to be outputted is in the Value property and can be changed. This property is also useful to suppress printing of repeated values.

## **QRGroup**



The QRGroup component is typically used to print group headers and footers when some datafield change during report creation. This can be used to create master detail reports from SQL queries or simply to print a group footer when some data changes. The Simple Report in the demo program demonstrates how a QRGroup can be used to put a spacer band whenever the first letter of the company name changes.

You can nest QRGroup components up to ten levels. Whenever a group breaks, all higher level groups will break too.

### property DataSource : TDataSource

Select the datasource you want this group connected to.

### property DataField : String

Select the datafield you want this group to be connected to. The group will reprint when the value of the selected field change or a lower level group is reprinted.

### property FooterBand : TQRBand

Selects the band to be used as this group's footer band. The band type of the selected band is automatically set to rbGroupFooter.

### property HeaderBand : TQRBand

Selects the band to be used as the header band for this group. The BandType property of the selected band is automatically set to rbGroupHeader.

### property Level : Integer

Sets the level of this group. Group 0 is the lower group (outer) and 9 is the inner group. Please make sure not two groups are on the same level.

### Event OnNeedData : procedure (var Value : String)

The OnNeedData event can be used to modify the value of the group data before the decision is taken whether to break the group or not. The Simple Report in the demo program demonstrates how to use this event to make the group depend on the first letter in a company name instead of the entire field.

## **QRLabel**



Static text on reports are printed using the QRLabel component. Set the Caption property to the text you want to print. QRLabel will not wrap the text on several lines if it doesn't fit in the designed width. You can also change the caption property during report generation.



### property Alignment : TAlignment

The Alignment property specifies how text is aligned within the component. These are the possible values:

***taLeftJustify***     Align text to the left side of the control  
***taCenter***             Center text horizontally in the control  
***taRightJustify***     Align text to the right side of the control

### property AlignToBand : Boolean

If True, the band will align horizontally to the entire band with, not only in its given rectangle. So if Alignment is *taRightJustify* and AlignToBand is True, the text will be right aligned on the band (page) instead of just within the control. Similarly, if Alignment is *taCenter* and AlignToBand is True, the text will be centred on the band (page). The default is False.

### property AutoSize : Boolean

When the AutoSize property is True, the component resizes to the width and length of the current string.

### property Caption : String

The caption property contains the text string to be printed on the report.

### property Color : TColor

Selects the background color for the control

### property Font : TFont

The Font property is a font object that controls the attributes of text printed to the report. To modify a font, you change the value of the Color, Name, Size, or Style properties of the font object.

### property ParentFont : Boolean

The ParentFont property of a printable control determines if the control will use its own Font property for font selection or if it will get font information from its parent QRBand.

### property Transparent : Boolean

The Transparent property determines if the control is transparent. You could place a transparent label or text control on top of a bitmap, and the control won't hide part of the bitmap. If Transparent is True, the Color property (for background color) is ignored.

### event OnPrint : procedure (sender : TObject; var Value : String)

The OnPrint event is called just before the control is printed. The string to be outputted is in the Value property and can be changed. This property is also useful to suppress printing of repeated values.

## QRMemo



The QRMemo control is used to print multiline text not bound to a data field. QRMemo will *not* stretch to fit the height of the text automatically.

### property Alignment : TAlignment

The Alignment property specifies how text is aligned within the component. These are the possible values:

***taLeftJustify***     Align text to the left side of the control  
***taCenter***             Center text horizontally in the control  
***taRightJustify***     Align text to the right side of the control

### property Color : TColor

Selects the background color for the control

### property Font : TFont

The Font property is a font object that controls the attributes of text printed to the report. To modify a font, you change the value of the Color, Name, Size, or Style properties of the font object.

### property Lines : TStringList

The Lines property contains all the text of the QRMemo control. It's a standard TStringList and you can use methods like Clear and Add to enter text into the control.

### property ParentFont : Boolean

The ParentFont property of a printable control determines if the control will use its own Font property for font selection or if it will get font information from its parent QRBand.

### property WordWrap : Boolean

If WordWrap is True lines in the QRMemo will be wrapped to fit within the width of the control. If False, lines will be printed as they are entered in the Lines property and all text might not show.

## **QRPreview**



Using the QRPreview component it is very easy to make preview forms with your own look and feel. All you need to do to create a basic preview form is to drop a QRPreview on a form, make an event handler for the QRPrinter.OnPreview event and call the ShowModal event of your preview form. For more information on creating custom preview forms see *Creating a Custom Preview Form* on page 12.

### property PageNumber : integer

PageNumber selects the page to show in the preview component. The selected page will automatically be painted when setting this property.

### property Zoom : integer

Select the current zoom level with this property. It can be set to virtually any value, setting it to 100 will give you approximately a 1 to 1 reproduction of your page.

### method ZoomToFit

Calling ZoomToFit will automatically zoom the entire page will fit in the QRPreview controll. If you resize the control, the zoom value will be recalculated automatically.

### method ZoomToWidth

Calling ZoomToWidth will automatically zoom the page so it will fit just in the width of the QRPreview controll. If you resize the control, the zoom value will be recalculated automatically.

## **QRPrinter**

The QRPrinter is not a component on your component palette. Instead, one instance is created on application startup and is used throughout your application. It's the QRPrinter which takes care of all the low level stuff like recording your report as it's created, printing it, calling up the preview and even saving and loading reports.

QRPrinter can even be used independent of the QuickReport component. Using the QRPrinter.Canvas like any other canvas you can print just about anything, then use the QRPrinter.Preview method to bring up a preview before printing.

### property Canceled : boolean

The QRPrinter.Canceled property tells if the Cancel method has been called or not.

**property Canvas : TCanvas (Delphi 1.0) TMetafileCanvas (Delphi 2.0)**

The Canvas is actually a Windows metafile used for creating the pages.

**property Copies : integer**

Number of copies to print. Note that not all printer driver support this option.

**property EnableOpenBtn : boolean**

If True (Default) the Open Report button will appear on the standard preview form

**property EnablePrintBtn : boolean**

If True (Default) the Print report button will appear on the standard preview form.

**property EnableSaveBtn : boolean**

If True (Default) the Save report button will appear on the standard preview form

**property FromPage : integer**

The FromPage property can be used to set the starting page if you want QRPrinter to print only a range of the prepared pages.

**property LeftWaste : integer**

LeftWaste contains the unprintable area on the left edge of the page. It is only valid during report creation.

**property Orientation : TPrinterOrientation**

Use the Orientation property to select portrait (poPortrait) or landscape (poLandscape) orientation of the page. This property is set automatically when you choose QuickReport print.

**property Page : TMetafile**

When the report is prepared, the Page property points to a metafile containing the currently selected page.

**property PageCount : integer**

The PageCount property returns the total number of pages send to the QRPrinter object in this document.

**property PageHeight : integer**

QRPrinter.PageHeight is the height of the current printable area, in screen resolution.

**property PageNumber : integer**

The PageNumber property is used to select what page QRPrinter makes available via the Page property. It is set automatically when you change the value of a QRPreview.PageNumber property. Generally you will not change this property directly, but access the PageNumber property of the QRPrinter object, since this will also force a repaint of the preview.

**property PageWidth : integer**

QRPrinter.PageWidth is the width of the current printable area, in screen resolution.

**property PaperBin : integer**

Selects the output bin to be used for the printjob. The possible options are:

dm bin\_Upper

dm bin\_Lower

dm bin\_Manual

dm bin\_EnvManual

dm bin\_OnlyOne

dm bin\_Middle

dm bin\_Envelope

dm bin\_Auto

dmbin\_Tractor  
dmbin\_LargeFmt  
dmbin\_Cassette

dmbin\_SmallFmt  
dmbin\_LargeCapacity

### property PaperLength

Select a non zero value for this property will set up a custom paper length. The unit is 1/10<sup>th</sup> mm, making a value of 254 equal one inch. Setting this property automatically sets the PaperSize property to rpCustom. Selecting a predefined paper size will set this property to 0 again. You should always set the PaperWidth property if you set PaperLength. Setting the QuickReport.PaperLength property will automatically set this property when the report is printed.

### property PaperSize : TQRPaperSize

Selects the paper size to be used for this print job. Setting this to other than qrpDefault will override the setting in your printer driver. Note that not all printer drivers support all paper sizes. A paper size selected for a QuickReport component will automatically be carried over to QRPrinter when the report is printed.

<b><i>qrpDefault</i></b>	Use the default paper size for the currently selected printer driver.
<b><i>qrpLetter</i></b>	Letter size paper (8 1/2 x 11 in)
<b><i>qrpLegal</i></b>	Legal size paper (8 1/2 x 14 in)
<b><i>qrpA3</i></b>	A3 size paper (297 x 420 mm)
<b><i>qrpA4</i></b>	A4 size paper (210 x 297 mm)
<b><i>qrpA5</i></b>	A5 size paper (148 x 210 mm)
<b><i>qrpCustom</i></b>	Use the paper size set by the PaperWidth and PaperLength properties

### property PaperWidth

Select a non zero value for this property will set up a custom paper width. The unit is 1/10<sup>th</sup> mm, making a value of 254 equal one inch. Setting this property automatically sets the PaperSize property to rpCustom. Selecting a predefined paper size will set this property to 0 again. You should always set the PaperLength property if you set PaperWidth.

### property PreviewCaption : String

Sets the caption of the standard preview form. Note that this property is read only in the shareware version.

### property PrinterOK : boolean

If a printer is installed and available on your system the PrinterOK property is True, otherwise it's False.

### property ProgressCaption : String

Sets the caption of the progress form. Note that this property is read only in the shareware version.

### property ShowProgress : boolean

If True, a progress bar will appear during report printing, saving and loading. This property is automatically set by the QuickReport component.

### property Status : TQRPrinterStatus

Read Only. The Status property tells the current status of the QRPrinter object. As long as you don't use the QRPrinter directly you should not have to worry very much about the setting of this property. The possible values are:

<b><i>mpReady</i></b>	The QRPrinter object is ready to accept new jobs (BeginDoc)
<b><i>mpBusy</i></b>	The QRPrinter is currently active receiving a job
<b><i>mpFinished</i></b>	A job is finished (EndDoc is called) and can be previewed, printed or saved
<b><i>mpPrinting</i></b>	A job is currently being sent to the printer
<b><i>mpPreviewing</i></b>	A preview is currently active

### property Thumbs : integer

The Thumbs property selects how many report pages are to be printed across each sheet of paper. The default is one, causing one report page to be printed on each sheet. If you set it to two, four report pages (2x2) will be printed, three will print nine (3x3) and so on.

#### **property Title : String**

The title property is displayed in the Print Manager and in any network printer queue. It's set automatically by the QuickReport component if you not printing directly to the QRPrinter yourself.

#### **property ToPage : integer**

The ToPage property can be used to set the ending page if you want QRPrinter to print only a range of the prepared pages.

#### **property TopWaste : integer**

TopWaste contains the unprintable area on the top edge of the page. It is only valid during report creation.

#### **property TotalPageHeight : integer**

TotalPageHeight contains the total paper height, including the unprintable areas at top and bottom. The value is in screen resolution.

#### **property TotalPageWidth : integer**

TotalPageWidth contains the total paper width, including the unprintable areas at left and right edge. The value is in screen resolution.

#### **property ZoomButtons : boolean**

If True (Default) the standard preview form will have three buttons for selecting the zoom level, Fit In Window, 100% and Page Width. If False the preview form will instead have a combo box with some more choices.

#### **event OnPreview**

The OnPreview event is used if you want to create your own preview forms instead of using the standard preview. Please see *Creating a Custom preview form* on page 12 for more information on how to use this event handler.

#### **method BeginDoc**

If you want to use the QRPrinter object directly you must call BeginDoc before printing anything to the Canvas property.

#### **method Cancel**

Call the Cancel method if you want to cancel the current job.

#### **method CleanUp**

If you use the QRPrinter object directly you must call the CleanUp method when you are finished with a job to free it from memory.

#### **method EndDoc**

If you use the QRPrinter object directly call EndDoc to indicate that you have reached the end of the job and now want to print, preview or save the job.

#### **method Load(Filename : String)**

If you want to load a previously saved report you call the QRPrinter.Load method. The file must be a .QRP file of the correct format.

#### **method NewPage**

If you use the QRPrinter object directly you use the NewPage method to indicate that you want to create a new page. *Do not call this method in any events during a regular QuickReport creation, instead use the QuickReport.NewPage method which will make sure any page footers and headers are printed correctly.*

### method Preview

The Preview method brings up a print preview of the current job. The job has to be completed (EndDoc) before you can call the Preview method. Usually you will just call the preview method of your QuickReport component, which will take care of all preparation of the job.

### method Print

The Print method prints the current job. The job has to be completed (EndDoc) before you can call the Print method. Usually you will just call the Print method of your QuickReport component, which will take care of all preparation of the job.

### method Save(Filename : String)

If you want to save the current report to a file you can call the QRPrinter.Save method. The file will be saved as a .QRP file. This is a proprietary file format, but it will be documented in a future version.

## QRSysData



The QRSysData component is used to print various system information, like page number, date and time and report title.

### property Alignment : TAlignment

The Alignment property specifies how text is aligned within the component. These are the possible values:

<b>taLeftJustify</b>	Align text to the left side of the control
<b>taCenter</b>	Center text horizontally in the control
<b>taRightJustify</b>	Align text to the right side of the control

### property AlignToBand : Boolean

If True, the band will align horizontally to the entire band with, not only in its given rectangle. So if Alignment is taRightJustify and AlignToBand is True, the text will be right aligned on the band (page) instead of just within the control. Similarly, if Alignment is taCenter and AlignToBand is True, the text will be centred on the band (page). The default is False.

### property AutoSize : Boolean

When the AutoSize property is True, the component resizes to the width and length of the current string.

### property Color : TColor

Selects the background color for the control

### property Data : TQRSysDataType

The Data property is used to select the information displayed by the QRSysData component. The possible values are:

<b>qrsTime</b>	Current time in the format HH:MM
<b>qrsDate</b>	Current date in the format set by the global ShortDateFormat variable
<b>qrsDateTime</b>	Current date and time separated by a comma
<b>qrsPageNumber</b>	The current page number
<b>qrsReportTitle</b>	The report title as it appears in the TQuickReport.ReportTitle property
<b>qrsDetailCount</b>	Total number of records to be printed in report

*qrsDetailNo* Current record number to be printed

### property Font : TFont

The Font property is a font object that controls the attributes of text printed to the report. To modify a font, you change the value of the Color, Name, Size, or Style properties of the font object.

### property ParentFont : Boolean

The ParentFont property of a printable control determines if the control will use its own Font property for font selection or if it will get font information from its parent QRBand.

### event OnPrint : procedure (sender : TObject; var Value : String)

The OnPrint event is called just before the control is printed. The string to be outputted is in the Value property and can be changed. This property is also useful to suppress printing of repeated values.

### property Text : String

Any text you want to print immediately in front of the data can be put into the Text property. Typically you would set this to 'Printed at : ' for a QRDBCalc component with the Data property set to qrsDateTime.

### property Transparent : Boolean

The Transparent property determines if the control is transparent. You could place a transparent label or text control on top of a bitmap, and the control won't hide part of the bitmap. If Transparent is True, the Color property (for background color) is ignored.

## QRShape



The QRShape component is used to draw simple shapes on your reports.

### property Brush : TBrush

Selects the Brush to be used for painting the shape. The Brush property has two sub properties, Color and Style. The Brush.Color selects the color to use when painting and the Brush.Style selects the actual Brush to use. The possible styles are:

<i>bsSolid</i>	Solid brush with the current color
<i>bsClear</i>	No brush is used
<i>bsBDiagonal</i>	Horizontal lines upwards to the right
<i>bsFDiagonal</i>	Horizontal lines upwards to the left
<i>bsCross</i>	Cross pattern
<i>bsDiagCross</i>	Diagonal cross pattern
<i>bsHorizontal</i>	Horizontal lines
<i>bsVertical</i>	Vertical lines

### property Height : Integer

The Height of the shape

### property Pen : TPen

The pen to use when drawing the outline of the shape.

### Property Shape : TQRShapeType

Use the Shape property to select what kind of shape you want to print. The possible values are:

<i>qrsRectangle</i>	Draws rectangles on the report
<i>qrsCircle</i>	Draws circles and ellipses on the report
<i>qrsVertLine</i>	To draw vertical lines on the report

**qrsHorLine** To draw horizontal lines  
**qrsTopAndBottom** Draws horizontal lines on the top and bottom of the shape  
**qrsRightAndLeft** Draws vertical lines on the right and left edge of the control

### property width

The width of the shape

## QuickReport



The QuickReport component is your report controller. Its job is to transform your Delphi form into a report. After you have dropped this component on the form you can call its *Print* and *Preview* methods to bring up the report. During design time you can right click the QuickReport component to bring up a menu. From here you can preview your report at design time, without running your application.

Please note that dropping a QuickReport component on your form will resize it to make add scrollbars to your form. This is on intention to make it easier to do report layouts.

### property PageCount : Longint

Runtime and read only. This property tells the total page count of the report. Note that it is not a valid value before the report is prepared.

### property PageNumber : Longint

Runtime and Read Only. PageNumber tells the current page number during report creation. This is the same value displayed by a QRSysData with the Data property set to PageNumber.

### property RecordNo : longint

Runtime and Read Only. RecordNo indicates the current record number in your (master) table. This is the same value displayed by a QRSysData component with the Data property set to RecordNo.

### property ColumnMarginInches : Longint

The ColumnMarginInches property sets the spacing put between columns in a multi column report. The spacing is set in 1/10th of inches. Use this property together with the LeftMarginInches to make a mailing labels report fit on your label layout.

### property ColumnMarginMM : longint

This is the same as the ColumnMarginInches property, except the value is set in MM rather than inches.

### property Columns : integer

Sets the number of columns to print on the page. The column width is set automatically depended on the Columns property, the LeftMargin and the ColumnsMargin properties.

### property DisplayPrintDialog : Boolean

If True, a print dialog will be displayed when you call the QuickReport.Print method. *Note that the print dialog will not be displayed if you click the Print button from the standard preview form, regardless of the setting of this property.*

### property LeftMarginInches : Longint

The LeftMarginInches property sets the left margin of the page in 1/10th of inches. The unprintable area of the printer is added to this value automatically.

### property LeftMarginMM : Longint

This is the same as the LeftMarginInches property, except that the value is given in MM rather than inches.

### property OnStartPage : TNotifyEvent



The OnStartPage event is called whenever a new page is started.

### property Orientation : TPrinterOrientation

The Orientation property selects the paper orientation to use when printing the report. The value can be poPortrait (default) or poLandscape. Note that to set this value from your source code you need to include the Printer unit in your uses clause, since the TPrinterOrientation type is declared in that unit.

### property PageFrame : TQRFrame

Controls the drawing of a frame around the entire page. You can set the top, bottom, right and left sub-properties to True or False to select what part of the frame you want to draw. The line style can be set with the Style and Color sub-properties.

### property PaperLength

Select a non zero value for this property will set up a custom paper length. The unit is 1/10<sup>th</sup> mm, making a value of 254 equal one inch. Setting this property automatically sets the PaperSize property to rpCustom. Selecting a predefined paper size will set this property to 0 again. You should always set the PaperWidth property if you set PaperLength.

### property PaperSize : TQRPaperSize

Selects the paper size to be used for this report. Setting this to other than qrpDefault will override the setting in your printer driver. Note that not all printer drivers support all paper sizes.

<b><i>qrpDefault</i></b>	Use the default paper size for the currently selected printer driver.
<b><i>qrpLetter</i></b>	Letter size paper (8 1/2 x 11 in)
<b><i>qrpLegal</i></b>	Legal size paper (8 1/2 x 14 in)
<b><i>qrpA3</i></b>	A3 size paper (297 x 420 mm)
<b><i>qrpA4</i></b>	A4 size paper (210 x 297 mm)
<b><i>qrpA5</i></b>	A5 size paper (148 x 210 mm)
<b><i>qrpCustom</i></b>	Use the paper size set by the PaperWidth and PaperLength properties

### property PaperWidth

Select a non zero value for this property will set up a custom paper width. The unit is 1/10<sup>th</sup> mm, making a value of 254 equal one inch. Setting this property automatically sets the PaperSize property to rpCustom. Selecting a predefined paper size will set this property to 0 again. You should always set the PaperLength property if you set PaperWidth.

### property ReportTitle : String

Sets the title for this report. The report title will appear in the Print Manager and in any network printer queue. It is also the value printed by a QRSysData component with the Data property set to ReportTitle.

### property RestartData : Boolean

If True (Default) QuickReport will move to the first record in the dataset before printing the report. If False, report creation will start at the current record.

### property ShowProgress : Boolean

If True (Default) a progress form will appear while the report is being prepared and printed. The progress form has a Cancel button the user can press to abort the current process.

### property SQLCompatible : Boolean

Not all databases allow the retrieval of the number of records in the table, and the BDE will create an exception if the operation is tried with a database that doesn't support it. Setting the SQLCompatible to True will allow QuickReport to work with such databases. Note that the progress gauge will not move during report creation if SQLCompatible is True.

### **property TitleBeforeHeader : Boolean**

If False(Default) the Page header band is printed first also on page one. If you want the Title band to appear before the page header band on the first page, set this property to True.

### **method NewPage**

Calling the NewPage method during report creation forces QuickReport to insert a page break at the current position. This can typically be done in the AfterPrint event of a band.

### **method Prepare**

The Prepare method is used to Prepare the report and send it to the QRPrinter object. It is called automatically by the Print or Preview methods, but you might want to call it yourself if you want to create a report without printing or previewing it. If you do so, remember to call the QRPrinter.CleanUp method when you are finished using the report.

### **method Preview**

Call the Preview method to bring up a print preview of your report. If you haven't created an event handler for the QRPrinter.OnPreview event a default preview form will be displayed. Otherwise it's up to you to display a preview form, usually one with a QRPreview component on it.

### **method Print**

The Print method will prepare and print your report. If the DisplayPrintDialog property is True a Print Dialog will be displayed before the report is prepared.

### **event OnEndPage : TNotifyEvent**

The OnEndPage event is called when when QuickReport is about to move to a new page.

### **event AfterDetail : TNotifyEvent**

The AfterDetail event is called when the detail bands are printed for a (master) record.

### **event AfterPrint : TNotifyEvent**

The AfterPrint event is called when the report creation is finished. You can use it to close any datasets you opened in the the BeforePrint event.

### **event BeforeDetail : TNotifyEvent**

The BeforeDetail event is called just before the detail bands for a (master) record are printed.

### **event BeforePrint : procedure (var PrintReport : Boolean)**

The BeforePrint event is called before any preparation of the report is started. Actually, the operation can be cancelled by setting the PrintReport parameter to False.

### **event OnFilter : procedure (PrintRecord : Boolean) of object**

The OnFilter event is used for local filtering of tables. The event handler will be called each time the record pointer is advanced through the dataset, allowing you to reject the record by setting the PrintRecord parameter to False. Note that the QRDetailLink component has it's own OnFilter event to filter records in a detail section of a report. Please see *How do I filter Records* for more information.

### **event OnNeedData : procedure (MoreData : Boolean) of object**

The OnNeedData event is used if you want to print data from other sources than Delphi DataSources. To print such a report you would leave the QuickReport.DataSource property blank and provide a event handler for this event. The event handler would then retrieve the data from whatever source you want and place the result into the Caption property of QRLabels on your report. When there are no more data to print you set the MoreData parameter to False.

Note that the QRDetailLink component also has a OnNeedData event, which can be used to print event

driven data in a *detail section* of a report.

For more information on event driven reports please see *Creating Reports from non BDE Datasources*.

## QuickReport Registration Form

To purchase the 16 bit version of QuickReport and the source code for both the 16 bit and 32 bit version of QuickReport fill out the registration form bellow and send or fax it to QSD AS

\_\_\_\_\_  
Date

\_\_\_\_\_  
Company Name

\_\_\_\_\_  
Contact

\_\_\_\_\_  
Postal address

\_\_\_\_\_  
Zip Code

\_\_\_\_\_  
State, Country

Send by Postal mail (Add \$10) : \_\_\_\_ E-mail : \_\_\_\_

E-mail address : \_\_\_\_\_

Payment method:

Included Check : \_\_\_\_ Included Cash/Money Order : \_\_\_\_

Visa : \_\_\_\_ MasterCard : \_\_\_\_ CompuServe SWREG : \_\_\_\_

Credit Card number : \_\_\_\_\_ Exp. Date : \_\_\_\_\_

Signature : \_\_\_\_\_

The registration fee is **\$79.00** US and registration can be done by sending in this form. Registered users can download the registered version and source code from Internet or get it by postal mail for an additional US \$10.

The CompuServe SWREG number for QuickReport is 8644.

For payment with Credit Card or through SWREG this registration form can be faxed to QSD at fax:

**+47 22 41 74 91**

For Cash or Money order the Registration form must be send together with payment to:

**QSD AS, Radhusgt. 9, 0151 Oslo, Norway**

Full registration information and an online registration form can also be found at our web site.

### QuickReport Mailing List

If you want to receive an e-mail notifications when new versions of QuickReport are released please send a

e-mail to [qmail@qsd.no](mailto:qmail@qsd.no) with *SUBSCRIBE QUICKREPORT* as message subject. Please do not use this address for questions or comments about QuickReport.

## Files you can download from our internet server

Over the coming weeks the following will be made available for download from our Web site, <http://www.qsd.no>:

- International versions of the help file and resource files (German, French, Italian)
- The QuickReport FAQ
- QuickReport 1.0XBDE, a QuickReport that runs totally independent of the BDE
- Updated 16 and 32 bit versions of QuickReport
- The documentation in Word and HTML format

## 3<sup>rd</sup> party QuickReport printable components

Several QuickReport users have created some nice printable components based on the TQRCustomControl class and are willing to share those components with other QuickReport users. You can download these components from our Web site, <http://www.qsd.no> or our ftp server, <ftp://www.qsd.no>. If you want to submit a component you have created please follow the instructions on the web site.

## Version history

### Version 1.0b

Version 1.0b is available as both 16 and 32 bit. The full 32 bit version is free upgrade for Delphi 2.0 users. The 16 bit version is shareware and users who have registered previous version are entitled to a free upgrade.

- \* New properties QRPrinter.PaperWidth, QRPrinter.PaperLength and QRPrinter.PaperSize, QRPrinter.Copies, QRPrinter.PaperBin
- \* New properties QuickReport.PaperSize, QuickReport.PaperWidth, QuickReport.PaperLength
- \* New properties QRBand.Frame.DrawTop, DrawBottom, DrawLeft, DrawRight for controlling which part of the frame to draw
- \* New property QuickReport.PageFrame lets you print (top/bottom/right/left) frame around the entire page
- \* Bug fix, correct data record is now always printed in the last group footer
- \* Bug fix, printmask now works correctly on QRDBCalc with integer fields
- \* Bug fix, using an overlay band should no more cause an extra page to be printed
- \* Bug fix, using the QuickReport.NewPage will now position text correctly on the new page
- \* Bug fix, band frames now works correctly with left margin set and also with pen width=1
- \* Bug fix, saving and loading landscape reports now works correctly
- \* Bug fix, group headers/footers are now printed correctly even if first group data is blank
- \* Cosmetic updates to the toolbar on the standard preview form

### Version 1.0a

This is the generally distributed 16 bit version. It has not been available as 32 bit for Delphi 2.0

- \* Bug fix, the OnPrint event of QRLabel, QRDBText, QRDBCalc now works correctly

### Version 1.0

This version is distributed with Delphi 2.0. It's generally not distributed as a 16 bit.

### Beta versions

To avoid confusion the version history for beta versions are not included.

### Credits

QuickReport and the documentation is written by Allan Lochert.

Thanks to hundreds of users on the Internet and CompuServe who have submitted bugs and feature requests, struggling with the not-always-so-good beta versions of QuickReport. Without their help the creation of QuickReport would not have been possible.

Thanks also to Borland International for their great support and for making the 32 bit version of QuickReport available with Delphi 2.0.