

QuickReport for Delphi

version 0.97a January 1996
(c) copyright 1995,1996 QSD as and Allan Lochert, All Rights Reserved
World Wide Web <http://www.qsd.no>
Compuserve 101372,3471
Internet allan@qsd.no
SWREG# 8644

**!! This is a preliminary documentation - there are some inconsistency between
!! the components provided and this text. Please bare over with this until the
!! product is more complete.**

Table of Contents

This document is divided into the following sections

I Introduction

- About QuickReport
- Copyright, Distribution and Registration
- QuickReport Mailing List

II Installation

- Installing QuickReport
- Installing the report template
- The demo program

III Using QuickReport

- Creating Reports
- Master/detail reports
- Creating a custom preview
- How do I ...
 - format output of a data field?
 - print master detail reports with several detail tables?
 - combine several reports into one?
 - filter records?
 - make a group keep an entire group on one page?
- Why are my...
 - fonts clipped or displayed at a wrong size during preview?
 - reports suddenly rescaled, all the fonts have changed size?

IV Component Reference

- QRBand
- QRCustomControl
- QRDBCalc
- QRDBText
- QRDetailLink
- QRGroup
- QRLabel
- QRPreview
- QRPrinter
- QRShape
- QRSysData
- QuickReport

V Bug list

VI Version history

VII Getting Technical Support

VIII Registration

SWREG

Credit Card

Check, cash

Money transfer

PART I Introduction to QuickReport

Introduction

QuickReport for Delphi is a package of several components to make it easy to include printed reports and print preview in your Delphi Programs.

QuickReport is fast and adds a very small footprint to your programs. There are no extra files to distribute since QuickReport links right into your executable file. While not as powerful as some standalone report generators, QuickReport should cover most of your printing needs. In addition, since QuickReport is so tightly integrated with Delphi there are several things you can do with QuickReport that is hard to accomplish with other products.

QuickReport is developed using Delphi 1.0 running under Windows 95. There are no compatibility problems with Windows 3.11 or 3.1. A 32 bit version will be available when Delphi 2.0 is released.

If you are upgrading from a previous version of QuickReport please read the version history at the end of this document for any changes you need to make to your reports or your code. If you come across a bug, please check that it is not in the list of known bugs later in this document before reporting it.

This is some of QuickReport's features

- * Design your reports visually using the Delphi form designer
- * Powerful on-screen preview
- * Reports can be viewed and printed from the Delphi form designer without running your program
- * Reports are build up of bands and may contain Title bands, page header and footer bands, multiple detail bands and summary bands in addition to group header and footer bands.
- * Use multiple detail bands. Print them all or enable/disable them depending on the data to be printed
- * Report from any DataSource, including TTable and TQuery or even from arrays, lists or whatever data you want
- * Ten levels of fully automatic data grouping. Print headers and footers for data groups
- * Master / detail reports can have any number of detail tables
- * Print mailing labels and multi column reports
- * Print labels, database fields, memo fields, shapes, bitmaps, metafiles and icons
- * Perform calculations like summary and counting of fields automatically. Calculations can be reset at any grouping level
- * Automatically scale down the pages to print several report pages on each sheet of paper
- * Finished reports can be saved to files and later loaded again for viewing or printing
- * Use any Windows font, size, style and color
- * Use BeforePrint and AfterPrint events with report bands to perform tasks like data lookup and calculations
- * Portrait or landscape printing of reports

The QuickReport package consists of the following Delphi components

TQuickReport	This is your report "controller"
TQRPrinter	A low level printer object with print, preview and other functions
TQRDetailLink	
TQRBand	Use QRbands to create the different parts of your report
TQRGroup	Control data grouping
TQRLabel	Put a QRLabel on a band to print text
TQRDBText	Like QRLabel, but connects to a datasource to print data fields
TQRDBCalc	Perform basic calculations like summing and counting on data fields
TQRShape	Print rectangles and circles
TQRSysData	Printing system data like time, date, page number and more
TQRPreview	Preview component to use in your own preview forms

Things to come in the future

- * Help file
- * Copy report to clipboard
- * Multiple page memo fields
- * Runtime report designer

Copyright, Distribution Policy and Registration

QuickReport for Delphi will be freeware until version 1.0. From then on QuickReport will be shareware and there will be a registration fee of US \$65 or NOK400 (Norwegian kroner). Registered users will receive full source code and will be entitled to royalty free use and distribution of QuickReport in their own programs. Version 0.95a is free to use in your programs.

If you distribute QuickReport you must do so without changing anything in the original QuickReport zip file. You use this program at your own risk. I am not responsible for any loss or damage caused by the use of QuickReport. QuickReport is copyright 1995 Allan Lochert.

Registration details is at the end of this document.

The latest version of QuickReport can always be found at <ftp.qsd.no/pub/qsd/QREPxxx.zip> where xxx will be the current version number.

Please send bug reports by e-mail to allan@qsd.no. Your comments and thoughts about QuickReport are also most welcome.

Remember that until version 1.0 *THIS IS BETA SOFTWARE. There are bugs and all features might not work 100% as planned - yet.*

QuickReport Mailing List

If you want to receive a notification when new releases of QuickReport is available send a mail to qmail@qsd.no with SUBSCRIBE QUICKREPORT as message subject.

Installation

This is a brief description of how to install QuickReport on your system

- * Copy all QuickReport files to some directory, say DELPHI\QUICKREP
- * Start Delphi
- * Choose Options | Install Components
- * Click Add
- * Type your QuickReport directory followed by QUICKREP.DCU; C:\DELPHI\QUICKREP\QUICKREP.DCU
- * Click OK
- * Click OK

You should now have a new page on your component palette called Report. There you will find

several new components.

If you want to copy only the necessary files to your DELPHILIB directory and install them from there then these are the files to copy: QUICKREP.DCU, QUICKREP.DCR, QRPREV.RES

For more information on how to install components look up "Adding and Removing Components from the Library" in the Delphi online help.

Installing the report template

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

- * Open the file QRTEMPL.PAS in the form designer
- * Choose Options | Gallery
- * Choose Form Templates tab
- * Type "QuickReport template"
- * Click Browse
- * Locate the QRTEMPL.BMP file and click OK
- * Click OK

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

The QuickReport Demo Program

There is a simple demo program included with QuickReport to show you some basic capabilities and help you get started with QuickReport.

The project is called DEMO.DPR and is in your QuickReport directory.

The demo program uses the Customer table installed in the DBDEMOS alias when you installed Delphi. If you choose not to install the demo programs with Delphi the demo will not run.

There are four reports you can view or print in this demo:

- * Simple report - a simple customer listing, showing how to create a very basic report.
- * Memos and Pictures - a report using the Biolife table, showing how QuickReport can handle memo fields and images from blob fields
- * Master/detail - an example on how to create master/detail reports using the QRDetailLink component. This report is a three level master/detail report using the Customer, orders and Items tables. All order lines are printed for all orders for all customers. Several examples on how to perform calculations during report generation can also be found here. This report is about 35 pages and will give some impression on the performance you get with QuickReport.
- * Mailing Labels - shows how to create multi column reports, including mailing labels.

The examples is intended to be a valuable resource of information on creating reports with QuickReport.

PART III USING QUICKREPORT

QuickReport Basics

QuickReport is a *banded* report generator. Your reports are build up of several different kind of bands, onto which you put visible components like database fields, labels, images or any other printable component. < more info to come >

Creating reports

QuickReport can create several types of reports, in fact with a little knowledge of Delphi and some creativity QuickReport can be used to print just about *any* type of report! Remember that even if reports are created as forms they are not intended to be shown on screen as forms. A report form must include a TQuickReport component, and you use this component's PRINT and PREVIEW methods to create your report. Do not try to bring up a report using the form's SHOW or ShowModal methods. A report form should never be your application's main form.

Simple list reports

Creating a report which simply lists the contents of a TTable or a TQuery is very easy. Follow the step by step instructions below to create a report listing all the customers in the DBDemos.Customers table included with Delphi.

1. Create a new project with a form and a single button on the form. This is your application's main form. Save the form unit as MAINFORM and the application as QRTEST
2. Add a blank form to the project. Set the name property to MYREPORT. Set the form width to about 800. The actual number is not critical
3. Add a TTable component to the form. Name the component CustomerTable, set the DataBaseName property to DBDEMOS and the TableName property to CUSTOMER. Set the Active property to TRUE.
4. Add a TDataSource component to the form. Name it CustomerDS and set DataSet property to CustomerTable
5. Add a TQuickReport component to the form. Name the component REP. Set the DataSource property to CustomerDS
6. Add a QRBand component to the form. Set the BandType property to rbDetail.
7. Put a TQRDBText component on the band. Set the DataSource property to CustomerDS and the DataField property to CompanyName
8. Save the form as REPORT
9. Go back to your MAINFORM. Add MYREPORT to the USES clause at the top of the source code
10. Double click the button on the form. This will bring up the Button1.OnClickEvent. Add the following code to the event: MyReport.Rep.Preview;
11. Run your project

You have now created a project with a report. Your program can now preview, print, save and load reports!

Master/Detail reports

QuickReport can print master/detail reports very easily. Master/detail reports can be printed from both TQuery components and from TTable components connected with the MasterSource / MasterField properties. In addition QuickReport can create complex master / detail reports with records from unlimited number of detail tables.

You will see how to create both types of reports below

Master / Detail report using TTable components

Creating a master / detail with TTable components gives very fast reports. In addition you can include detail data from as many detail tables you want. You could create a customer summary report listing all your customers (master data) and for each customer list all the orders the customer has placed (detail data), all your contact persons with the customer (detail data), all the product categories the customer is interested in (detail data) and any other table you would want to link to the customer table. Each of these detail tables can contain their own header band, detail band and footer band.

To create a master detail report using TTable components follow the instructions below. You might also want to take a look at the MDREP report in the demo project. It is a master detail report created using this approach.

1. Create a standard report following the example in the above chapter, Creating Reports.
2. Change the MyReport.ReportType property to qrMasterDetail
3. Add a new TTable component to the report. Name it OrdersTable, set the DataBaseName property to DBDEMOS, the TableName property to Orders, the MasterSource to CustomerDS and in the MasterFields dialog connect the secondary index (custno) to Customer.Custno. Set the Active property to TRUE
4. Add a TDataSource component to the report. Name it OrdersDS and set the DataSet property to OrdersTable
5. Add a TQRBand to the report. Set the BandType property to rbSubDetail. Name it OrdersDetailBand
6. Add a TQRDetailLink component to the report. Set the DataSource property to OrdersDS. Set the DetailBand property to OrdersDetailBand. Set the Master property to REP.
7. Put a TQRDBText component on the OrdersDetailBand. Set the DataSource property to OrdersDS and the DataField property to OrderNo
8. Double click the QuickReport component to preview the report

If you wanted more detail tables you would repeat step 3 to 7 for each detail table to include. To change the printing order of the tables rearrange the creation order of the TQRDetailLink components. The detail tables are printed in the order corresponding to the creation order to the QRDetailLink components.

Master / Detail reports using a TQuery component

Please see TQRGroup for some information on this. < more info to be added >

Reports from Arrays and other datasources

... To be added ...

Creating Custom Preview

A professional looking print preview is essential to the overall feel of your product. One nice feature of QuickReport is the ability to create your own preview forms. QuickReport is intended to integrate 100% with your application and to enable that you can very easily create your own report preview using the TQRPreview component.

There is also a very simple preview form included with the demonstration program. Take a look at DEMOPRE.PAS.

Basically creating a preview form consist of two steps

- * Create a form with a TQRPreview component, plus any other components you would want.
- * Show the your preview form in the QRPrinter.OnPreview event.

Following is a step by step guide to create a very simple preview form:

1. Open the Test project created above
2. Select File | New Form and double click on a blank form if you get the form gallery
3. Set the Name property of the form to MyPreviewForm
4. Drop a TQRPreview component on the form
5. Set the Align property to alClient
6. Select File | Save file and save the unit as MYPREV
7. Open the MAINFORM and view the source code
8. In the uses clause at the top add QUICKREP and MYPREV to the end of the list
9. Insert the following line in the MAINFORM.OnCreate event:

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

10. Add The following procedure to the end of the code:

```
procedure MyPreview;  
begin  
    MyPrevForm.ShowModal;  
end;
```

13. Compile the project and preview the simple demo report.

You have now created a preview form that shows the first page of the report at 100% of the size. Maximise the form, and you see more of the report.

Now you can add an action bar to your preview form. Put components on it to change the PageNumber and Zoom properties of the aPreview component.

How do I ...?

This sections gives answers to some of the most commonly 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 print master detail reports with several detail tables?

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?

Check out the TQuickReport.OnFilter event and the corresponding TQRDetailLink.OnFilter event. These events get called every time the current record is changed and can be used to discard records. < more info to come >

How do I keep an entire group on one page?

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

QuickReport works best with TrueType fonts. To be able to create a good preview you need to use a font which can be resized. Try using only TrueType or PostScript fonts in your reports.

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

Set the TForm.Scaled property to False for your report forms. This will prevent Delphi from trying to rescale your to fit on screen.

PART IV QuickReport Component Reference

The components are listed in alphabetic order.

TQRBand component

You build your reports by creating different QR Bands and putting visible QuickReport controls on them. There are several different band types, like detail band, page header band, page footer band and so on.

QRBand properties and methods:

property AfterPrint : procedure (BandPrinted : Boolean) of object
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.

property BandType : TQRBandType
BandType selects when this band will be printed.

* rbTitle

This band is printed once at the start of the report. The QuickReport.HeaderBeforeTitle property decides if the PageHeader or Title band will appear first.

* rbPageHeader

Page header bands appear on top of each page. The QuickReport.HeaderBeforeTitle property decides which band will be printed first on the first page. You can have as many Page header bands you want.

* rbDetail

The Detail band gets printed once for every record in your dataset. If you design a master/detail report using the QRDetailLink component this band will be the detail band for your master table. You can have as many detail bands you want.

* rbDetailGroup

If you create a master/detail report using the QRDetailLink component then this band will be the Detail band for your Detail Table(s). You must connect the DetailGroup band to the DetailLink component using the QRDetailLink.DetailBand property. This band can also be used as header and footer bands for the QRDetailLink component.

* rbPageFooter

This band is printed on the bottom of every page.

* rbSummary

The Summary band is printed on the last page of the report, after all detail bands and group footers.

* rbGroupHeader

* rbGroupFooter

* rbColumnHeader

If you print a multi column report this band will be printed on top of each column.

* rbOverlay

If you want some static text or graphic to be printed on every page of your reports then you can use the rbOverlay band. This band will be printed once on each page and it overlay all other text and graphic printed on the page.

property BeforePrint : procedure (var PrintBand : Boolean) of object
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.

property Color
Sets the background color of the band.

property Enabled : Boolean

Enables and disables the printing of this band. If this property is false the BeforePrint and AfterPrint events will be skipped. You can change this property during report progress, for instance to print different detail bands depending on the type of data to print.

property Font

The default font for the band. Use a True Type font to ensure that QuickReport is able to scale the font properly during preview.

property ForceNewPage : Boolean

property Frame : TPen

property Ruler : TQRRuler

This property will tell QuickReport to display 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:

- * qrrNone No ruler is displayed
- * qrrInchesH Horizontal ruler lines every inch
- * qrrInchesV Vertical ruler lines every inch
- * qrrInchesHV Horizontal and vertical ruler lines every inch
- * qrrCmH Horizontal rulers every centimetre
- * qrrCmV Vertical ruler lines every centimetre
- * qrrCmHV Horizontal and vertical ruler lines every centimetre

QRCustomControl Component

The QRCustomControl component is the base class for all printable QuickReport components. If you want to create your own printable components you should make them children of this class. QRCustomControl has two virtual methods you might override.

procedure Print(X,Y : Integer); virtual;

The Print method prints the component to the QRPrinter object at the given position. The default print method prints the text of the protected Caption property. If you want another behaviour you will need to override this method.

procedure Stretch(var size : Integer); virtual;

If your component will resize itself vertically at printing time, like a stretching memo field you must return the new component height with this method. If your component will have a static size you don't have to worry about this method.

property AutoSize

property height

The height of the control in pixels.

property Left

The left horizontal position of the control in pixels, counting from the left margin.

property Top

The vertical position of the control within the band it is printed on.

property Visible

If True (default) the control is printed. If False, the control is not printed.

property width

The width of the control in pixels.

TQRDBCalc Component

This component automates the process of summing and counting of data fields. In addition to these properties the QRDBCalc has all the properties and events it inherits from TQRDBText.

property AsInteger : Longint read only

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

property AsReal : real read only

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

property Operation : (qrcSUM, qrcCOUNT)

Set the operation you want to perform.

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 ResetGroup : TQRGroup

The QRDBCalc will be reset when the group it is connected to breaks.

TQRDBText Component

The QRDBText component is a data aware text control. Text is taken from a database field.

property DataSource : TDataSource

The datasource you want to connect this control to.

property DataField : String

The datafield you want to connect this control to.

In addition the QRDBText component has all the properties from QRCustomControl and QRLabel, except the Caption property.

If you want to format the output of the QRDBText component you do so by adding a non visible datafield component. Double click the TTable or TQuery component to bring up the field components editor and add all fields from the datasource. Then find the field you want in the object inspector. If your table is called MyTable and the field is Name, you will find a component called MyTableName. This component has a DisplayMask property. Use this property to format the output.

```
TQRCalcOperation = (qrcSUM, qrcCOUNT, qrcMAX, qrcMIN, qrcAVERAGE);
```

- | | |
|--------------|--|
| * qrcSUM | Sums the value |
| * qrcCOUNT | Counts the number of times this component is updated |
| * qrcMAX | Displays the maximum value of this field |
| * qrcMIN | Displays the minimum value of this field |
| * qrcAVERAGE | Calculates the average of this field |

TQRDetailLink Component

QRDetailLink is the second way of creating master/detail reports with QuickReport. Typically you will use QRDetailLink with TTables (or TQueries) when you have only two levels of linking. The strength of QRDetailLink however, is that you can have as many detail tables you want at the same level. Consider the following report layout

```
Customer information (master table)
  Contact person at the customer (first detail table)
  Another contact person

  Customer order (second detail table)
  Another customer order
  Summary of customer orders

  Project the customer is involved in (third detail table)
  Another project the customer is involved in

Next customer
. . . .
```

This kind of report can very easily be created using the QRDetailLink component. You would place four TTables on your report, Customer, Contacts, Orders and Projects. You would also place a TDataSource for each of these TTables. Then you would place three QRDetailLink components on the report and connect them to Contacts, orders and Projects. Each of these QRDetailLink components can then have its own Header, detail and footer band. Before you print the report you would then set up the usual master/detail relationship using the MasterSource/MasterField properties of the TTable components.

The QRDetailLink component can also be used if you want to write a report which actually looks like several reports in a row.

```
property DataSource : TDataSource
The datasource to connect this QRDetailLink to
```

```
property DetailBand : TQRBand
The band to use as this QRDetailLink's Detail band. It is printed for every record in the detail table matching the current master record, provided you have set up the master/detail relationship correctly of course. See the Delphi manuals for help on how to do this if you are unsure.
```

```
property FooterBand : TQRBand
The band to use as this QRDetailLink's footer band. The footer band is printed after all detail records for the current master record.
```

```
property HeaderBand : TQRBand
The band to use as this QRDetailLink's Header band. The header band is printed before all the detail records for the current master record.
```

```
Property OnNeedData : procedure (var MoreData : Boolean) of object
If you want to print from an array or some other source except a DataSource you can use the OnNeedData event to retrieve the information. See the QuickReport.OnNeedData event for a more in depth explanation on this.
```

TQRGroup

There are two ways of printing master/detail reports with QuickReport. One is to use a TQuery component with a join Query and QRGroups to group the data, the other is to use the DetailLink component combined with DetailGroup bands. Each method has its unique strengths. If you want master/detail reports with more than two 'layers' then using QRGroup and TQuery will probably be the best approach. Imagine this report layout:

```
Customer information (master table)
  Contact person at customer (first detail table)
    Orders the contact has placed for the customer (second detail table)
```

```
    Sum all orders the person has placed
  Next contact person at the same customer
    His orders
    His total orders
Total orders for the entire customer
Next customer
  Contact person
    Orders
. . . .
```

This report has one master table (customers) and two detail tables (contacts and orders). You would then join them in a query like this:

```
Select * from customer a, contacts b, orders c
where a.custno=b.custno and b.contactno=c.contactno
order by custno, contactno, orderno
```

You would then put two QRRGroup components on the form, connect the first to `custno` and the second to `orderno`.

For simple one level master detail reports or reports with several tables at the same level see the QRDetailLink component.

QRGroup properties

property DataSource : TDataSource
Select the datasource you want to connect this group to.

property DataField : String
Select the data field you want to connect this group to. The group will reprint when the value of Datafield change or a higher level group is reprinted.

Property FooterBand : TQRBand
Select a band to be printed as this groups footer band.

Property HeaderBand : TQRBand
Selects a band to be printed as this groups header band.

Property Level : Integer
This group's level. The outer (highest) level is 0, the inner (lowest) level is 9.

Property OnNeedData : procedure (var Value : String) of object

TQRLabel Component

The QRLabel component is used to display text on a report. The caption property can be changed in any events during report preparation. If you want a to output text from a data field use the QRDBText component instead.

property Alignment : (taRight, taLeft, taCenter)
Sets the alignment of the text within the given rectangle. Se also AlignToBand

property Caption
Caption contains the actual text to be printed.

property AlignToBand
If AlignToBand is False (default) the text will be printed and aligned within the rectangle given by

the Left, Top, Width and Height properties. If AlignToBand is True, the text will be horizontally aligned to the entire page. I.e. if you set AlignToBand:=True and Alignment:=taCenter the text will be centred on the page.

property Color

The background color of the text. See also the Transparent property.

property Font

If you want to use another font than the default for the form or QRBand set the Font property.

property OnPrint procedure : (sender : TObject; var Value : String)

The OnPrint event is called each time the QRLabel is about to be printed. The caption of the label is sent in the Value parameter. You can change this parameter without affecting the value of the Caption, only what is printed for this instance.

property Transparent

If Transparent is False (default) the background of the text will be painted before the text is drawn. If Transparent is True, the background will not be painted.

In addition the QRLabel component has all the properties from the QRCustomControl component.

TQRMemo Component

property Alignment

property Color

property Font

property Lines

property WordWrap

TQRPreview component

QRPreview is used to create your own preview forms. This component takes all the hassle out of displaying the actual preview. To call your own preview form instead of the standard form supplied with QuickReport assign an event handler to the QRPrinter.OnPreview event. In this event handler you bring up your own form which must contain a QRPreview component. You can then select what page to view using the PageNumber property.

TQRPreview properties and methods:

procedure ZoomToWidth

Sets the zoom property to value so the page width just fits on the preview component.

procedure ZoomToFit

Sets the zoom property so the entire page is shown on.

property PageNumber : Integer

Setting this property causes the QRPrinter.PageNumber property to be set and the preview to be repainted with the new page.

property Zoom : Integer

Sets the zoom factor for the preview in percent.

TQRPrinter component

TQRPrinter is a low level printer object with preview, save and load capabilities. An instance of this object called QRPrinter is automatically created when your application starts.

QRPrinter is like a Printer object with recording capabilities. All print commands you send to its Canvas property are recorded in a Windows Metafile object. If you use the NewPage method the

current page is kept in memory and QRPrinter creates a new one. When you have created the entire printout you can choose to preview, print or save the result. QuickReport uses this object for all its printing.

Example on how to use QRPrinter:

```
if QRPrinter.Status=mpReady then with QRPrinter do begin
  BeginDoc;
  With QRPrinter.Canvas do begin
    Font.Color:=clBlue;
    Font.Size:=30;
    TextOut(10,10,'QuickReport is Great');
  end;
  EndDoc;
  Preview;
  Cleanup;
end;
```

QRPrinter has the following methods and properties

`procedure BeginDoc`

Start a new print job. This sets the Status property to mpBusy.

`procedure Cancel`

Cancel the current print job. This sets the Status property to mbReady

`procedure Cleanup`

After you are finished with the current print job this command will free all pages from memory. This sets the Status property to mpReady.

`procedure CopyPageToClipboard`

Copy the currently selected page to the clipboard. This requires that EndDoc has been called.

`procedure EndDoc`

Tells QRPrinter that you are finished sending commands to the canvas. This sets the Status property to mpFinished.

`procedure Load(Filename : String)`

Loads a previously saved printout from a file.

`procedure NewPage`

Tells QRPrinter to start printing a new page. The current page is kept in memory and a new one is created. Page number is increased by one. This requires the Status property to be mpBusy.

`procedure Preview`

This brings up print preview. It requires the Status property to be mbFinished. If you have assigned an event handler to the OnPreview event this gets called and nothing else is done.

`procedure Print`

This prints the current report. It requires the Status property to be mpFinished.

`procedure Save(Filename : String)`

This saves the current report to a file. The Status property must be mpFinished. The file format is a QuickReport internal format. Documentation for this will be released later.

`property Canceled : boolean`

True if the Cancel method has been called. This can be used to check if users pressed a Cancel

button.

property Canvas : TCanvas

This is the canvas you use to do your printing. It understands all standard Tcanvas commands except those that returns information, like TextWidth.

property EnableOpenBtn : boolean

If True, the Open file button will appear on the standard preview form.

property EnablePrintBtn : boolean

If True, the Print button will appear on the standard preview form.

property EnableSaveBtn : boolean

If True, the Save file button will appear on the standard preview form.

property ToPage : integer

Specify the last page to be printed.

property FromPage : integer

Specify the first page to be printed.

property OnPreview : procedure of object

Use this event if you want to display your own preview form instead of the standard included with QuickReport. Creating a custom preview is very easy using the supplied TQRPreview component. See the demo program for an example on how to do this.

property Orientation : TPrinterOrientation

Set to poPortrait or poLandscape to select printer orientation.

property Page : TMetafile read only

This is a metafile representation of the currently selected page. Status must be mpFinished for this metafile to be valid.

property PageHeight : integer read only

The current page height in pixels.

property PageCount : Integer read only

Total number of pages sent to the QRPrinter in the current print job.

property PageNumber : integer

Select the page you want QRPrinter to make available for viewing or copying to clipboard. The Status property must be mpFinished.

property PageWidth : integer read only

The current page width in pixels.

property ShowProgress : boolean

If true a progress form will appear indicating the progress of printing, saving and loading. The form has a Cancel button and the Cancelled property will be set to true if the user presses this button.

property Status : TQRPrinterStatus read only

Indicates the current QRPrinter status. It can be one of the following:

- * mpReady QRPrinter is ready to accept a new print job
- * mpBusy A print job is currently being recorded
- * mpFinished The print job has been ended

- * mpPrinting QRPrinter is sending a job to the printer
- * mpPreviewing A preview is currently active

property Title : String

The current print job title. This title will appear in the print manager and in any network print queue.

property Thumbs : integer

QRPrinter can print several scaled down images of pages on each sheet of paper. If you set Thumbs to two QRPrinter will print two by two images, if you set it to three, three by three images will be printed and so on.

QRShape Component

QRShape is used to display simple lines and figures on the report.

property Brush

The brush to use when painting the shape.

property Pen : Tpen

The pen to use.

Property Shape : (qrsRectangle, qrsCircle, qrsVertLine, qrsHorLine)

The shape property determines what type of figure to draw.

In addition the QRShape component has the properties from the QRCustomControl component.

QRSysData Component

The QRSysData component displays various system information. In addition to these properties QRSysData has all the components that QRLabel has, except the Caption property.

property Data : TQRSysDataType

The Data property decides what information the component will display.

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

property Text : String

The Text property is printed before the actual system data.

QuickReport component

procedure NewPage

This method can be called in any event during report creation to force QuickReport to perform a page break.

procedure Prepare

Call the prepare method if you want to create the report without automatically printing or previewing it. Prepare creates the report and sends it to the QRPrinter object.

procedure Preview

The Preview method first prepares the report if necessary, then calls the QRPrinter.Preview method. If you haven't assigned an event handler for the QRPrinter.OnPreview event the standard preview form is displayed. Otherwise your OnPreview event handler is called. The report is finally freed from memory before returning.

procedure Print

The Print method first prepares the report if necessary, sends it to the QRPrinter object, and then calls the QRPrinter.Print methods. The Report is finally freed from memory.

property PageCount : Longint read only

The total number of pages in the report. This property is not valid before the report is finished.

property PageHeight : Longint read only

The current page height, taken from the QRPrinter object.

property PageNumber : Longint read only

The current page number. This property can be access in events during report preparation.

property PageWidth : Longint read only

The current page width, taken from the QRPrinter object.

property RecordCount : Longint read only

The total number of records in the master dataset. Se also the SQLCompatible property.

property RecordNo : longint read only

The current record in the dataset. If you are printing a master/detail report using the QRDetailLink component this property indicates the record number in the master table.

property AfterDetail : TNotifyEvent

The AfterDetail event is called after the detail bands for one record are printed.

property AfterPrint : TNotifyEvent

The AfterPrint event is called when the report is prepared and sent to the QRPrinter object.

property BeforeDetail : TNotifyEvent

The BeforeDetail event is called before the detail bands for a record is printed.

property BeforePrint : TNotifyEvent

The BeforePrint event is called before the report is prepared.

property Columns : integer

The Columns property can be used to print multi column mailing labels or phone directory style reports. Use the PrintOrder property to set way records will print. Also se the QRBand.BandType=rbColumnHeader.

property DataSource : TDataSource

Set the DataSource property to the datasource you print the records from. QuickReport automatically advances through this dataset and prints a set of detail bands for each record. If you don't spesify a DataSource and not a OnNeedData event handler only one set of detail bands will be printed. Also see the OnNeedData event and RestartData property.

property DisplayPrintDialog : Boolean

If DisplayPrintDialog is set to True a print dialogue will appear before the report is printed. This dialogue will let the user select printer, pages to print and print to file.

property LeftMarginInches : Longint
The left margin in 1/10th inches

property LeftMarginMM : Longint
The left margin in millimetres

property OnEndPage : TNotifyEvent
The OnEndPage event is called whenever a page brake is about to be performed.

property OnNeedData : procedure (var MoreData : Boolean) of object
The OnNeedData event is used if you want to print a report of data not coming from a regular DataSource, but from arrays, variables, text files or whatever you like. If you leave the DataSource property blank and instead write an OnNeedData event handler this handler gets called whenever QuickReport is going to print a new set of detail bands. Set the MoreData parameter to false when you are printing the last set of data.

property OnStartPage : TNotifyEvent
The OnStartPage event is called whenever a new page is started.

property Orientation : TPrinterOrientation
Selects the paper orientation. Orientation can be set to poPortrait and poLandscape. Please note that the TPrinterOrientation type is declared in the Printer unit. If you want to set this property you need to include the printer unit in your uses clause.

property PrintOrder : (ColByCol, RowByRow)
PrintOrder is used to set how the text will flow on a multi column report. Text can be printed column by column or row by row.
Note - RowByRow will be ignored in version 0.97 of QuickReport

property ReportTitle : String
The ReportTitle will appear in the Print Manager and in any network printing queue. It will also printed wherever you have used a TQRSysData component set to display ReportTitle.

property ReportType : (qrStandard, qrMasterDetail)
Set ReportType to qrMasterDetail if you want QuickReport to print the detail tables you have linked using the QRDetailLink component. Please see the chapter on Master/detail reports for more information on creating such reports.

property RestartData : Boolean
If True (default) QuickReport will move to the first record in the dataset before printing starts. If False, printing will start at the current record.

property ShowProgress : boolean
If True (Default) a progress form will appear during report preparation and printing. The progress form has a Cancel button.

property SQLCompatible : boolean
Not all Databases support the RecordCount function. If you get an error message saying 'function not supported' try setting the SQLCompatible to True.

property TitleBeforeHeader : boolean
If False (Default) the page header will be printed before the Title band on the first page. You can reverse this by setting the TitleBeforeHeader property to True.

PART V Bugs

This is a list of the bugs I know about

- * TQRShape will not always be updated in the form designer when changing properties for the pen or brush
- * No clipping is done on shapes and images

Please report any bugs to allan@qsd.no. Remember to tell which version of QuickReport you are using.

PART VI Version History

Version 0.97a

- * Design time preview now only prepares the first five pages
- * Progress form now appears during preparation of design time preview
- * Added about form to the component menu
- * Bug fix, ForceNewPage now works also if a GroupFooter is printed alone on a page
- * Bug fix, the OnFilter event now works properly also for TQuickReport, not only TQRDetailLink
- * Bug fix, QRDBCalc is no more including the first record from the next group when used with TQRGroup
- * Bug fix, Group footer bands are now correctly printed at the end of the report

Version 0.97

- * New property, TQRDetailLink.Master, used to create *any* type of master/detail report. Please see the enhanced master/detail example report for an example on how to use this
- * New property, TQRDetailLink.PrintBefore, if True the sub detail table will be printed before the Detail band of the master table. Default is False.
- * New properties, TQuickReport.ColumnMarginInches and ColumnMarginMM
- * New property, ParentFont for TQRLabel, TQRDBCalc, TQRSysData, TQRMemo
- * New files included: all strings are now moved to the QRCONST.RES resource file. The identifiers are declared in the QRCONST unit, both the compiled file and source file is included for this unit. This should make it possible for QuickReport to work with any other components that use resource files
- * Added first and last page buttons to the standard preview form. First and previous button disabled when displaying first page, next and last button disabled when displaying last page
- * Zoom combobox on the standard preview form replaced by three buttons - fit in window, 100% and fit in width
- * New property, TQRPrinter.ZoomButtons, can be set to default to get the old Zoom combobox back
- * Added functionality to the TQRDBCalc component, Min, Max and average
- * Property changed, TQRDBCalc.ResetGroup changed to ResetBand. QRDBCalc are now designed to be reset after a specific band is printed, not when a group breaks. This is for compatibility with the TQRDetailLink component
- * Property removed, TQuickReport.ReportType. This is automatically detected now and the property is no longer needed
- * The TQuickReport popup menu (right clicking) now includes Preview and About
- * Removed qrPageCount as an option for the TQRSysData. This will be added again later
- * New method, TQRDBCalc.Start, causes a QRDBCalc component to be reset. Can be called from any method during report creation
- * Bug fix, the QRBand.ForceNewPage property no longer creates an extra blank page
- * Bug fix, right alignment no longer clips the rightmost character
- * Bug fix, Transparent property now works with right alignment
- * Bug fix, TQRBDText.Autosize is now stored correctly
- * Bug fix, QuickReport now works when no printer is installed. New property QRPrinter.PrinterOK will show if a printer is installed. Print button on standard preview is disabled if QRPrinter is false.
- * Bug fix, the load report dialog box now show all saved QuickReport (*.QRP) files again.
- * Bug fix, the save report dialog box correctly adds the .QRP extension as default. Overwrite warning turned on

- * Bug fix, deleting a band connected to a TQRDetailLink component no longer causes an Access Violation
- * Bug fix, using a QRDetailLink without a DataSource no longer causes an error
- * Bug fix, using QRDetailLink now never prints the last record twice
- * Bug fix, deleting a component linked to a QRDetailLink no longer causes an error
- * Bug fix, using the OnFilter event now works even if no records mach. This also works on any detail level now.
- * Bug fix, TQRShape now repaints itself correctly
- * Several enhancements to the demo reports, primarily the master/detail report which is now includes a master and two detail tables

Version 0.96

- * TQRCustomPreview component replaced with TQRPreview. This component has build in scrollbars so the ScrollUp and ScrollLeft properties are gone. Scrolling is now smooth and flicker free. Preview is also somewhat faster.
 - * A shadow is painted behind the preview page to give it a nicer apperance.
 - * Preview now shows actual paper size, including non printable areas
 - * The report form is minimized during preparation of design time preview (double clicking the QuickReport component). This effectivly removes all flashing during the process and gives a major speed advantage
 - * New events, TQuickReport.OnFilter and TQRDetailLink.OnFilter, use for easy filtering of report data
 - * New method, TQuickReport.NewPage, can be called in any event during report creation to force a page break
 - * New icons for all QR components on the component palette
 - * Pressing ALT+E now exits the standard preview form
 - * Switching page in preview now automatically resets scroll position to top and left
 - * All strings moved to QRPREV.RES. Edit this file if you want to change or nationalize text output
 - * Added Zoom, Zoom to fit and Zoom to Width controls to the demo preview form
 - * Bug fix, QRMemo is now positioned correct horizontally too!
 - * Bug fix, fixed paint method for empty QRDBText components when switching DataSource from not active to active
 - * Bug fix, printing barcode fonts now works if AutoSize is False
 - * Bug fix, *all* QuickReport controls now check for deletion of *any* linked components
 - * Bug fix, column header band are now printed correctly on every page
 - * Bug fix, left margin is now printet correctly on every page
 - * Bug fix, swiching pages now works even after printing from preview form
 - * Bug fix, the buttons on the standard preview screen are now displayed right in 8, 15 and 16 bit video modes
 - * Bug fix, value of QRPrinter.Status changed from mpFinnished to mpFinished
 - * The help file is now really included! No b-keywords yet, though.
 - * QuickReport registration is now available via SWREG at Compuserve. SWREG# 8644
 - * Technical support is now available at Compuserve, user id 101372,3471
- 32 bit version:
- * The progress form now uses a TProgressBar instead of TGuage
 - * Scrollbar thumb tracking in the preview
 - * Demo reports changed to use the new data module feature.

Version 0.95c

- * New component, TQRDetailLink, use to create master detail reports with unlimited number of detail tables
- * New property, TQuickReport.ReportType. Use to select standard report or master/detail
- * New property, TQuickReport.Columns, sets the number of columns to print per page
- * New property, TQuickReport.PrintOrder, Set to ColByCol or RowByRow to control how multi column reports are printed
- * New property, TQuickReport.TitleBeforeHeader, set to true if you want the report title to be

printed before the page header on first page

- * New band type, rbColumnHeader, for the obvious (?) purpose
- * New band type, rbOverlay, printed as an overlay on each page
- * New report included in the demo program to show how to create multi column mailing labels
- * Removed property TQRDBText.PrintMask, use the datafields DisplayMask property to format data fields
- * Bug fix, QRMemo is now positioned at the correct vertical position
- * Bug fix, printing with the OnNeedData event now prints the last set of detail bands correctly
- * The report title is now displayed on the progress form during preparing and printing. The progress form border is changed to bsDialog
- * Preparing reports is now somewhat faster, mostly with reports with many components on them
- * QRDBText components connected to empty data fields will now display the name of the component in the form designer
- * Changed the dotted frame around QRbands in the designer to a dotted line below each band. This gives a cleaner appearance.
- * Changed palette page name from 'Report' to 'QuickReport'. *To install this version first uninstall any previous version, rebuild your library, remove the Report page from the palette. Then reinstall QuickReport. THIS IS VERY IMPORTANT!* Sorry, I just had to do this at some stage...
- * The source code is now fully Delphi 32 compatible
- * Totally rewritten component reference and updated most other parts of the documentation. All properties are now documented. The skeleton of a help is also included.
- * QUICKREP.RES is changed to QRPREV.RES

Version 0.95b

- Bug fix, first record in the dataset is again printed (bug in 0.95a)

Version 0.95a

- * Added Exit button to the standard preview form.
- * Pressing Escape in the standard preview form now closes preview
- * The progress form now has a Cancel button which will cancel preparing, printing, saving and loading of reports
- * New method, QRPrinter.Cancel, will cancel preparing, printing saving and loading of reports
- * Bug fix, data bound fields now contains the correct record in a group footer, not the first record in the next group
- * Bug fix, printing a range of pages no longer prints a blank page at the end
- * Bug fix, TQRPrinter.ShowProgress now works properly
- * TQuickReport has got its ShowProgress property back. This automatically overrides the value set in TQRPrinter
- * Bug fix, QRPrinter.EnableSaveBtn now works correctly
- * Bug fix, Setting the QRBand.ForceNewPage combined with group header no longer screws up the first page

Version 0.95

- * New component TQRPrinter, a Printer object with many new features. This component does not appear on the component palette, one instance is created at application startup
- * New component TQRMemo for printing multiline labels.
- * Design time preview of reports, double click the QuickReport component to bring up a preview. You can even print and save the report from this preview
- * Print dialog now supports print range, selected and print to file
- * New and hopefully nicer progress form with progress bar
- * There are now only three files required for QuickReport: QUICKREP.DCU, QUICKREP.DCR and QUICKREP.RES
- * PREVIEW files removed, these files are no longer needed
- * Setting a ReportTitle now causes the title to appear in print manager and on networks
- * Demo program totally rewritten, two new reports and a simple custom preview form
- * New preview form with save and load buttons

- * Removed event TQuickReport.OnPreview. Use TQRPrinter.OnPreview instead
- * Removed property TQuickReport.ShowProgress. Use TQRPrinter.ShowProgress instead
- * Support for printing of images from tables using the TDBImage component
- * QRDBCalc now prints the correct value again
- * Bug fix, clipping of last character is now definitely history
- * Default frame width for bands is now 0
- * The report template is now updated and should not give any error messages when loaded
- * QuickReport is now reentrant in the way that you can now call the print method directly from the preview form
- * Using QuickReport on a network now works correctly, TESTFILE is never created
- * More accurate and complete documentation. Still a lot to do here, though...

Version 0.94

- * Support for unlimited number of detail bands, title and summary bands and page headers and footers. This involves several changes to existing reports
- * New property, TQRBand.BandType
- * Properties removed, TQuickReport.TitleBand, PageHeaderBand, PageFooterBand, SummaryBand. If you open a report created with a previous version you will get error messages saying 'property does not exist'. Just ignore the error message. *Remember to set the BandType property for all bands*
- * Removed the undocumented property TQuickReport.SecondDetailBand since it is no longer needed
- * New events, TQuickReport.BeforeDetail and TQuickReport.AfterDetail
- * New property, TQuickReport.SQLCompatible for compatibility with SQL servers that don't support the RecordCount functionality
- * New property Visible for all printable properties
- * New property TQRBand.Frame for printing frames around bands
- * Added a horizontal scrollbar to the report designer so you can create really wide reports
- * Bug fix, QuickReport no longer creates an empty file 'TESTFILE'
- * Bug fix, TQRDBText.PrintMask property now works properly for fields of type float
- * Bug fix, initial page header, last page footer and summary bands are now printed when using the OnNeedData event for report printing
- * Bug fix, setting AutoSize to True should no longer cause clipping of the rightmost character
- * Source code for the standard preview is now included in the distribution files

Version 0.93c

- * Harddisk space is now used to generate the report if there is not enough memory. This limits the report size to 32767 pages or available disk space. This version does not swap TImages to disk.
- * New event TQuickReport.OnNeedData for printing reports from arrays, lists etc.
- * New properties, TQuickReport.LeftMarginInches and TQuickReport.TLeftMarginMM
- * New property TQuickReport.RestartData, set to false if you don't want the report to automatically go to the first record in the dataset
- * A vertical scrollbar now automatically appears in the form designer when a TQuickReport component is created. This allows for creation of very large report bands
- * AutoSize now defaults to false for TQRDBText, TQRSysData and TQRDBCalc
- * Bug fix, TQRSysData now displays date and time according to local settings and the extra colon after hours<10 is now gone
- * Bug fix, any group break will now force all lower level groups to break
- * Bug fix, more trouble with vertical positioning of memo fields fixed
- * Bug fix, background color for all text components are now printed correctly
- * Master/detail demo report changed from using a TSQL to two TTables with a master/detail join
- * Thanks again, Jon;-)

Version 0.93b

- * New property, TQRBand.Ruler, displays horizontal and/or vertical rulers in inches or cm
- * New property, TQRDBCalc.AsReal, returns current calculated value of float fields

- * New event, OnPrint event for TQRLabel, TQRDBText, TQRDBCalc, TQRSysData for user formatting or any other purpose. Changes to caption is not carried over to the next record
- * New shapes for TQRShape added, qrsHorLine and qrsVertLine to draw vertical and horizontal lines
- * Bug fix, landscape printing now works properly
- * Bug fix, removed GPF at program shutdown if an image was printed
- * Created simple report template for the form gallery
- * Demo program changed to create one of the reports dynamically
- * Some major code cleaning
- * More error trapping put into code
- * Corrected a lot of spelling errors in the documentation. Many thanks to Jon Duerdoth for pointing them all out to me!

Version 0.93a

- * QuickReport can now handle much larger reports. Up to 32276 pages with 32276 components on each page. Only limit now is memory. This will be addressed in a later version. Tested this with a 1300 page report.
- * Preview of large reports is much faster.
- * TQRCustomPreview component is updated
- * Simple splash screen to show report progress added.
- * New property TQuickReport.ShowProgress, to enable and disable splash screen
- * New properties, TQRLabel.AlignToBand and TQRSysData.AlignToBand, use to align components on the page rather than inside its own box
- * TQuickReport.OnPreview event has new format. *Changes needed to any existing OnPreview event handler*
- * Bug fix, clipping is now done around all text components
- * Bug fix, all components will now show on the preview, also the last on the report.
- * Bug fix, center and right alignment now places components at exact right position
- * Bug fix, TImage will now print even is stretch property is false. No clipping, though
- * Bug fix, memo field stretching should now work properly
- * Added registration information to documentation
- * Information on how to create your own Preview forms added to documentation

Version 0.93

- * New preview system with many features
- * New component TQRCustomPreview for use in your own custom preview forms
- * New format of QRBand.BeforePrint and AfterPrint events to allow not printing selected records. *Changes are needed to any existing BeforePrint and AfterPrint event handler declarations!*
- * Metafiles and Icons are now supported via TImage
- * Fixed bug in QRGroup and QRDBCalc - deleting the DataSource component will now be picked up by all the appropriate components
- * New property, TQRDBCalc.PrintMask
- * Bug fix, right and center alignment now also works with QRLabel
- * Bug fix, if an assigned summary band is deleted, TQuickReport is now updated correctly
- * Removed Customer and Orders table from the archive. Master/detail demo now uses the customer and orders tables in the DBDEMOS alias
- * Major structure changes to the code

Version 0.92a

- * Documentation reformatted in ms-write. WordPad was not Write compatible after all! I just love Microsoft...

Version 0.92

- * New component TQRDBCalc
- * New component TQRSysData
- * New properties for TQRGroup : DataField, DataSource and GroupData

- * New property TQRBand.ForceNewPage
- * Bug fix - taRight and taCenter alignment now works properly
- * Bug fix - Memo fields will now print at the correct vertical position
- * Demo program updated to use new components. Colors added to demo report.
- * Master - detail report added to demo program, two tables added for this report
- * Documentation rewritten and updated in MS WordPad to make it more readable

Version 0.91d

* Fixed bug causing report to crash if an empty band was printed. More bugs in the demo program fixed.

Version 0.91c

* Reference to unneeded unit Editu.pas removed from Demo program. Fixed problem with scrollbars not always displayed in preview.

Version 0.91b

* New clipping routine implemented.

Version 0.91a

* Disabled clipping around bands as this caused printing trouble on some systems. This is just a temporary hack until I find out how to do this the right way.

Version 0.91

* New Component TQRShape, for printing rectangles and circles. Support for TShape removed.

* Added memo field support via QRDBText. Bands will stretch to make room for the entire memo field.

* Minor updates to demo program.

Version 0.90, July 1,

* First beta release

PART VII Getting technical support

Questions on using QuickReport will be answered on both Internet and CompuServe. Please use the following addresses to submit questions and comments:

Internet: allan@qsd.no
CompuServe: 101372,3471

Please note that the amount of mail coming in is sometimes quite high. Registered users will get priority with their questions. I try to answer all incoming mail though.

PART VIII Registration

All beta versions (pre 1.0) of QuickReport can be used freely in your programs, provided you make no changes to it.

When version 1.0 is ready QuickReport will become Shareware. The price is set to US\$65. Registered will get access to download the full source code from Internet and remove the 'Unregistered QuickReport' message on the progress form.

If you want to register QuickReport now you will get free upgrade to any 1.x versions of QuickReport.

SWREG

QuickReport registration is available at CompuServe SWREG. Type GO SWREG at any

CompuServe prompt. The SWREG number for QuickReport is 8644. No additional registration form is necessary when registering via SWREG.
Please note that the SWREG price for QuickReport is \$75, due to a CompuServe fee.

Credit Card

Credit card registration is currently not available, but you will shortly be able to pay the registration fee with VISA or MasterCard.

Check or cash

Payment can be made with a check for US\$ 65 + \$10 handling fee. This additional fee might seem strange but I have to pay US\$ 15 to cash a check in foreign currency.

An EuroCheque for NOK400 is OK.

Please fill out the registration form below and send together with check/cash to the following following address:

Allan Lochert
Uranienborgvn. 9a
0351 Oslo
Norway

Money transfer/SWIFT

Your bank can arrange for a regular money transfer or a SWIFT transfer. Not all banks know about SWIFT, though. Please transfer to the following account and send in a registration form by mail or fax at +47 22 60 49 72

Allan Lochert
Uranienborgvn. 9a
0351 Oslo
Norway

Account no: 6030 05 47055
Bank: Christiania Bank, Stortorget 7, 0107 Oslo, Norway

--- cut --- cut --- cut --- cut --- cut --- cut

QuickReport version 0.97 Registration Form

Date

Name/contact

Postal address

zip/city

Country

E-mail address

Send source code by postal mail instead of e-mail ? _____ (+ \$5 postage/handling)

Signature

SEND TO:

Allan Lochert
Uranienborgvn. 9a
0351 Oslo
Norway