

# Breaking the Mould

by Lut Mentz

This article is about report creation and breaking away from the conventional master-detail structure to make a report that looks more like a document than a listing. The example also shows how to access multiple datasets in a report.

## It's in the detail ...

Banded report generators like QuickReports make it easy to create 'standard' master-detail reports. The price is that the code always lists the data sequentially. Detail information lies below the master information, giving the appearance of a computer listing, however much it is embellished. Getting away from this stricture involves a small amount of coding which is now described in full.

## The Report

The report we will create will produce a panel for each master record, in which master and detail information and summary statistics will sit side-by-side. Figure 1 shows the top of the first page.

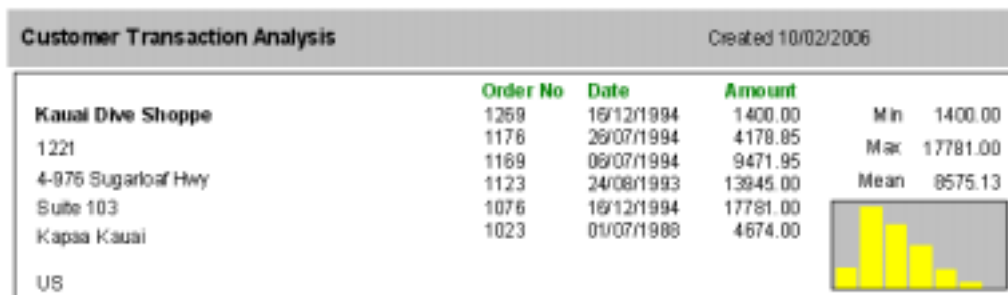


Figure 1 - The top of the first page, showing the panel structure.

The data comes from the BDEDEMOS database, tables Customer and Orders.

## The Design

We start with a three-band structure, page header, detail band and page footer. The header and footer are unremarkable, the detail band carries the controls to display master and detail information. The report is driven by a QRAbsTable component. We use this because it allows the possibility of getting our master and detail data from non-standard sources. In this example, the data is from BDE table components, but it could be acquired in any way whatever in principle.

The design is shown in Figure 2. Customer fields will be placed in QRLabel2-6 and 15. QRMemo1, 2 and 3 will list the order information. The three labels appearing as QRLabel8 show the statistics and the QRImage control below them will show the trend graph. Three datasets, the abstract table and two BDE Tqueries are used. The TQuery components will access the customer and orders tables. The abstract table is set as the report dataset.

The abstract table and the single QRDBText control are linked and actually drive the report. The abstract table has been given a single field, 'Customer' of type ftString. This is done in the IDE by editing the FieldDefs property of the table. The dataset is then assigned to the DBText as usual and its field set to the only possibility, 'Customer'.

When the report is executed, the SetFieldValue event is called whenever the dataset requires data for one of its fields. This will happen once per record in our report because we have one DBText control (actually it happens twice but we'll get to that). As you may have guessed, most of the action will take place in the GetFieldData event, which is the subject of the next section.

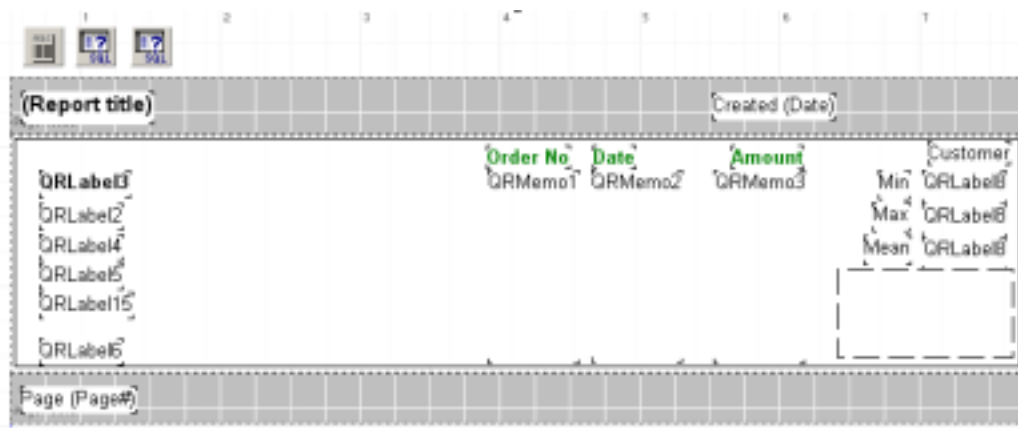


Figure 2 - The report design in the Delphi 7 IDE.

## The Code

This report is designed with a fixed detail band height and we would like the detail band not to break across pages. This is done by setting the `KeepOnOnePage` property of the detail band to 'True'. Now there is a slight complication because the data values are required to estimate the band height before it is printed. This causes the `SetFieldData` event to be called first by the height checking code, then again when the band is printed. The matter is dealt with by code which checks if the call is odd or even numbered. It sounds tricky but the logic is fine and the code works exactly as expected. The first lines of the event code are listed below.

---

```

procedure TMDReport.QRabsTable1SetFieldValue(const DataStr: String;
  Field: TField; buffer: Pointer; var MoreData: Boolean);
var
  dataval : string;
  amount : double;
begin
  // give data to the buffer - it is not shown by design so we pass nothing
  if Field.FieldName = 'Customer' then
    strcpy( pchar(buffer), ' ');
  // this test is required if we switch on 'KeepOnOnePage'
  if Detailband1.KeepOnOnePage and (( dataCallnum mod 2 ) = 0) then
    begin
      inc(dataCallnum );
      exit;
    end;
  // load customer information

```

---

The code above is executed on every call to the event. The next part is called only on the even numbered calls.

---

```

  // load customer information
  qrlabel3.Caption:= custQuery.fieldbyname( 'Company').asString;
  qrlabel2.Caption:= custQuery.fieldbyname( 'CustNo').asString;
  qrlabel4.Caption := custQuery.fieldbyname( 'Addr1').asString;
  qrlabel5.Caption := custQuery.fieldbyname( 'Addr2').asString;
  qrlabel15.Caption := custQuery.fieldbyname( 'City').asString;
  qrlabel6.Caption := custQuery.fieldbyname( 'Country').asString;

```

---

Here, the values from the Customer table are loaded into the display controls.

## Handling Sub-detail data

This is standard code. We set up a query that gets the data we need, then loop through the dataset loading the QRMemo controls. Also in the loop we collect the statistical data, and save the amount values in an array for later plotting.

---

```

// load orders memo
saleQuery.SQL.Text := 'select * from orders where custno = ''' + dataVal + ''' order by
    OrderNo desc;';
saleQuery.Open;
saleQuery.first;
qrmemo1.Lines.Clear;
qrmemo2.Lines.Clear;
qrmemo3.Lines.Clear;
recCount := 0;
totAmount := 0.0;
maxAmount := 0.0;
minAmount := 200000.0;
while not saleQuery.Eof and (recCount < 8) do
begin
    qrmemo1.Lines.Add( saleQuery.fieldbyname('OrderNo').AsString);
    qrmemo2.Lines.Add( saleQuery.fieldbyname('saleDate').AsString);
    amount := saleQuery.fieldbyname('itemsTotal').Value;
    qrmemo3.Lines.Add( format('%10.2f',[amount]));
    if amount > maxAmount then maxAmount := amount;
    if amount < minAmount then minAmount := amount;
    totAmount := totAmount + amount;
    saleQuery.next;
    amounts[recCount] := amount;
    inc(recCount);
end;

saleQuery.Close;

```

---

The value of 8 for the number of displayed records was got by trial and error.

## Winding up

It remains only to display the statistical information, advance the Customer dataset and check for end of data.

---

```

// Statistics
if recCount = 0 then minAmount := 0.0;
qrlabel8.Caption := format('%10.2f',[minAmount]);
qrlabel10.Caption := format('%10.2f',[maxAmount]);
if recCount > 0 then
    totAmount := totAmount / recCount;
qrlabel12.Caption := format('%10.2f',[totAmount]);
PlotAmounts;
// Advance master table and set MoreData field.
custquery.Next;
inc(dataCallnum);
moredata := not custquery.Eof;

end;

```

---

That's it except for some initialisation that is done in the BeforeScroll event of the abstract table.

---

```

procedure TMDReport.QRAbsTable1BeforeOpen(DataSet: TDataSet);
begin
    // Open master dataset
    CustQuery.Open;
    // We need to count the calls to GetFieldData
    dataCallnum := 0;
    // Make a bitmap to draw on
    qrimagel.Picture.Bitmap := TBitmap.create;
    qrimagel.Picture.Bitmap.Width := qrimagel.Width;
    qrimagel.Picture.Bitmap.Height := qrimagel.Height;

end;

```

---

## Remarks and extensions

Our simple example shows the power of the method but does not exploit it fully. As already mentioned, the data could have been obtained in any way allowable in Delphi. Also there could be more than one sub-detail table opened and listed.

It would also be straightforward to display any number of 'Master' tables in one detail band. It should also be pointed out that this method will work without an abstract table. The master data in this case could be a BDE dataset and the code we put in the GetFieldValue event can be modified and put in the BeforePrint event of the detail band. The possibilities, while not endless, are numerous.

The source code and text of this article can be got at this URL [www.qusoft.com/Article.zip](http://www.qusoft.com/Article.zip).

---

*Lut is a member of the technical support team at QBS Software.*

---

**Problems with your project?**

**Need an extra pair of eyes to get another view on things?**

**Brian Long** offers troubleshooting and training services on .NET, C#, Delphi, Delphi for .NET and C++Builder in the UK, Europe and overseas.

If you need some help call Brian Long today on: +44 (0)7770 663053, email [brian@blong.com](mailto:brian@blong.com), or visit [www.blong.com](http://www.blong.com)