Charts From Tables: Connecting ChartFX To TQuery

by Rashid Ali

A very important aspect of all database applications is communicating information to the user. Textual information is often sufficient, but when a comparative judgment is to be made it is much better to present the information graphically. There is a lot of wisdom in the saying 'every picture tells a story'.

The March issue of *The Delphi Magazine* reviewed two charting Tools for Delphi: Graphics Server and Light Lib Business. These offer lots of functionality, but of course they cost money!

The aim of this article is to show how to use the ChartFX graphing VBX control, included free with Delphi 1, to present information retrieved from database tables pictorially. Although the principles could doubtless be applied to the ChartFX OCX which is included with Delphi 2, I've not yet had time to test this out.

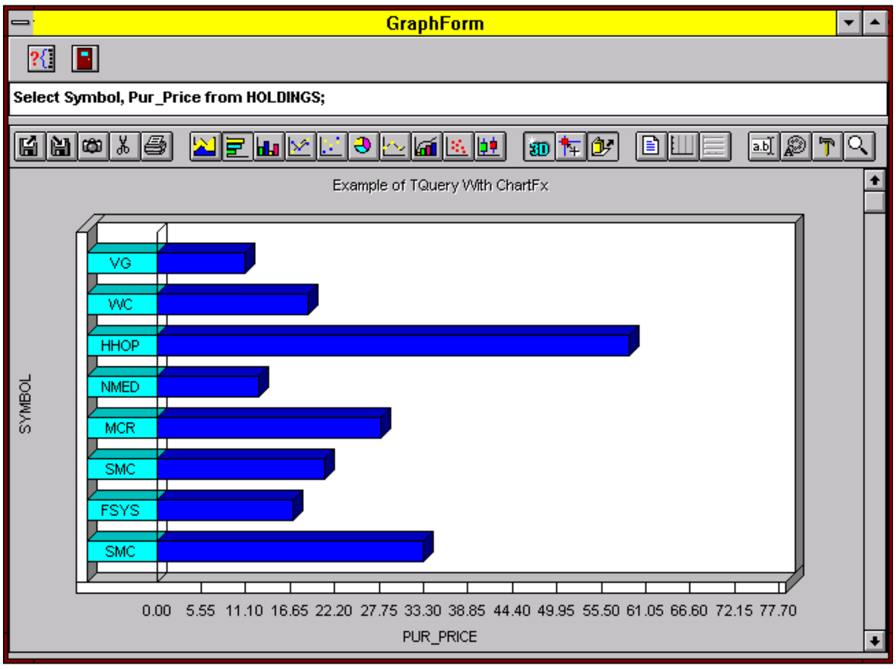
The simplest way to use this control is to place the ChartFX component on a form and set the various properties. When your program runs, default data is then displayed, which is easily altered using the data editor. The presentation quality of this VBX control is very good, considering it comes free with Delphi!

Connecting To TQuery

To graph data from a database table using ChartFX, we first of all need to run a query on the table and then transfer the query results to ChartFX.

The screenshot above shows the example application included on this month's disk. A TMemo is used to enter an SQL query (a default query is included in the code). The HOLDINGS table is in DBDEMOS

➤ Listing 1



Select Symbol, Pur_Price from HOLDINGS;

➤ *The sample application: full source code is on this month's disk*

```
procedure TGraphForm.SQLExexBtnClick(Sender: TObject);
var
  NumRecs : Integer; { data counter }
  NumCols : Integer; { number of columns in the data }
  ColCntr : Integer; { index for the for loop }
begin
  DataSource1.DataSet := Query1;
  Query1.DatabaseName := 'DBDEMOS';
  Query1.Close;                  { close any pending queries }
  Query1.SQL.Clear;              { clear previous SQL query text }
  Query1.SQL.Add(SQLEdit.Text);  { get the SQL query }
  Query1.Active := True;         { run the query }
  NumRecs := Query1.RecordCount; { count number of records retrieved }
  NumCols := Query1.FieldCount;  { number of fields in the DataSet }
  { fill data, assuming MultiColumns }
  for ColCntr := 0 to NumCols-2 do begin
    { initialise ChartFx with number of columns and number of data points }
    ChartFX1.OpenData[COD_VALUES] := makelong(NumCols-1,NumRecs);
    { tell ChartFx to expect data for the current series }
    ChartFX1.ThisSerie := ColCntr;
    NumRecs := 0;
    Query1.First;
    repeat
      { fill labels }
      ChartFX1.Legend[NumRecs] := Query1.Fields[0].AsString;
      { fill data columns }
      ChartFX1.Value[NumRecs] := Query1.Fields[ColCntr+1].AsFloat;
      NumRecs := NumRecs+1;
      Query1.Next;
    until Query1.EOF;
    { Graph titles }
    ChartFX1.Title[CHART_LEFTTIT] := Query1.Fields[1].FieldName;
    ChartFX1.Title[CHART_BOTTOMTIT] := Query1.Fields[0].FieldName;
    ChartFX1.Title[CHART_RIGHTTIT] := 'Example of TQuery With ChartFx';
    { close data channel for current series }
    ChartFX1.CloseData[0] := ColCntr;
  end;
  Query1.Active := False; { close the query }
end;
```

and contains stock market purchase prices (field PUR_PRICE) for various companies, identified by their stock market symbols (field SYMBOL). TQuery and TDataSource components make a connection to the table.

When the program runs, it first of all displays a graph based on ChartFX's built-in default data. The top left button on the form, when clicked, retrieves the SQL query from the memo and executes it, displaying the graph resulting from the query.

Listing 1 shows the OnClick event handler for the button, which contains all the important code. Note that internally ChartFX will number graph series starting from 0, which must be taken into account when setting up the loop for passing the data to it. The rest of the code should be self-explanatory.

For a new graph, just enter an appropriate SQL query in the memo field and click the button to execute.

Rashid Ali runs RALI Consultants, specialising in network solutions, statistical process and quality control. He is currently busy converting lots of old DOS applications to Delphi but in his spare time likes to create masterpieces from wood instead: cabinets, windows and even conservatories! He can be contaced by email as qra@ralicon.demon.co.uk