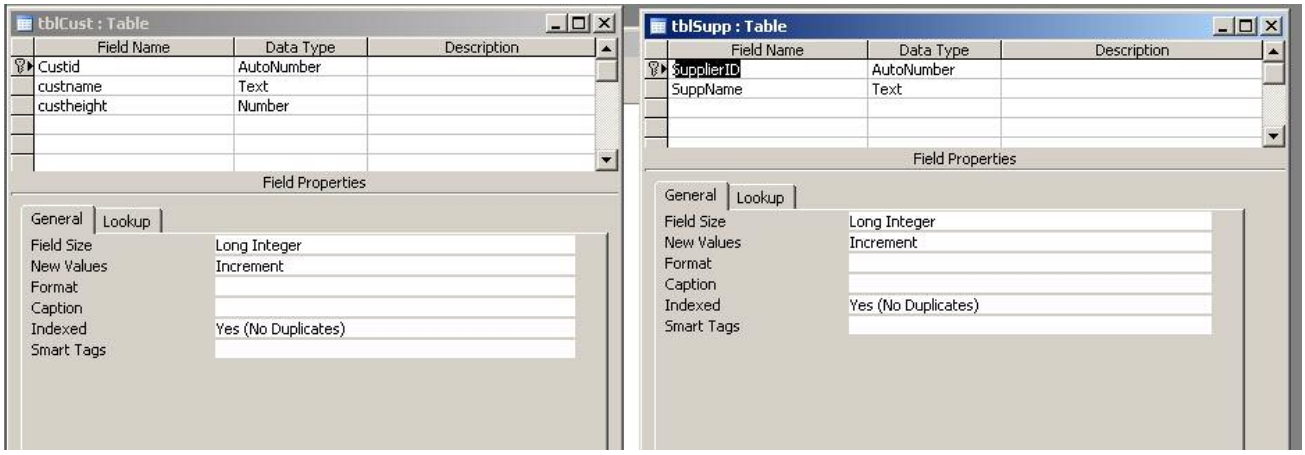
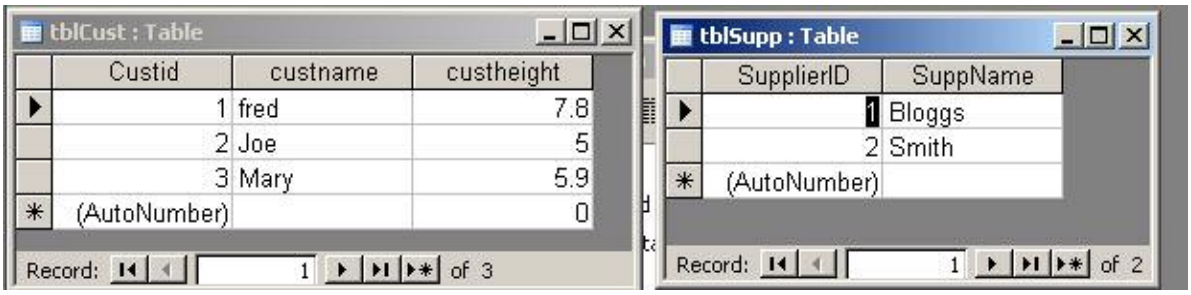
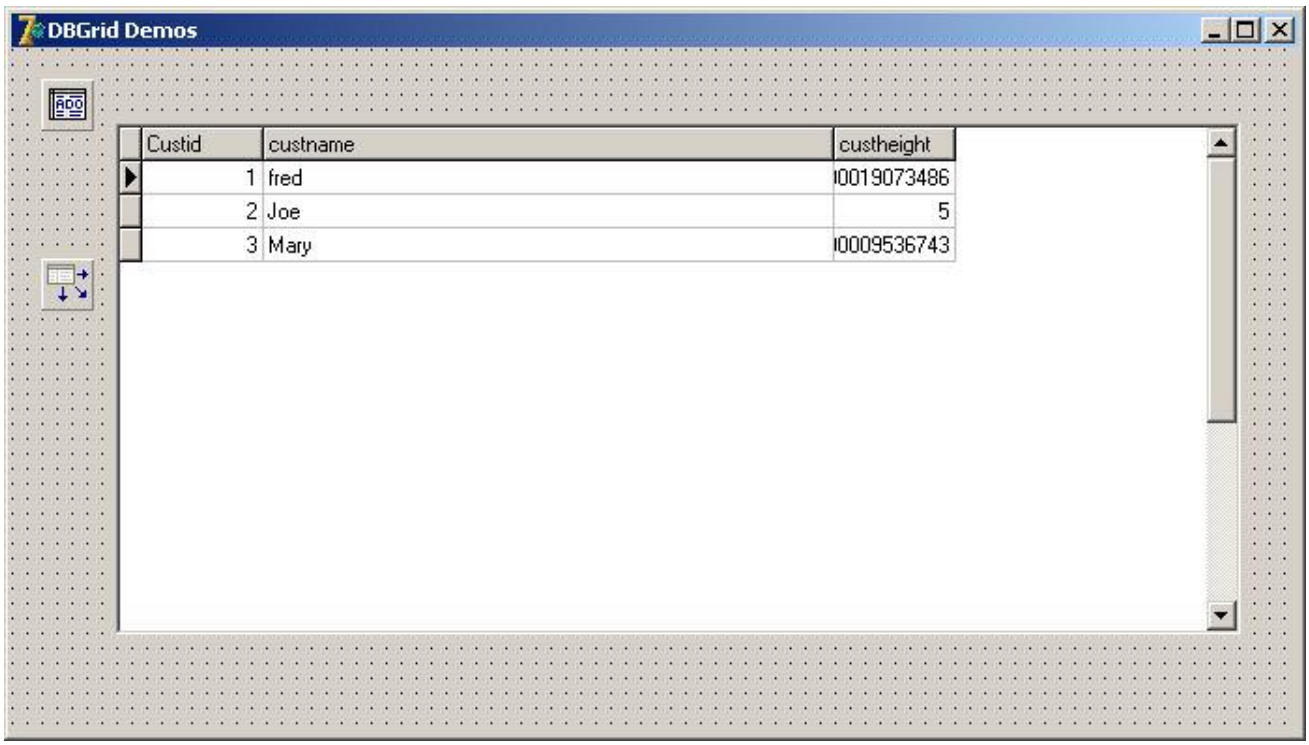
Using DBGrids : A Miscellany

These examples use an Access database which has two tables as shown below:





Create a Delphi Application and save it. On Form1, add an ADOTable (ConnectionString built at design time to connect to the database shown; make the TableName property tblCust to start with. Add a DataSource (set its DataSet to ADOTable1) and a DBGrid (set its DataSource to DataSource1). If ADOTable1 is made Active, then at design time it should look like:



Note that the values in the Custheight column are not rendered correctly: the original values entered into the Access table of 7.8 and 5.9 are rendered respectively as 7.80000019073486 and 5.90000009536743.

Copying from a DBGrid into another application such as MS Word

When producing these notes, I tried to copy these values (above) from the grid into this document by selecting the grid cell, pressing CTRL+C then clicking this document (MS Word) and pressing CTRL+V, the conventional method of copying between applications. It didn't work. Further experiments showed that you can copy in this way from Edit boxes (and related controls such as Memo boxes, but not from DBGrids. One workround this problem is to sense a CTRL+C keypress in the DBGrid, using an OnKeyDown event, then write the field value which is displayed there into an Edit box (which need not be visible), then select the Edit box's Text and use the Edit box's CopyToClipboard method. This puts the required text onto the clipboard, and moving to another application such as Notepad or Word allows you to paste the text conventionally. Assuming you've added an Edit box (here called `ed` with its Visible set to false), the code for DBGrid's OnKeyDown event could be:

```
procedure TForm1.DBGrid1KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if (key=ord('C')) and (Shift=[ssCtrl]) then
   begin
    ed.Text:=DBGrid1.SelectedField.AsString;
    ed.SelectAll;
    ed.CopyToClipboard;
   end;
end;
```

Notes:

1. The key pressed, as returned by the reference parameter `Key` will be given by the ASCII code of the (uppercase) key pressed, i.e. `ord('C')`. Whether ALT, or SHIFT or CTRL are also pressed is given by the **shiftstate** of the key, which is a set of control keys, e.g. `[ssCtrl]` as in this example; `[ssCtrl,ssAlt]` would indicate both CTRL and ALT keys pressed.
2. Clicking anywhere on a row of a DBGrid at runtime causes the underlying dataset's current record to be changed to that displayed on that particular row. The SelectedField property of the DBGrid returns the value of the field of that record corresponding to the selected cell, i.e. the value displayed in the selected cell. The AsString qualifier ensures that the value (which is a TField type) is converted to a string type so that itb can be treated as text.
3. The CopyToClipboard method copies only selected text, hence the need to select all of the text.

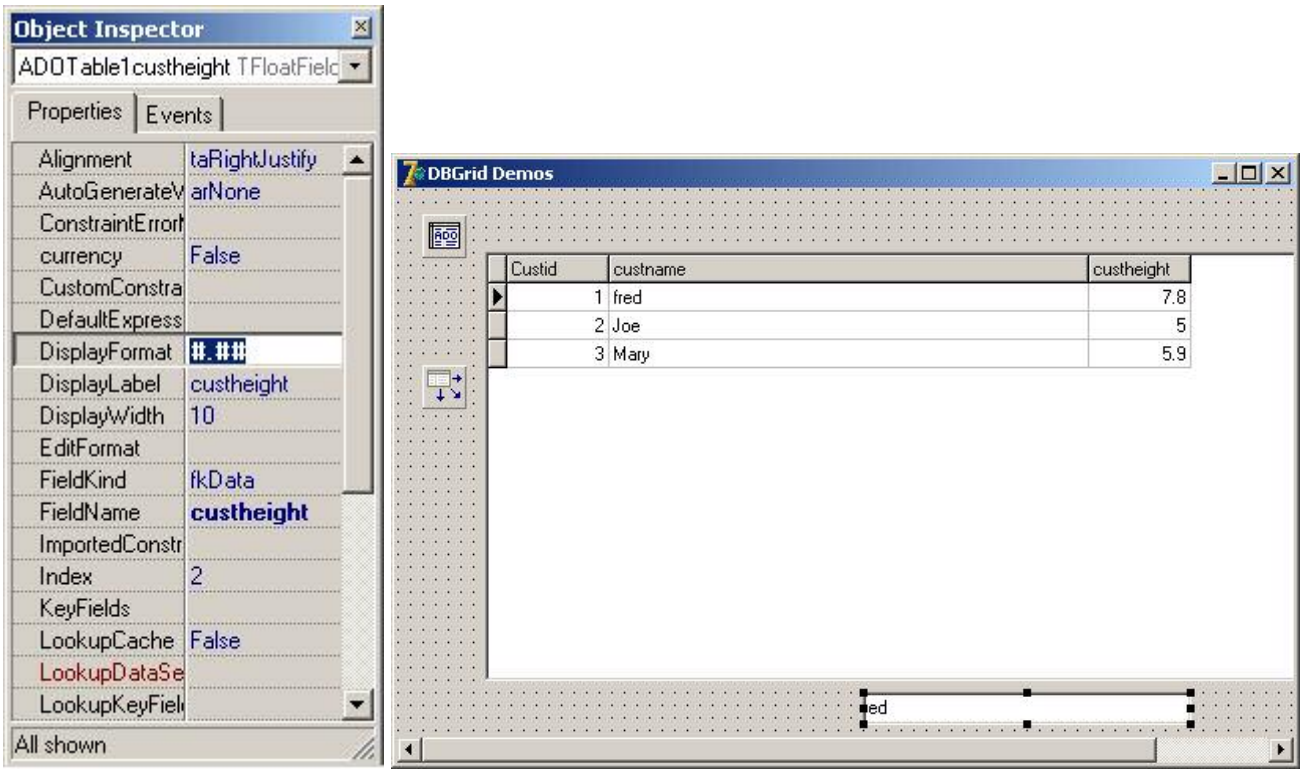
This is quite clumsy, but it works.

Curing the display formatting problem

The problem arises because the fields concerned are stored not as the decimal digits originally entered, but as floating point numbers. Displaying them requires conversion back from the binary floating point format into a meaningful decimal display. Delphi will do this to a high degree of precision, which incorporates rounding errors, unless you tell it to do otherwise. The display format is a property of the DBGrid's dataset, not the DBGrid itself.

Select this dataset, i.e. ADOTable1 (in design mode), and (if in Delphi 7), the Object TreeView window will show under tblCust (ADOTable1) a list of fields. Expand this list if necessary, and

select the offending field custheight. The fields are themselves represented in Delphi as objects – Tfield objects, and the selected field object will appear detailed in the Object Inspector. Select the DisplayFormat property, by default blank, and enter #.## which means display all digits before the decimal point, but rounded to a maximum of 2 digits after the decimal point. You can find further information about displayformat strings in the Delphi online help.

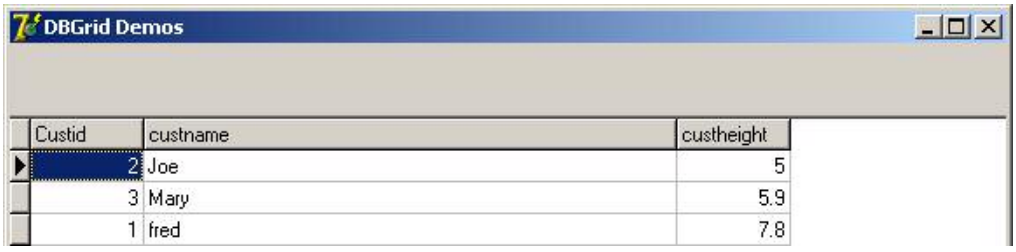


Sorting the records as displayed in a DBGrid using OnTitleClick

A DBGrid contains a useful event called OnTitleClick which occurs when you click on the title row of the DBGrid at runtime. It returns the parameter `column` which identifies the grid column (i.e. field) which was clicked on. Now an ADOdataset such as an ADOTable has a Sort property, which is a string specifying on what field the dataset is to be sorted. The simple OnTitleClick event handler below allows easy sorting of the displayed records:

```
procedure TForm1.DBGrid1TitleClick(Column: TColumn);
begin
  ADOTable1.Sort:=Column.FieldName;
end;
```

Clicking on the custheight title cell causes sorting of the records as shown on the right.



You can specify whether sorting is ascending (default as above) or descending by adding a qualifier to the string, e.g.:
`ADOTable1.Sort:=Column.FieldName + ' DESC';` Note the space before DESC.