TurboPower Orpheus in a Unicode World

OvcTCComboBox

The Table CellComboBox internally uses a record with the name TCellComboBoxInfo for storing its data. This record contained a variant record with either a ShortString or a Stringlist combined with ShortString. We cannot store Unicode data in a ShortString, so those ShortStrings had to be changed to strings. But since we cannot use strings in a variant record, we had to change it into a normal record. Here's the old and the new version of the TCellComboBoxInfo record:

Old:

```
  TCellComboBoxInfo = packed record
    Index : integer;                {index into Items list}
    case integer of
      0 : (St       : ShortString);  {string value if Index = -1}
      1 : (RTItems  : TStrings;      {run-time items list}
           RTSt     : ShortString);  {run-time string value if Index = -1}
  end;
```

New:

```
  TCellComboBoxInfo = packed record
    Index : integer;     {index into Items list}
    St      : string;    {string value if Index = -1}
    RTItems : TStrings;  {run-time items list}
    RTSt    : string;    {run-time string value if Index = -1}
  end;
```

Inside the function OvcTableGetCellData you're supposed to return a pointer to a TCellComboBoxInfo record:

```
type
  TMyRecord = record
    mrComboBox3     : TCellComboBoxInfo;
  end;

procedure TForm1.OvcTable1GetCellData(Sender: TObject; RowNum: Longint;
  ColNum: Integer; var Data: Pointer; Purpose: TOvcCellDataPurpose);
begin
  Data := nil;
  if (0 < RowNum) and (RowNum <= 199) then
    case ColToFieldMap[ColNum] of
      9 : Data := @MyDB^[RowNum].mrComboBox3;
    end;
end;
```

Code that fails with the above changes

The example EXTBL02.dpr that ships with Orpheus was using a pretty interesting approach for returning this data structure. Let us take a quick look at a simplified version of the record that stores the row of a table and the GetCellData method:

```
type
  TMyRecord = record // this stores a row of the table
    // [...]
    mrSimple        : string;
    mrPicture       : TOvcDate;
    mrComboBox3Int  : integer;
    mrComboBox3Items: TStringList;
  end;

procedure TForm1.OvcTable1GetCellData(Sender: TObject; RowNum: Longint;
  ColNum: Integer; var Data: Pointer; Purpose: TOvcCellDataPurpose);
begin
  Data := nil;
  if (0 < RowNum) and (RowNum <= 199) then                      {!!.01}
    case ColToFieldMap[ColNum] of
      9 : Data := @MyDB^[RowNum].mrComboBox3Int;
    end;
end;
```

As you can see, TMyRecord does not use the TCellComboBoxInfo record. TurboPower simply put an index and a TStringList into that record. This used to work fine, as long as the memory layout of TMyRecord (starting at mrComboBox3Int) and TCellComboBoxInfo remains identical. But there is no guarantee about that. To be able to use Unicode in Orpheus, the layout of the TCellComboBoxInfo record had to be changed as described above. With that change, the example fails without a compiler error at runtime.

What we need to do is change TMyRecord, so that it uses the TCellComboBoxInfo record. Then we return a pointer to the TCellComboBoxInfo record.

```
type
  TMyRecord = record
    // [...]
    mrSimple        : string;
    mrPicture       : TOvcDate;
    mrComboBox3     : TCellComboBoxInfo;
  end;

procedure TForm1.OvcTable1GetCellData(Sender: TObject; RowNum: Longint;
  ColNum: Integer; var Data: Pointer; Purpose: TOvcCellDataPurpose);
begin
  Data := nil;
  if (0 < RowNum) and (RowNum <= 199) then                      {!!.01}
    case ColToFieldMap[ColNum] of
      9 : Data := @MyDB^[RowNum].mrComboBox3;
    end;
end;
```