Tips & Tricks

DataList

DataList is a very useful program that is included with Delphi 1 (but not Delphi 2, though it compiles fine in Delphi 2) in the \DELPHI\DEMOS\DB\DATALIST directory. It analyzes database structures, showing a list of aliases, with a list of tables for each alias, plus fields and indexes for each table. The problem for Paradox tables is that it does not show the primary index. To set a database to a primary index all you have to do is type `Table1.IndexName := ''`. So, in the file MAIN.PAS, look for the line `IndexListbox.Clear;` (in 2 places) and after it add

```
IndexListBox.Items.Add('');
```

The index name is blank but the index fields will show.

Contributed by Glenn Shukster, CompuServe 72734,123

Wave Audio Supported?

I received an email recently, following up my *Audio-Enabled Components* article in Issue 9, asking if I knew a way to check if a machine supported playing Wave Audio data. I checked the MMSYSTEM.PAS unit from the VCL library and found a function which does the trick.

The function `waveOutGetNumDevs: Word;` returns the number of wave output devices installed. A simple but fairly safe way to check if you can play a .WAV file is to see if `waveOutGetNumDevs` returns a non-zero result. If it does you can play the file. If you are using `TWave` from Issue 9 you can change the `Play` method of `TWavePlayer`, `TImageSnd` and `TSndBitBtn` to create 'safe' code for using the `TWave` class.

Contributed by Paul Warren, HomeGrown Software Development, British Columbia, Canada (email hg_soft@uniserve.com)

Project History List

Normally Delphi lists the four most recently opened projects in its history list in the `File` menu. You can actually increase this by editing DELPHI.INI.

Just add the number of entries you want at the end of the file under `[Closed Files]`, making sure they are sequentially numbered. The actual text does not matter as it will be over-written by Delphi. The following example gives you 11 projects (the current one and 10 others):

```
[Closed Files]
File_0=C:\DELPHI\VL2\SRC\VETLINK.DPR,1,49,1,74
File_1=C:\NEW\DOCDEMO.DPR,1,1,1,1
File_2=C:\NEW\PROJECT1.DPR,1,1,1,1
File_3=C:\DELFI\HL2\SRC\HAIRLINK.DPR,1,1,19,40
File_4=C:\DELFI\VL2\SRC\VL2_DBUG.DPR,1,1,8,31
File_5=C:\DELFI\HL2\SRC\HL2_DBUG.DPR,1,1,1,1
File_6=C:\DELFI\VL2\TST\GRAPHS.DPR,1,1,1,1
File_7=C:\DELFI\VL2\TST\GRAPHS.DPR,1,1,1,1
File_8=C:\DELFI\VL2\TST\GRAPHS.DPR,1,1,1,1
File_9=C:\DELFI\VL2\TST\GRAPHS.DPR,1,1,1,1
```

Contributed by Rohit Gupta (email rohit@ww.co.nz)

Setting The Cursor

I am running a simulation model. While it is running, I want to change the cursor to `crHourGlass` everywhere except over a couple of buttons (`Pause` and `Stop`). When the simulation is finished, I turn the cursor back to default everywhere. The relevant code (copied from my child application) is shown in Listing 1. The key line is marked; without this, it may work, provided you don't have things like `TPopupMenu`, which is a `TComponent` descendant. Since `TComponent` doesn't have a cursor, if there is a `TComponent` descendant on the form, the program will give a run-time error without the key line.

Contributed by Dr Ian Johnson, IMJ Consultants (email: greenhat@northnet.com.au)

Auto-Increment Off Track

Paradox's auto-increment fields are great, but what can you do if somehow the auto-increment counter gets off track? In this situation, when you try to add a record, the BDE will give you a key violation error, as its internal counter is *less* than the last record! The answer is to restructure the table using the Database Desktop (Delphi 1) or Database Explorer (Delphi 2). If you change the auto-increment field to a *numeric* type, you will then be able to set the Valcheck lower limit *above* your highest record. When you then convert the field type *back* to auto-increment, the value will stick.

Contributed by Mike Orriss, CompuServe 100570,121

➤ *Listing 1*

```
procedure TMDIChild.SetHGcursor(cursor2hg : boolean);
var i : integer;
begin
  for i := 0 to ComponentCount - 1 do
    if ( Components[i] is TControl ) then  {Key Line!!}
      if cursor2hg then begin
        (Components[i] as TControl).cursor :=
          crHourglass;
        BtnPause.cursor := crDefault;
        BtnStop.cursor := crDefault;
        cursor := crHourglass;   {sets cursor for form}
      end else begin
        (Components[i] as TControl).cursor :=
          crDefault;
        cursor := crDefault;     {sets cursor for form}
      end;
end;
```

Custom DBGrid Display

Sometimes it can be useful to display something other than the actual field contents in a DBGrid cell. To do this, use the GetText and SetText event handlers for TField. Listing 2 shows a simple example for a field called ID which actually contains A, B and C but is *displayed* (and updated) by all the controls as 1, 2 and 3.

Contributed by Mike Orriss, CompuServe 100570,121

Adding Indexes

There is a quirk you need to be aware of when adding indexes to your tables. For example, if you try to use:

```
InvTbl.AddIndex('cusname', 'name', []);
```

you will get an error *'Invalid Index/Tag name. Index:cusname'*. The rule is, if the index name is the same as the field name, you need to specify ixCaseSensitive (which is the default setting), but if the index name is *not* the same as the field name you must use ixCaseInsensitive. So, for the example above, the code is:

```
InvTbl.AddIndex('cusname', 'name',
  [ixCaseInsensitive]);
```

or

```
InvTbl.AddIndex('name', 'name', []);
```

Contributed by Mike Orriss, CompuServe 100570,121

Masking TDBedit

To apply a mask to a TDBedit which will capitalise the first letter of each word typed, add code to the OnKeyPress event handler:

```
procedure TForm1.DBEdit1KeyPress(Sender: TObject;
  var Key: Char);
begin
  with Sender as TDBEdit do
    if (Text = '') or (Text[SelStart] = ' ')
    or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key := UpCase(Key);
end;
```

Contributed by Mike Orriss, CompuServe 100570,121

No Combo Duplicates

When adding items into a Combobox you often want to ensure that the resulting list does not contain any duplicate items.

To achieve this, use this code fragment when adding the items:

```
if Items.IndexOf(Text) < 0 then
  Items.Add(Text);
```

Contributed by Mike Orriss, CompuServe 100570,121

```
procedure TForm1.Table1IDGetText(Sender: TField;
  var Text: OpenString; DisplayText: Boolean);
var s: string;
begin
  s := (Sender as TStringField).Value;
  case s[1] of
    'A' : Text := '1';
    'B' : Text := '2';
    'C' : Text := '3';
  else
    Text := '9'
  end;
end;
procedure TForm1.Table1IDSetText(Sender: TField; const
Text: String);
var s: string;
begin
  case Text[1] of
    '1' : s := 'A';
    '2' : s := 'B';
    '3' : s := 'C';
  else
    s := 'Z'
  end;
  (Sender as TStringField).Value := s;
end;
```

➤ *Listing 2*

```
procedure TBag.SetFormPlace(AName: string;
  AForm: TForm);
var s: string[99];
    Place : TWindowPlacement;
begin
  Place.length :=SizeOf(TWindowPlacement);
  if not GetWindowPlacement(AForm.Handle,@Place) then
    exit;
  with Place do begin
    s := IntToStr(Flags);
    s := AppendS(s,ShowCmd);
    s := AppendS(s,ptMinPosition.X);
    s := AppendS(s,ptMinPosition.Y);
    s := AppendS(s,ptMaxPosition.X);
    s := AppendS(s,ptMaxPosition.Y);
    s := AppendS(s,rcNormalPosition.Left);
    s := AppendS(s,rcNormalPosition.Top);
    s := AppendS(s,rcNormalPosition.Right);
    s := AppendS(s,rcNormalPosition.Bottom);
  end;
  SetString(AName,s);
end;
```

➤ *Listing 3*

Form Coordinates

The GetPlacement API function can be handily used to get the coordinates a form would have *if* it were restored, but still *leaving it in the maximized state*. The extract from my TBag component showing in Listing 3 illustrates the use.

Contributed by Mike Orriss, CompuServe 100570,121

Delphi 2 DeleteFile

If you call DeleteFile(fname) with a variable of type String you may get a type mismatch error, saying the function expects a variable of type PChar. To fix this, specify the call as SysUtils.DeleteFile to ensure you are calling the Delphi routine, not the Windows API one. Similar problems occur with FindClose.

Contributed by Mike Orriss, CompuServe 100570,121