Dave's Development Blog

Software Development using Borland / Codegear / Embarcadero RAD Studio

Chapter 3: A simple custom menu (AutoSave) Fixed

By David | September 20, 2009 0 Comment

In the last chapter I said that the mechanism that I had might not work property with DLLs because if you are looking for an existing menu, then it might not have been created at the time you wizard is loading. So here I'm going to show you how to fix that.

First we need to add a new method to the class which will take the menu creation code out of the classes Create method:

```
TBlogOTAExampleWizard = Class(TInterfacedObject, IOTAWizard)
Private
  FTimer       : TTimer;
  FCounter     : Integer;
  FAutoSaveInt : Integer;
  FPrompt      : Boolean;
  FMenuItem    : TMenuItem;
  FINIFileName : String;
  FSucceeded: Boolean; // New
  Procedure SaveModifiedFiles;
  Procedure LoadSettings;
  Procedure SaveSettings;
  Procedure InstallMenu; // New
  Procedure TimerEvent(Sender : TObject);
  Procedure MenuClick(Sender : TObject);
Protected
  procedure Execute;
  function  GetIDString: string;
  function  GetName: string;
  function  GetState: TWizardState;
  procedure AfterSave;
  procedure BeforeSave;
  procedure Destroyed;
  procedure Modified;
Public
  Constructor Create;
  Destructor Destroy; Override;
End;
```

Next we need to remove the menu code from the Create method (cut it to the clipboard as you will need it in a minute):

```
constructor TBlogOTAExampleWizard.Create;

var
  iSize : DWORD;

begin
  FMenuItem := Nil;
  FCounter := 0;
  FAutoSaveInt := 300; // Default 300 seconds (5 minutes)
  FPrompt := True;     // Default to True
  // Create INI file same as add module + '.INI'
  SetLength(FINIFileName, MAX_PATH);
  iSize := MAX_PATH;
  iSize := GetModuleFileName(hInstance, PChar(FINIFileName), iSize);
  SetLength(FINIFileName, iSize);
  FINIFileName := ChangeFileExt(FINIFileName, '.INI');
  LoadSettings;
  FSucceeded := False;
  FTimer := TTimer.Create(Nil);
  FTimer.Interval := 1000; // 1 second
  FTimer.OnTimer := TimerEvent;
  FTimer.Enabled := True;
end;
```

Next we need to add the new method InstallMenu. You will note it has changed slightly from before as I wanted the “Auto Save…” menu to appear below the “Window List…” menu item.

```
procedure TBlogOTAExampleWizard.InstallMenu;

Var
  NTAS: INTAServices;
  mmiViewMenu: TMenuItem;
  mmiWindowList: TMenuItem;

begin
  NTAS := (BorlandIDEServices As INTAServices);
  If (NTAS <> Nil) And (NTAS.MainMenu <> Nil) Then
    Begin
      mmiViewMenu := NTAS.MainMenu.Items.Find('View');
      If mmiViewMenu <> Nil Then
        Begin
          mmiWindowList := mmiViewMenu.Find('Window List...');
          If mmiWindowList <> Nil Then
            Begin
              FMenuItem := TMenuItem.Create(mmiViewMenu);
              FMenuItem.Caption := '&Auto Save Options...';
              FMenuItem.OnClick := MenuClick;
              FMenuItem.ShortCut := TextToShortCut('Ctrl+Shift+Alt+A');
              mmiViewMenu.Insert(mmiWindowList.MenuIndex + 1, FMenuItem);
              FSucceeded := True;
            End;
        End;
    End;
end;
```

You should also note from above that I've added a shortcut to the menu item.

You should have noticed by now that there is a new field called FSuccess in the class which is initialised in the constructor to False. This is to indicate that our new menu has not been installed yet. Next we need to update the TimeEvent method to reference this new field and install the menu as follows:

```
procedure TBlogOTAExampleWizard.TimerEvent(Sender: TObject);
begin
  Inc(FCounter);
  If FCounter >= FAutoSaveInt Then
    Begin
      FCounter := 0;
      SaveModifiedFiles;
    End;
  If Not FSucceeded Then
    InstallMenu;
end;
```

That's it for the changes. Hopefully that's straight forward. Here are the files.

Category: Open Tools API Tags: Borland, CodeGear, Delphi, Experts, INTAServices, IOTAEditBufferIterator, IOTAEditorServices, IOTAWizard

Iconic One Theme | Powered by Wordpress