

# Dave's Development Blog

Software Development using Borland / Codegear / Embarcadero RAD Studio

## Chapter 9: Aboutbox Plugins and Splash Screens

By David | August 25, 2011

0 Comment

This time around I'm going to talk about the ability of Delphi 2005 (and above) to display information about your wizard/expert on the splash screen and also in the About dialogue box.

First of all we need some global private module variables to hold information that is common between the about box plugin and the splash screen. The reason for them outside a method is they need to be available to the **Initialization** section of the module. These variables are as follows:

### Implementation

...

### Var

```
{IFDEF D2005}
VersionInfo      : TVersionInfo;
bmSplashScreen   : HBITMAP;
iAboutPluginIndex : Integer = 0;
{$ENDIF}
```

...

In the above variable declaration there is a user-defined type **TVersionInfo** which has the following definition:

### Type

```
TVersionInfo = Record
  iMajor   : Integer;
  iMinor   : Integer;
  iBugfix  : Integer;
  iBuild   : Integer;
End;
```

This reason for this is not that it is a requirement of the Open Tools API but simply because I want to display the version number of the expert in the about dialogue and on the splash screen.

Next we need some constants and resource strings for displaying the information on the splash screen and in the about dialogue as follows:

```
{$IFDEF D2005}
Const
    strRevision : String = ' abcdefghijklmnopqrstuvwxyz';

ResourceString
    strSplashScreenName = 'My Expert Title %d.%d%s for Embarcadero RAD Studio ####';
    strSplashScreenBuild = 'Freeware by Author (Build %d.%d.%d.%d)';
{$ENDIF}
```

Now we can look at the creation of About box information in Wizard Initialization function. Firstly, the code loads a bitmap from the experts (DLL/BPL) resource file for display in the about box and on the splash screen. This bitmap should be 24 x 24 in size with the lower left hand side depicting the transparent colour. I've found that a white background works best if you want a single image to work on all the different splash screens. You can create this bitmap and add it to the RES file directly using the Borland Image Editor (or similar) or use a resource file to create the RES file which should then be included.

The resource file looks like this:

```
SplashScreenBitMap BITMAP "Images\SplashScreenIcon.bmp"
```

and should be named with the .RC extension. This can then be included in the main wizard / expert module as follows:

```
{$R '..\SplashScreenIcon.res' '..\SplashScreenIcon.RC'}
```

This bitmap is then passed to the [AddPluginInfo](#) method to create the about box plugin entry. You don't have to be as complicated as I've made it below but the below shows you how to display the experts version number. Note that this method expects that the [VersionInfo](#) record is already populated with version information. This happens in the Initialization section of the module along with the splash screen.

```
Function InitialisWizard : TWizardTemplate;
```

```

Var
    Svcs : IOTAServices;

Begin
    Svcs := BorlandIDEServices As IOTAServices;
    ToolsAPI.BorlandIDEServices := BorlandIDEServices;
    Application.Handle := Svcs.GetParentHandle;
    {$IFDEF D2005}
    // Aboutbox plugin
    bmSplashScreen := LoadBitmap(hInstance, 'SplashScreenBitMap');
    With VersionInfo Do
        iAboutPluginIndex := (BorlandIDEServices As IOTAAboutBoxServices).AddPluginInfo(
            Format(strSplashScreenName, [iMajor, iMinor, Copy(strRevision, iBugFix + 1, 1)]),
            '$WIZARDDDESCRIPTION$',
            bmSplashScreen,
            False,
            Format(strSplashScreenBuild, [iMajor, iMinor, iBugfix, iBuild]),
            Format('SKU Build %d.%d.%d.%d', [iMajor, iMinor, iBugfix, iBuild]));
    {$ENDIF}
    ...
End;

```

Next we will look at the splash screen information. This is somewhat different to the rest of the Open Tools API because the `BorlandIDEServices` variable at the point in time the modules are loaded is not necessarily set and available. To facilitate this, Borland / Codegear added a specific service interface for the splash screen called `SplashScreenServices`. The below code is in the `Initialization` section of the module so that it occurs as the DLL/BPL is being loaded and not when (later) the wizard is initialised. The code below gets the build information for the expert, then gets a reference to the resource image as above and passes all this information to the `AddPluginBitmap` method of the `IOTASplashScreenServices`.

```

...
Initialization
    {$IFDEF D2005}
    BuildNumber(VersionInfo);
    // Add Splash Screen
    bmSplashScreen := LoadBitmap(hInstance, 'SplashScreenBitMap');
    With VersionInfo Do
        (SplashScreenServices As IOTASplashScreenServices).AddPluginBitmap(
            Format(strSplashScreenName, [iMajor, iMinor, Copy(strRevision, iBugFix + 1, 1)]),
            bmSplashScreen,
            False,
            Format(strSplashScreenBuild, [iMajor, iMinor, iBugfix, iBuild]));
    {$ENDIF}
Finalization
    {$IFDEF D2005}
    // Remove Aboutbox Plugin Interface
    If iAboutPluginIndex > 0 Then
        (BorlandIDEServices As IOTAAboutBoxServices).RemovePluginInfo(iAboutPluginIndex);
    {$ENDIF}

```

End.

Obviously the missing piece is how to get the version information from the DLL/BPL. Below is the code to do this:

```

Procedure BuildNumber(Var VersionInfo: TVersionInfo);

Var
    VerInfoSize: DWORD;
    VerInfo: Pointer;
    VerValueSize: DWORD;
    VerValue: PVSFixedFileInfo;
    Dummy: DWORD;
    strBuffer: Array [0 .. MAX_PATH] Of Char;

Begin
    GetModuleFileName(hInstance, strBuffer, MAX_PATH);
    VerInfoSize := GetFileVersionInfoSize(strBuffer, Dummy);
    If VerInfoSize <> 0 Then
        Begin
            GetMem(VerInfo, VerInfoSize);
            Try
                GetFileVersionInfo(strBuffer, 0, VerInfoSize, VerInfo);
                VerQueryValue(VerInfo, '\\', Pointer(VerValue), VerValueSize);
                With VerValue^ Do
                    Begin
                        VersionInfo.iMajor := dwFileVersionMS Shr 16;
                        VersionInfo.iMinor := dwFileVersionMS And $FFFF;
                        VersionInfo.iBugfix := dwFileVersionLS Shr 16;
                        VersionInfo.iBuild := dwFileVersionLS And $FFFF;
                    End;
                Finally
                    FreeMem(VerInfo, VerInfoSize);
                End;
            End;
        End;
    End;

```

I hope that was all straight forward. From this point onwards I will be putting all the code that I've written about on the Open Tools API into a single project and can be compiled by the following versions of Delphi:

- Delphi 5
- Delphi 7
- Delphi 2006
- Delphi 2009
- Delphi 2010

as these are the only version I have, however I'm trying to ensure that the code will be correctly conditionally compiled to work with all other version of Delphi above Delphi 5. This code can be used as the

basis of your own expert and has been built modularly so that the discrete topics are in the main in their own module so you can pick and choose the bits you want.

#### [OTA Chapter 09 Files](#)

Dave.

Category: Open Tools API Tags: Borland , BorlandIDEServices , CodeGear , Delphi , Embarcadero , Experts , IOTAAboutBoxServices , IOTASplashScreenServices , IOTAWizard , OTA , RAD Studio , SplashScreenServices