Dave's Development Blog
Software Development using Borland / Codegear / Embarcadero RAD Studio

Understanding the IDE you're running in

By David | October 4, 2018 0 Commer

Overview

So I've been quiet for a while as I thought I didn't really have much new to talk about until I decided I needed a new Open Tools API project to make it easier to added, update or delete Third Party CHM HTML Help in the later IDEs (XE8 upwards). To do this at the moment you need to manually add entries to the registry key `Software\Embarcadero\BDS\##.#\Help\HelpHTML1Files\`. Why might you want to do this? Well if you use *CodeSite* or *EurekaLog*, etc you can have the IDE F1 button bring up the appropriate help topic for the identifier under the cursor (most of the time). I've also managed to find a HTML Help file for Win32 (although a bit old) so that pressing F1 on a Windows API call brings up the appropriate help.

Understanding your environment

When I started to write the method to lookup the installed HTML Help files for the appropriate IDE I found I was having to `$IFDEF` a lot of code / constants and I through that was messy and would require me to alter the code for newer versions of the IDE. So was there a way to do this dynamically (and easily)?

Environment Variables

One of the things that I'd noticed in the IDE options dialogue was that the Environment Variables section actually lists all your system's environment variables, not just those for RAD Studio. So on a hunch I decided to see if `GetEnviromentVariable` would return the `BDS` root directory for the currently running installation. Err.. yes it did (see the first line of the method below).

If you're not aware, window applications (in general) inherit the environment block from their parent process (the process that launched them, in this case Explorer). So my hunch was whether RAD Studio simply added more environment variables to its environment block using `SetEnvironmentVariable` and it seems that indeed it does.

What this means is you can now have easy access to all RAD Studios environment variables like the `BDSBIN`, `BDSCOMMONDIR`, `BDSCatalogRepository`, etc. You will have to check how far back in the past this works but this IDE plug-in is only needed for XE8 onwards as it was in XE8 that Embarcadero changed to HTML Help (thank you :-).

BDS Version Number

From the above, I now know the directory in which RAD Studio is running from and at the end of it is the version number and I need that number for my registry path to RAD Studio's settings. So I simply extract it with a regular expression (PS. the code below is NOT finished so could blow up if things are non-standard and will be improved before I'm finished).

BDS Registration Point

The last part of the puzzle is whether the IDE is running from a standard registry point or from a non-standard registry point. If you didn't know, you can always start a clean IDE if you pass the command line `-rXxxxx` to `BDS.exe` where `Xxxxxx` is an alternate registry point to the default `BDS`. This is useful for testing issues with plug-in and very important for writing and debugging them. So how do I find this out. Easy, you're in a Delphi application so iterate over the command line parameters looking for the one starting with `-r` (if it exists) and extract the remaining text as the registry point.

Implementing the above

So putting this all together I wrote the following to get all the third party CHM HTML Help files from the IDE's registry and add them to a generic list which is then rendered in a `TListView` on the options frame (the last bit isn't that important).

```
Procedure TframeTPIDEHelpOptions.InitialiseFrame;

Const
  strHelpRegKey = 'Software\Embarcadero\%s\%s\Help';
  strHTMLHelpFiles = 'HTMLHelp1Files';
  strBDSEnviroVar = 'BDS';

Var
  R : TRegIniFile;
  slHelp: TStringList;
  iHelp: Integer;
  strBDSDir: String;
  strFileName: String;

Begin
```

```
  strBDSDir := System.SysUtils.GetEnvironmentVariable(strBDSEnviroVar);
  R := TRegIniFile.Create(Format(strHelpRegKey, [GetIDERegPoint(), GetIDEVersionNum(strBDSDir)]));
  Try
    slHelp := TStringList.Create;
    Try
      R.ReadSection(strHTMLHelpFiles, slHelp);
      For iHelp := 0 To slHelp.Count - 1 Do
        Begin
          strFileName := R.ReadString(strHTMLHelpFiles, slHelp[iHelp], '');
          If CompareText(Copy(strFilename, 1, Length(strBDSDir)), strBDSDir) &lt;&gt; 0 Then
            FHelp.Add(THelpRecord.Create(slHelp[iHelp], strFileName));
        End;
    Finally
      slHelp.Free;
    End;
  Finally
    R.Free;
  End;
  PopulateListView;
  lvHelpSelectItem(lvHelp, Nil, False);
End;
```

Here's the code for getting the registration point from the command line (if it exists).

```
Function TframeTPIDEHelpOptions.GetIDERegPoint: String;

Const
  strDefaultRegPoint = 'BDS';
  iSwitchLen = 2;

Var
  iParam: Integer;

Begin
  Result := strDefaultRegPoint;
  For iParam := 1 To ParamCount Do
    If CompareText(Copy(ParamStr(iParam), 1, iSwitchLen), '-r') = 0 Then
      Begin
        Result := ParamStr(iParam);
        Delete(Result, 1, iSwitchLen);
        Break;
      End;
End;
```

There the code to get the BDS version number (as a string) from the BDS environment variable result (yes, I know it needs some error checking).

```
Function TframeTPIDEHelpOptions.GetIDEVersionNum(Const strBDSDir: String): String;

Var
  RegEx : TRegEx;
  M: TMatch;

Begin
  RegEx := TRegEx.Create('\d+\.\d', [roIgnoreCase, roCompiled, roSingleLine]);
  M := RegEx.Match(strBDSDir);
  Result := M.Value;
End;
```

I hope this has opened up a few more doors for people in terms of ideas for Open Tools API projects.

regards
Dave

Related posts:

1. Chapter 5: Useful Open Tools Utility Functions (5.7)
2. Notify Me of Everything… – Part 2.1 (Rename Fix) (5.3)
3. Chapter 4.1 – The Fix (5)

Category: Delphi Open Tools API RAD Studio