

Dave's Development Blog

Software Development using Borland /

Codegear / Embarcadero RAD Studio



Getting the up to date project source code from the IDE

By David | January 5, 2018

0 Comment

There may be instance where you want to get the source code for all your project's files in the IDE in order to parse them for some reason. Surprisingly for me I haven't done this before but now needed to do this in order to produce Check and Metric reports in [Browse and Doc It](#) for the entire active project.

You could get a list of all the files in the project and read those files from disk but if you do that you'll not have any of the in memory changed that you've made in the IDE's editors. The second option would be to ask the IDE to open all of your files in the IDE buffer (opening an `IOTAModule` loads it into memory but doesn't necessarily show the file in the editor). If you have a very large project this will be a problem and is unnecessary.

So ideally we need a means of using the IDE's live files and if the IDE hasn't got a file open, the file from disk but how do we understand what files the IDE has open? This provides to be quite straight forward. You can get all the projects files (not just `.pas` files for instance) by iterating over the `IOTAProject.GetModule` collection from zero to `IOTAProject.GetModuleCount - 1` and you'll get an `IOTAModuleInfo` interface in return. You can find out what type of module you have a reference to using the `ModuleType` property which returns an enumerate value. To understand whether the module is open in the IDE use the `IOTAModuleServices.FindModule` function. If it returns an instance, the module is open in the IDE else if it returns `Nil` the module is not open and you need to read the file from disk.

Below are a couple of methods from one of my editor views which reports coding checks on the whole project (I've removed code that's not necessary for this example).

```
Procedure TBADI ModuleChecksEditorView.SelectView;
```

```
Const
```

```
  setModuleTypesToParse = [omtForm, omtDataModule, omtProjUnit, omtUnit];
```

```
Var
```

```
  P: IOTAProject;
```

```

    iModule: Integer;
    ModuleInfo: IOTAModuleInfo;

Begin
    P := ActiveProject;
    If Assigned(P) Then
        Begin
            ...
            For iModule := 0 To P.GetModuleCount - 1 Do
                Begin
                    ModuleInfo := P.GetModule(iModule);
                    If ModuleInfo.ModuleType In setModuleTypesToParse Then
                        Begin
                            ProcessModule(ModuleInfo);
                            ...
                        End
                    End;
                End;
            End;
        End;
    End;

Procedure TBADIModuleChecksEditorView.ProcessModule(Const ModuleInfo : IOTAModuleInfo);

Var
    Module: IOTAModule;

Begin
    ...
    Module := (BorlandIDEServices As IOTAModuleServices).FindModule(ModuleInfo.FileName);
    If Assigned(Module) Then
        ExtractSourceFromModule(Module, ModuleInfo)
    Else
        ExtractSourceFromFile(ModuleInfo);
    ...
End;

```

Hope this is useful.

Regards

Dave.

Related posts:

1. [Chapter 5: Useful Open Tools Utility Functions \(8.1\)](#)
2. [Chapter 10: Reading editor code \(7.5\)](#)
3. [Handling Folding and Unfolding code in the IDE \(6.6\)](#)
4. [Chapter 12: Project Repository Wizards \(6.6\)](#)

5. [Chapter 13: Project creators \(6.5\)](#)

Category: Delphi Miscellaneous Open Tools API RAD Studio Tags: IOTAModule, IOTAModuleInfo, IOTAProject

Iconic One Theme | Powered by Wordpress