1. Delphi Package Tool (DPT)

1.1. Introduction

The latest version of this document is available here:
http://sourceforge.net/p/delphipackageto/code/HEAD/tree/bin/TheDelphiPackageTool.pdf

Important Note:
- This tool is only helpful, if you have the source code of all your packages. If you have some components without source code, then I recommend to not use this tool.
- This tool is for advanced Delphi Users and not for Beginners. The user must have knowledge about Projects, Packages and Project Group Files. The Tool does add/remove Registry-Keys in the Section "Known Packages","Disabled Packages" of Delphi!
- This tool does not change your files, unless you explicit enable the setting "Allow Delphi Package Tool to change files" in the Options-Dialog.

This tool will help you with these tasks:

- Build+Install Packages into Delphi IDE.
- Build of larger Software-Projects (containing 1..n Packages and 1..n Projects).
- Take over the Project from another computer.
- Improve your work speed.

I use it with Delphi 7,2006,XE for:

- For installing components.
- Rebuilding all Projects before checking in my files to the Source-Control.
- For switching between Projects which use different Versions and Components.
- For building the finale executable files.
- All path's handled by the Tool are relative to the location of the .bpg/.bdsgroup./.groupproj file. This means you can copy the project to another location and still recompile it.
- To compile open source projects downloaded from the internet. (it takes me only a few minutes to get the IDE ready after I have downloaded the source. Without this tool it takes hours to get all the Packages compiled and installed in the correct order.

This Tool is Freeware and Open-Source and the full source code is available sourceforge.net.

1.2.   News

Please see file **history.txt** for latest changes.
http://sourceforge.net/p/delphipackageto/code/HEAD/tree/bin/history.txt

1.3.   Why is there a need for a Delphi Package Tool ?

After reading the first chapter, you may think there is no need for a Delphi Package Tool. Well, here are some disadvantages:

Something is wrong with the Delphi IDE. Building all projects consumes a lot of memory which is also not released when you choose <Close All> from the Main menu. I had to restart Delphi to continue my work.

The Packages are just compiled/built, but not installed. You have to manually click on each Package and select <Install> command.
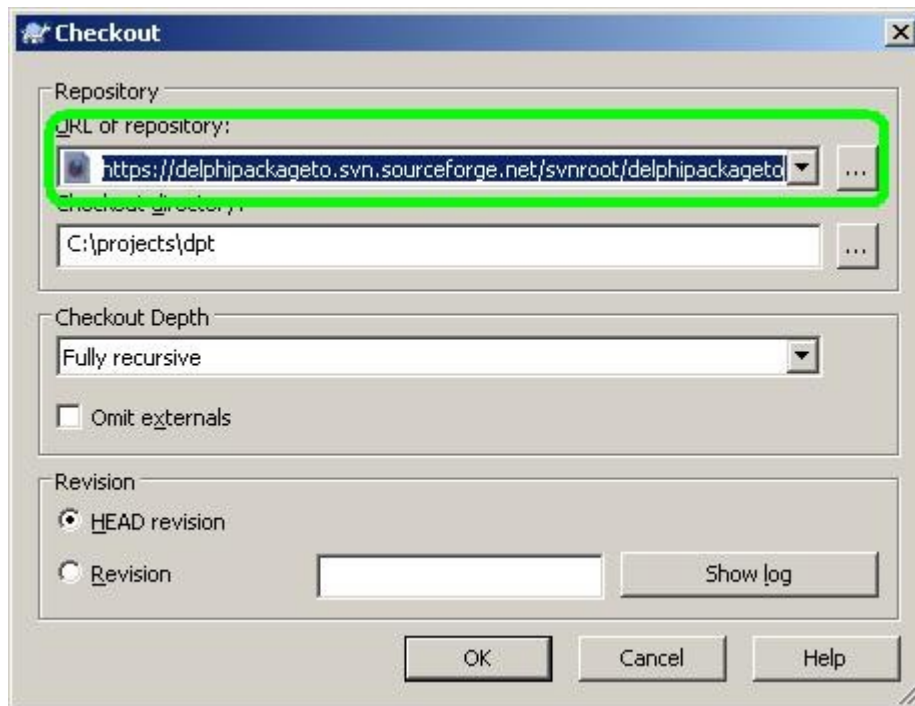
In the Delphi IDE, there is no drag&drop support to define the order of the projects.

Since the IDE needs functional packages to open a project, it's better to build& install packages without IDE.

1.4.   How to get the source code of Delphi Package Tool ?

The Source Code of DPT is available at sourceforge.net. I recommend you to install TortoiseSVN-client from here :http://tortoisesvn.net/downloads

After Installation of TortoiseSVN you can get the sources by calling function "CheckOut" like this:

https://delphipackageto.svn.sourceforge.net/svnroot/delphipackageto

To build the .exe-file just open the project in your favourite Delphi-IDE and compile.
There is no need to install additional components.

1.5. *How to help improving the DelphiPackageTool ?*

Well, the past 7 years were not easy with all these company and file changes.
There a some TODO's in the sourcecode which need to be solved.

If someone has information about return values of the command-line compiler
dcc32.exe I would be glad to get a hint.

For example:
- How to find out if the compilation was successful?
- How to find out if the compilation aborted with "file not found" message ?
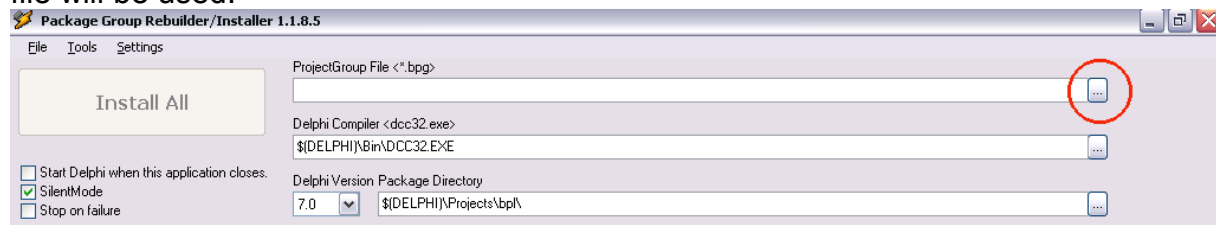
2. Features

- Same function as <Build All Projects> but without the need to start Delphi IDE. Detects if a Package is a Runtime or Designtime-Package. The Designtime-Package are being installed automatically in the Delphi Component Palette.

- The DelphiPackageTool can be used for Batch-file rebuilds of your Projects.

- The DelphiPackageTool creates .bat and .reg files.

- Makes it easy to take over a project from another Developer and prepares/installs the Components used to open the Project in Delphi.
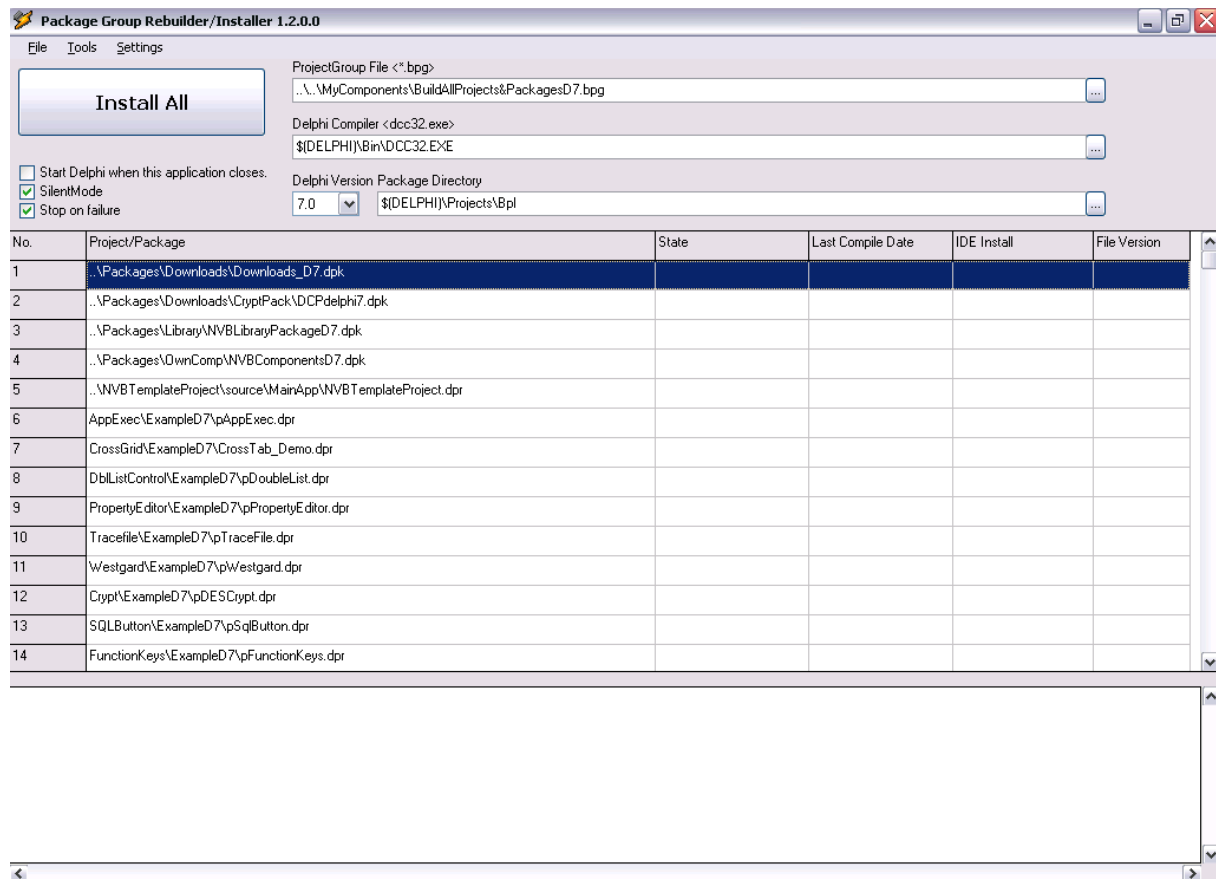
2.1.   How to use ?

I use this tool in my daily business. The tool is always running on my machine.
I use it as starting point for my tasks. So I do not click in the browser on a <.dpr> file to open the project. Instead I use this tool to open projects.
Basically you need to start working with the Package Group Files instead of handling single project <.dpr> or package files <.dpk>.

1. Close Delphi IDE.
2. Start the DelphiPackageTool.exe.
3. Check if the Delphi Compiler (dcc32.exe) and the Package Directory (\Projects\bpl) are correct.
4. Select the Package Group File you want to build or choose <New Project Group> from the Main Menu.
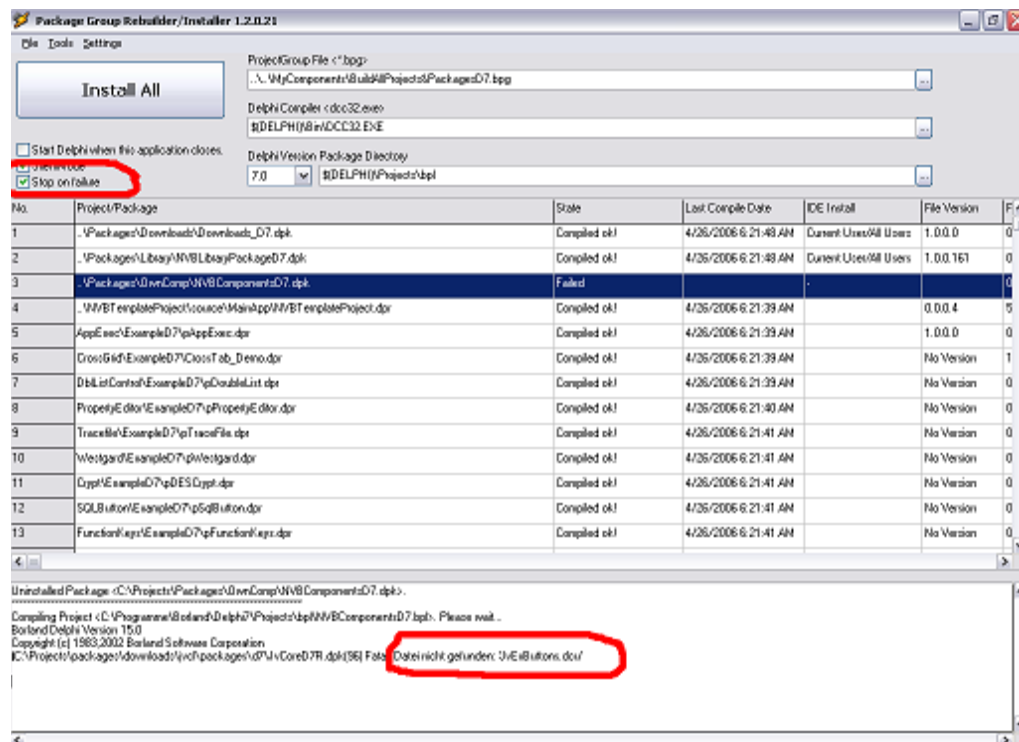
Note: The field <Package Directory> is only used if no .cfg-file is available.
If a .cfg-file is in the same directory as the .dpk file, then the output path from the .cfg file will be used.
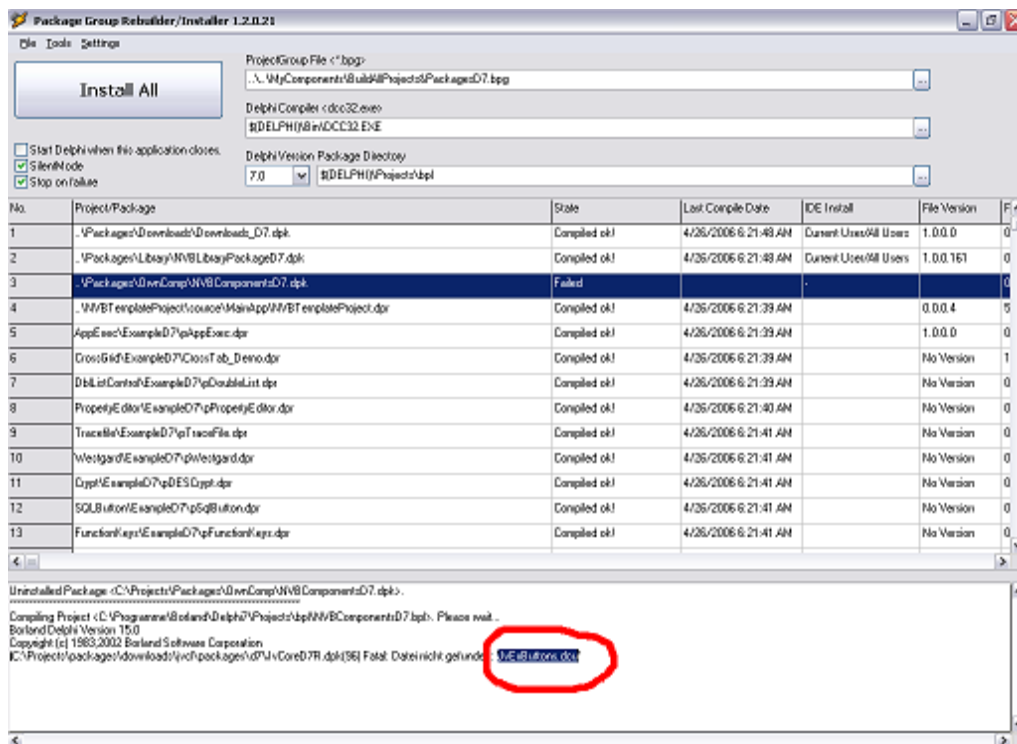

The Packages and Projects are displayed in the grid.
Make sure that Delphi is closed now. Otherwise the .bpl files can not be created because the Delphi IDE has loaded them.
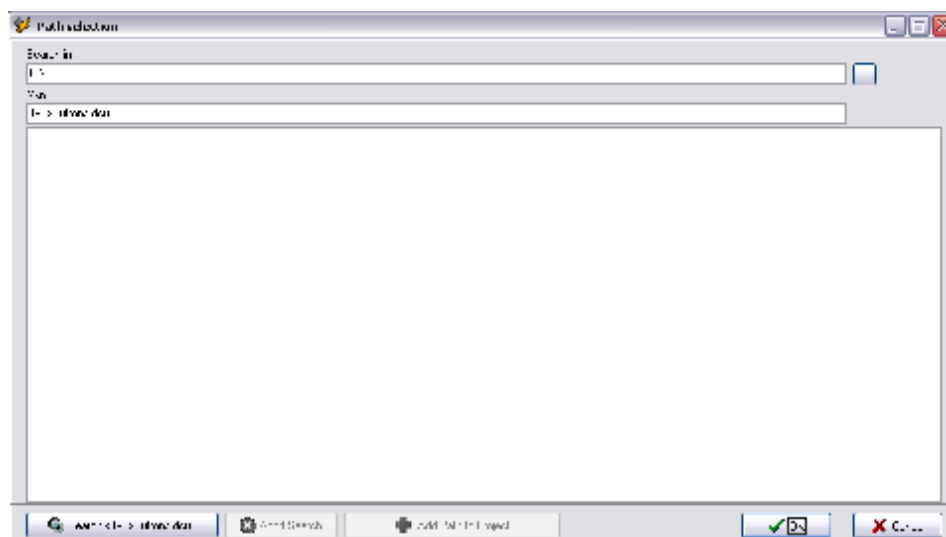
Press button <Install All>. The DelphiPackageTool starts to build all Packages and Projects. If you have tagged <Stop on failure> then the process stops on the first Package/Project that could not be compiled.
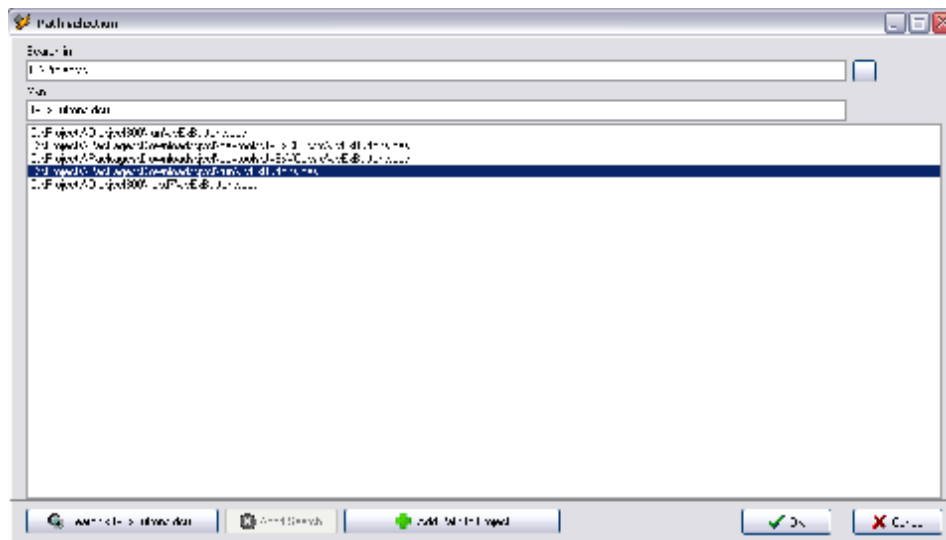
In this example it says "The FileJvExButtons.dcu could not be found". This is no problem. Just Double-Click onto the word <FileJvExButtons.dcu>.
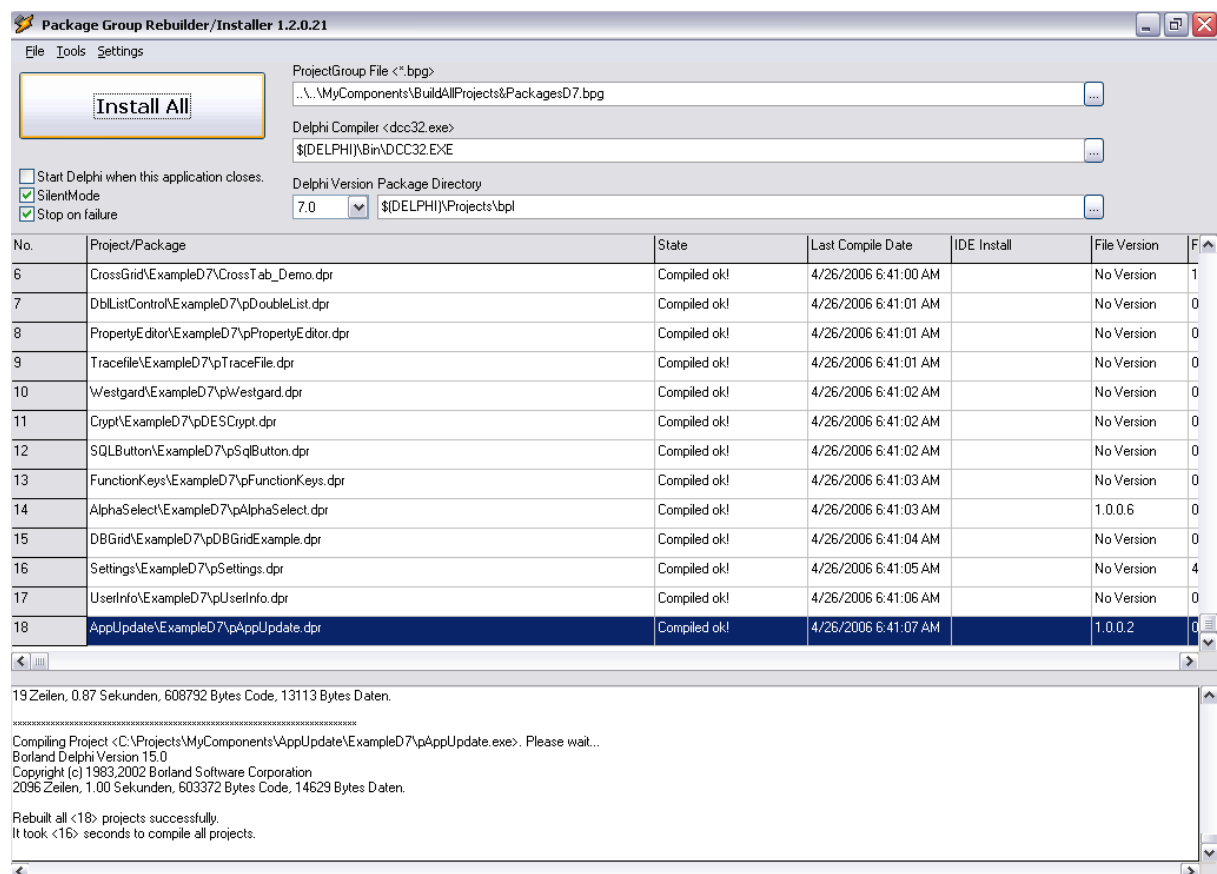

This will show up the search dialogue. Just press button <Search FileJvExButtons.*>. This will search for the units which may help.

Now select the file you need and press button <Add path to Project> and then <Ok>.

Close the Path Selection Dialogue and press again the button <Install All> and repeat the previous steps until all projects can be built.
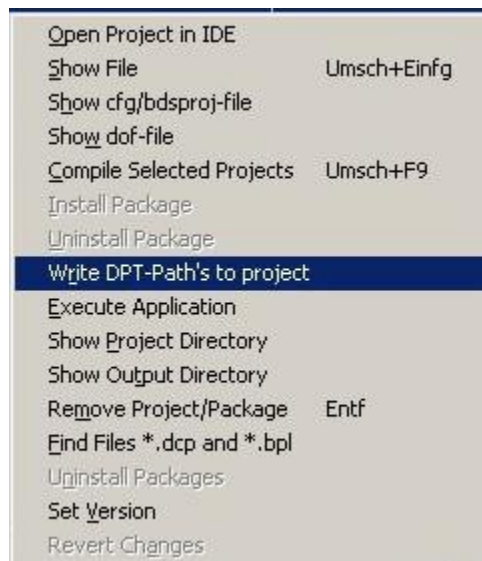

Congratulation, you did it ! All .bpl,.dcp and .exe files have been built now and the Packages have been registered in the Component Palette.

Now it's time to start the Delphi IDE.
Select the project or package you want to open and then do right click.
The following pop-up menu shows up.
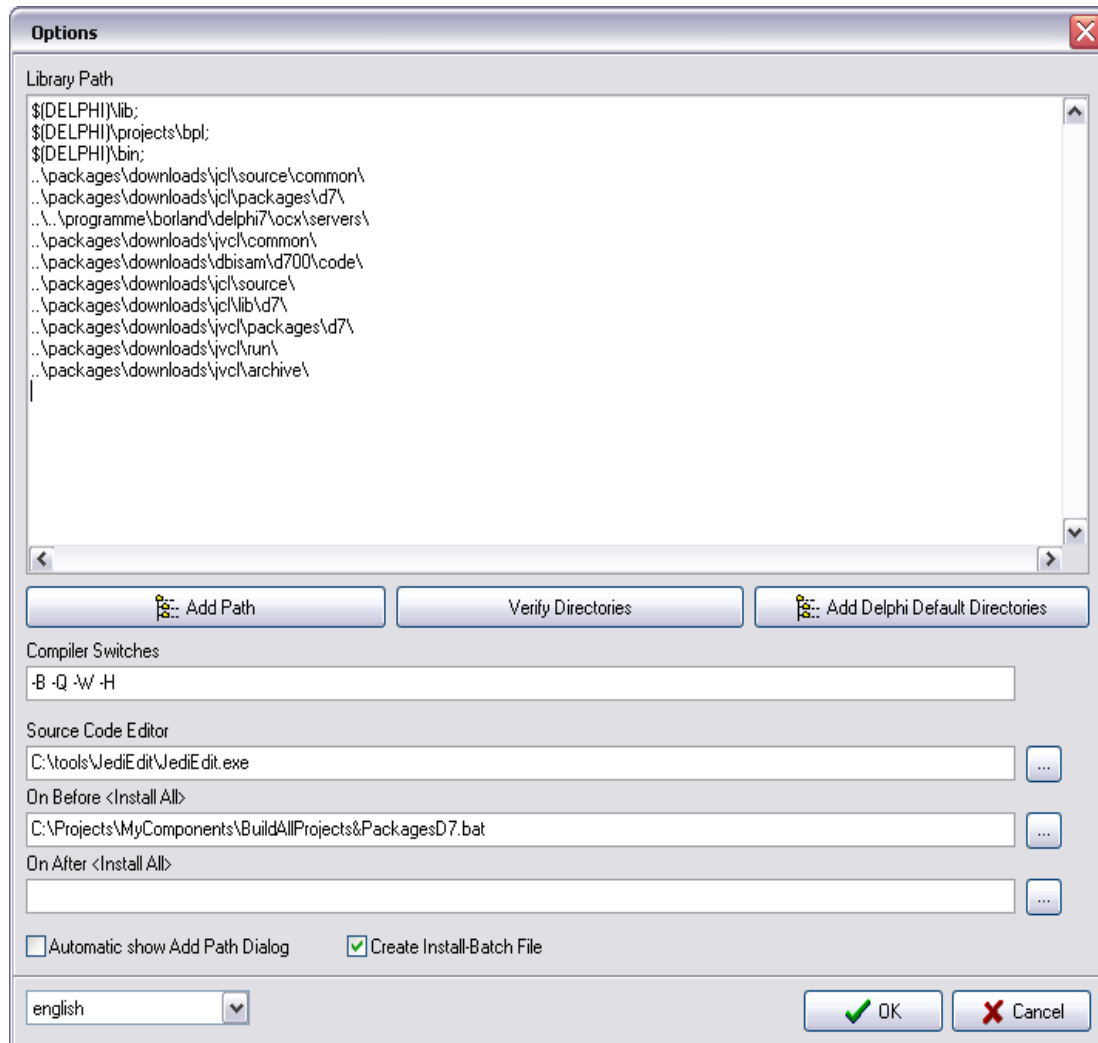
Choose <Write DPT-Path's to project>. This will add the additional path's to the project.

After that choose <Open Project in IDE>. Delphi will be started and the Project will be loaded.

- [Create a new Project-Group](#)
- [Open a Project in the IDE](#)
- [Setup Search Path](#)

2.2.   The Options Dialogue

From the Main Menu <Settings> the Options Dialogue can be opened.

Here the search paths can be defined.
Tip: Use as often as you can the Tags $(DELPHI) and $(PROGRAMFILES).

2.3. Nice to have

Call the DelphiPackageTool from the Command line

After you have successfully built all projects, you may want to control the DelphiPackageTool from a batch file.
In this case you can choose <Save Configuration> from Main menu to save the current settings to a file. E.g. myComponents.ini

Then create a batch-file with the following content.
DelphiPackageTool –o"myComponents.ini" –rebuild

For backwards compatibility with older Versions of DelphiPackageTool (Version 1.1.8.6 and older) the following statement still is valid.
Create a batch-file with the following content.

DelphiPackageTool -PC:\Projects\MyComponents\BuildAllProjects&PackagesD7.bpg -Rebuild

Save the batch file and run it. It executes the <Install All> command.
This will return value <0> if everything compiled OK. If an error happened then the
return value will be <1>.

The Batch-File needs the following files to run correctly.

DelphiPackageTool.exe
DelphiPackageTool.ini   // settings for the Delphi package tool.

yourpackagegroup.bpg   // the package group file you want to build.
yourpackagegroup.ini   // settings needed for the build process.
yourpackagegroup.txt   // contains the search path configuration

```
 // Commandline parameters
 cCleanupBplDir = '-cleanupbpldir';
 cCleanupAll   = '-cleanupall';
 cProject      = '-p';        //e.g. -p"C:\Projects\MyComponents\BuildAllProjects&PackagesD7.bpg" -rebuild
 cRebuild      = '-rebuild';
 cConfig       = '-n';        //e.g. -nRelease
 cPlatform     = '-a';        //e.g. -aWin32
 cInstall      = '-install';
 cUninstall    = '-uninstall';
 cProjIniFile  = '-o';        //e.g. –o"myComponents.ini" –rebuild
 cSilent       = '-silent';
```

Created Batch-File

The DelphiPackageTool can also be used to build a batch file.
If you have once run <Install All>, then a batch-file is created with the same name as
the .bpg file.
Also the .reg-files for the packages will be created.
This batch-file can be used without the need of DelphiPackageTool at all.
Note: The setting <Create Install-Batch File> in the Options Dialogue must be
enabled.

2.4.   *If something does not behave like expected*

To manage larger Projects with several Packages and Projects is not easy. If something does not work like you would expect then this chapter may help.

- You can get the sourcecode of the DelphiPackageTool and try to debug it.

- Also helpful might be the generated batch file <yourpackagegroup.bat>.

2.5.   *Known Problems*

Fatal: Internal Error: D10869

If you compile the project without the compiler switch -Q (quiet) and you use Delphi 7 and some of your unit-path's are rather long, then you will see this error message. This is a bug in the compiler and been fixed in a newer version.