

1. Delphi Package Tool (DPT).....	1
1.1. News.....	3
1.2. The Delphi Way of handling large Systems.....	13
1.3. Why is there a need for a Delphi Package Tool ?.....	13
1.4. How to get the source code of Delphi Package Tool ?.....	14
1.5. How to help improve the DelphiPackageTool ?.....	14
2. Features.....	15
2.1. How to use ?.....	15
2.2. The Options Dialogue.....	19
2.3. Nice to have.....	20
2.3.1. Call the DelphiPackageTool from the Command line.....	20
2.3.2. Created Batch-File.....	20
2.4. If something does not behave like expected.....	21
2.5. Known Problems.....	21
2.5.1. Fatal: Internal Error: D10869	21

1. Delphi Package Tool (DPT)

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 and Delphi 2006 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 the full Sourcecode is available sourceforge.net.

1.1. News

Please see history.txt for latest changes.

Some user's of DPT are on their way to XE3. So I received some huge patches to support platforms and build-modes.

In November 2012, I started with Version 1.9.1.x. This version shall become usable also for XE, XE2 and XE3 and remain backwards-compatible with previous versions of Delphi. I also cleaned&re-factored the sourcecode.

News in Version 1.9.1.7 (27.12.2012)

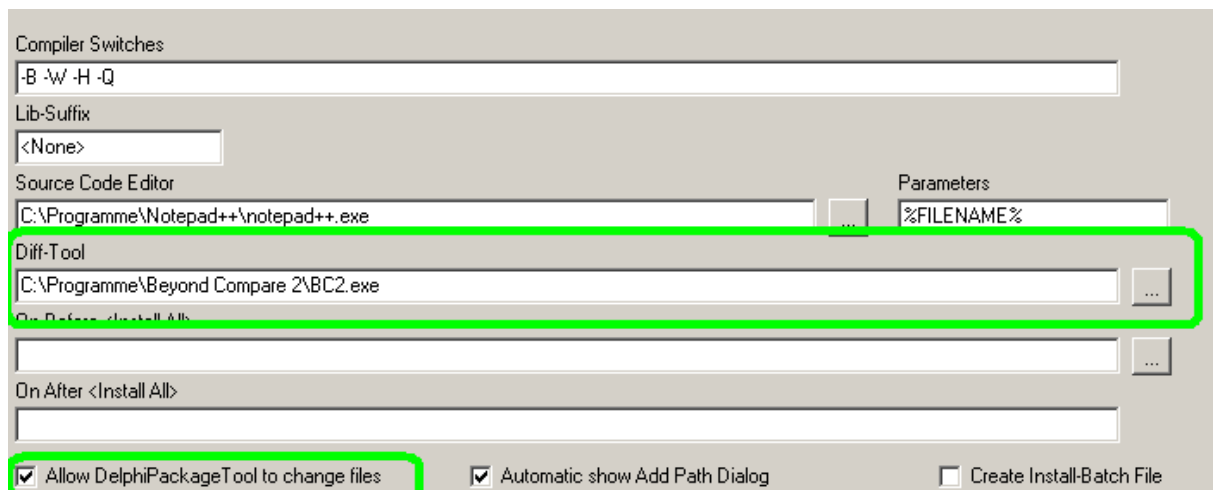
- supports platform and build-modes.

News in Version 1.9.0.139 (08.08.2010)

- Integrated SetVersion.exe (from Jason Penny) to set the Version of one/many/all projects/packages in one go.

News in Version 1.9.0.136 (10.05.2010)

- added new feature to setup a Diff-Tool to display changes.
- added patches from M.Mueller for better D2009/D2010 support.



If the setting “Allow DelphiPackageTool to change files” is enabled and you have setup a Diff-Tool, then the changes will be displayed in Diff-Tool first and when the Diff-Tool get's closed the user is asked to confirm the changes.

Note: If “Silent Mode” is enabled, then no Diff-Tool and Confirmation-Dialog will show.

News in Version 1.9.0.124 (25.03.2010)

The new release contains a lot of changes/fixes to make the DelphiPackageTool work with the different Delphi Version.

I don't want to add new features at moment.

I would like to make everything work for D5-D2010.

The DelphiPackageTool is now open-source.

See the chapter [How to get the source code of Delphi Package Tool ?](#)

Fixes are welcome !

News in Version 1.9.0.97 (05.09.2009)

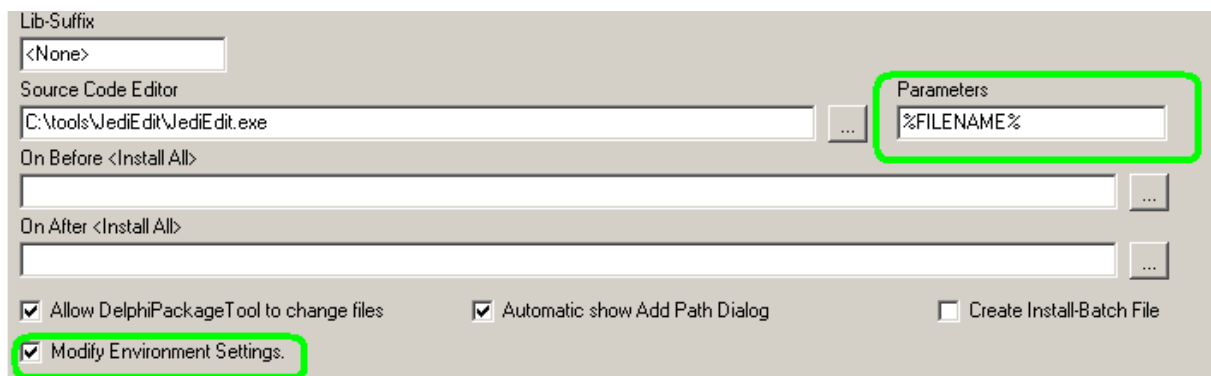
- added support for Delphi 2010. (could not test because command line compiler is not included in the Delphi 2010 trial version.)

News in Version 1.9.0.91 (15.04.2009)

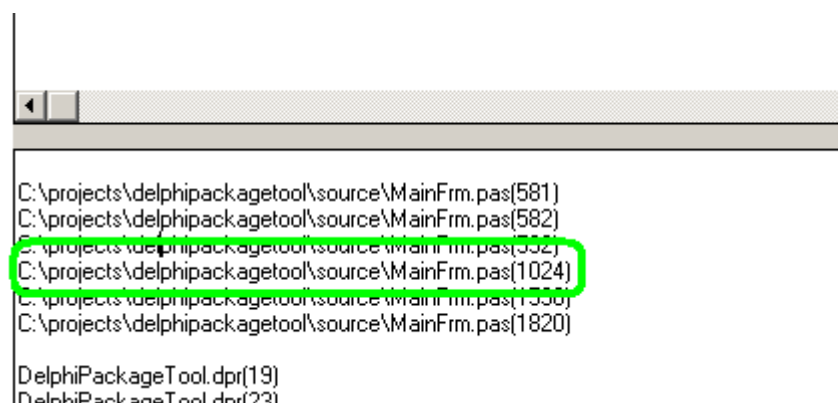
- Added Tag “\$(DELPHIVERSION)”. Useful if you want to set for example the dcu-folder to something like “..\dcu\d7”. You can now achieve this “..\dcu\\$(DELPHIVERSION)\”.
- A lot of changes/fixes to make it easier to compile a Project-Group with different compiler versions.

News in Version 1.9.0.82 (13.03.2009)

- new parameter in Options-Dialogue to define the command line parameters for the external editor. Valid place holders are %FILENAME% and %LINENO%.
- New Setting <Modify Environment>. If enabled, then the output-folder for .bpl/.dcp files will automatically be added to the Environment Settings of your computer.



- If you double click in the bottom Log-Window onto a filename, then the external Source Code Editor will show up and display the file.



- A lot of other improvements and still work to support newer Delphi Versions.

News in Version 1.9.0.26 (02.06.2008)

- a lot of changes to support .dproj and .bdsproj files.
Borland/Codegear made a lot of changes to the project-management and introduces new file-types with every new version.
It also looks like the support for .cfg files has been dropped in D2007.
Also the registry-keys are changed from version to version.
Since I do not own all Delphi Versions and also have not enough time to install/test with all versions, I need your feedback in case something does not as expected.

I also recommend reading of chapter [#3.4.If something does not behave like expected|outline](#)

News in Version 1.3.0.12 (18.02.2008)

- recompiled after change in tracefile component.

News in Version 1.3.0.11 (13.02.2008)

- removed the VCLSkin Component for two reasons:
 - a) it gave troubles.
 - b) this shall become an open source project and therefore the closed-source stuff must be removed.

News in Version 1.3.0.10 (08.02.2008)

- internal re-design to make it open-source one day.
- removed old-style multilanguage stuff. Moved strings to resourcestring section.
- changed the command SelectDirectory. (suggested by R.Kaiser).
- the application remembers now the position and size of the main form. (suggested by R.Kaiser).

News in Version 1.3.0.9 (12.12.2007)

- Fix for Problem with Decimal-Separator “,”.
- Clean-up in the tracefile.

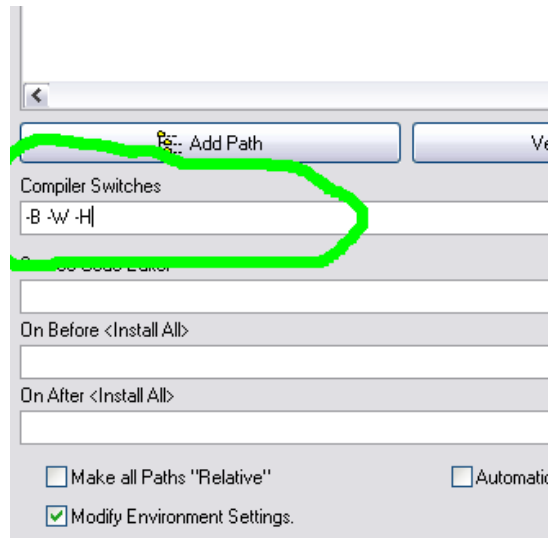
News in Version 1.3.0.0 (16.10.2007)

- The tool can now create a backup-zip file of the files used for compilation.

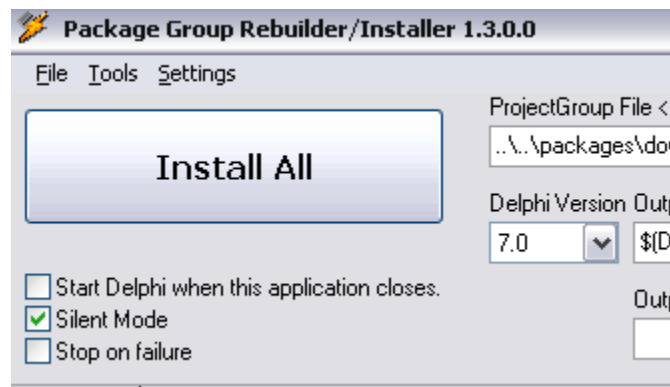
The main idea behind the new feature is, to create zip-file with all the files needed to give it to another person/workplace.

How to use:

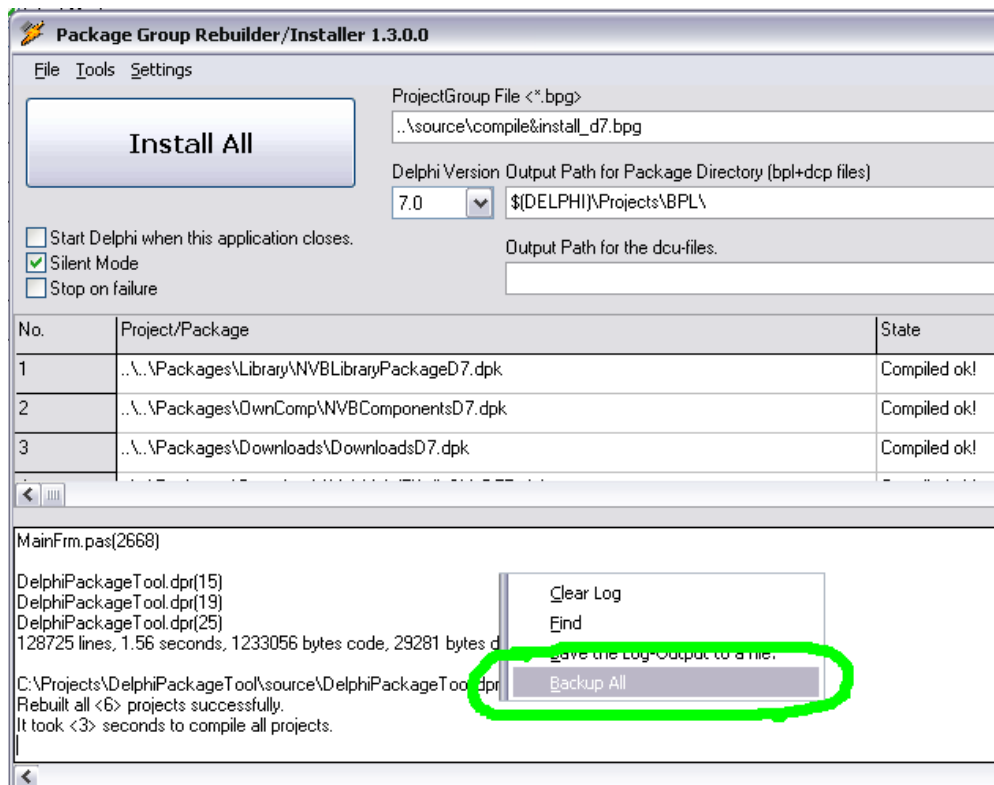
- 1.) Make sure the compiler switch –Q (quiet) is removed from
<Settings><Options>.



2.) Press on Button <Install All>.



3.) Choose <Backup All> from the Popup-Menu of the bottom Memo.
The output of the command-line compiler dcc32.exe will be searched for path/file names used for compilation and the creates a zip-file.



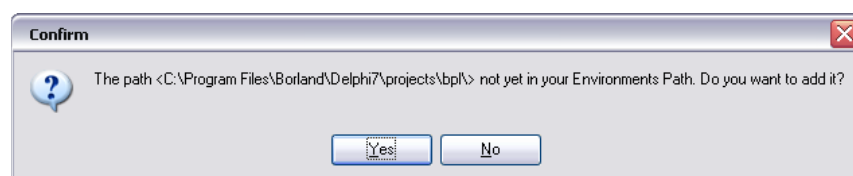
News in Version 1.2.0.73 (27-09.2007)

- fix for conditional defines. Did not work until now.
- changed way how the createProcess is used to call dcc32.exe. This has somehow influence on the working path.
- added setting to switch on/off the feature about modifying the environment path.
- improved compiler output display in the bottom memo.

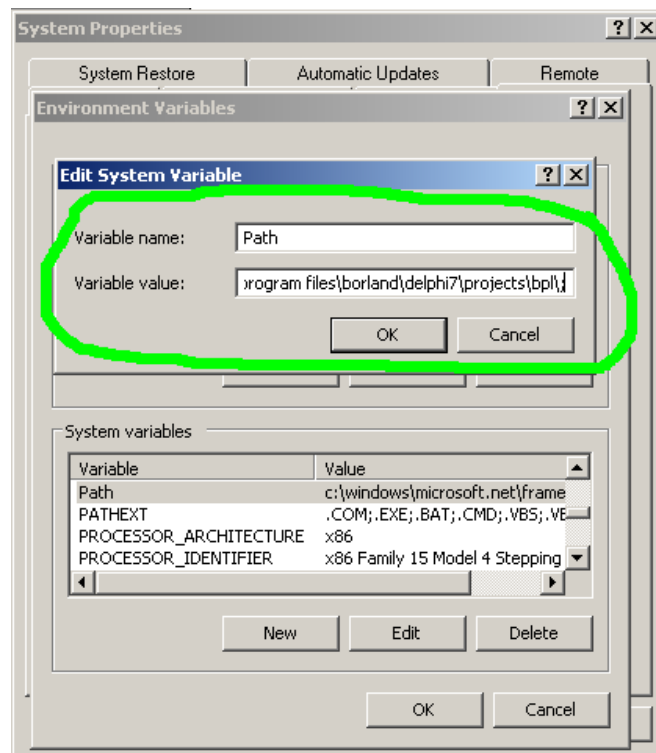
Thanks to M.Müller for the Bug-Report.

News in Version 1.2.0.70 (10-07.2007)

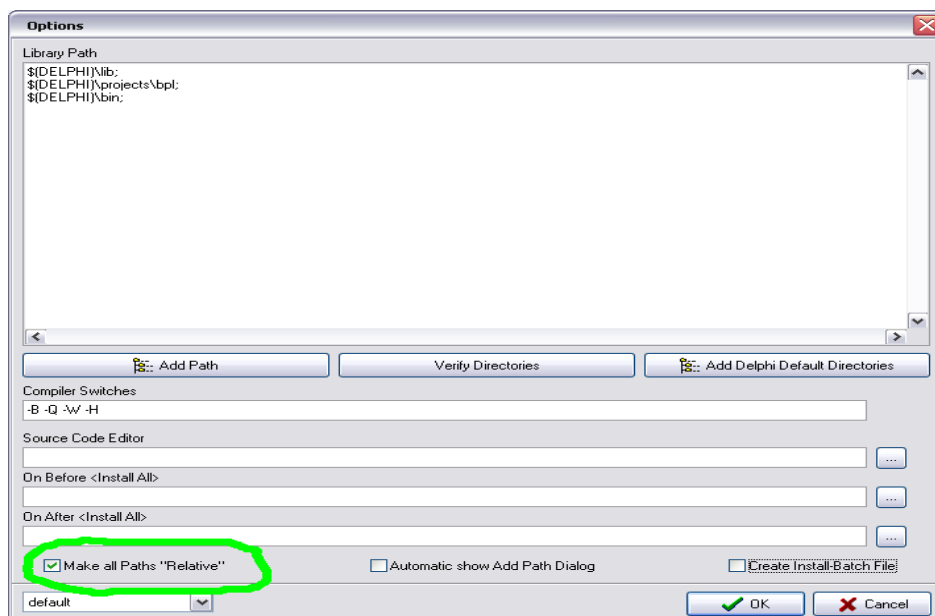
- The Delphi Package Tool now asks to add the <.bpl> directory to the windows environment path.



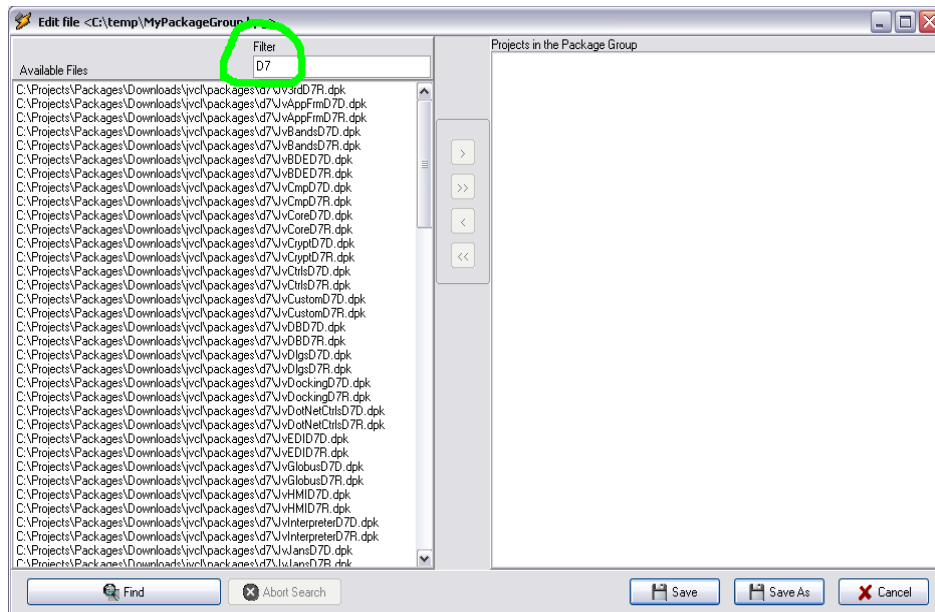
It does this stuff for you:



- this new feature <Make all path's "relative"> will remove all absolute path's in the <.cfg> and <.dof> and replace them with the corresponding relative path.

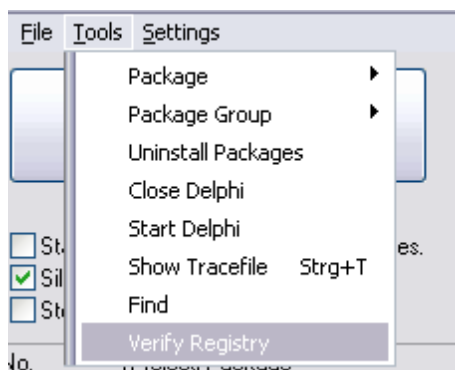


- I have added a Filterbox in the Project Group Editor. For example if you only want to have Packages which contain the <D7> tag then this box is very useful.

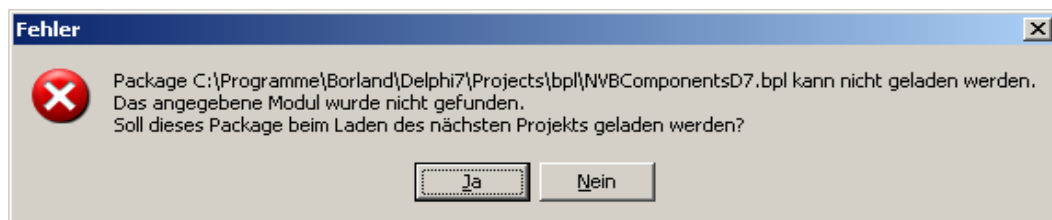


News in Version 1.2.0.31 (28-10.2006)

- The DelphiPackageTool now supports also Delphi 2006 (aka Delphi 10).
- New feature “Verify Registry”.



This will verify if the Packages referenced in the Registry are still available on the hard disk. If the file can not be found, then the wrong registry entry will be removed. This avoids messages like this when you start the Delphi IDE.

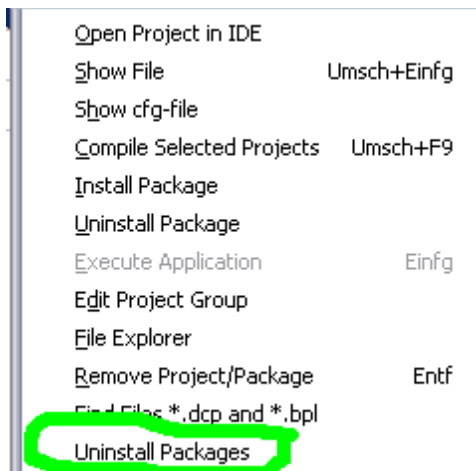


Package <XYZ.bpl> could not be loaded. The module could not be found.
Should this Package be loaded next time when opening the project ?

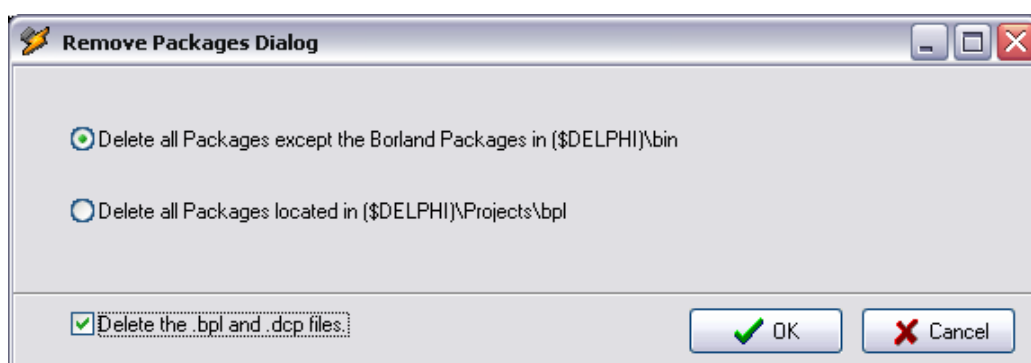
If you open your projects directly from within the DelphiPackageTool, then above described feature is automatically executed.

News in Version 1.2.0.22 (07-07.2006)

Added “Uninstall Packages”. The purpose of this feature is to have the possibility to clean up the Packages in Delphi. If you often have to switch between projects which use different packages or different versions of the same packages then this feature might be interesting for you.



The following dialogue will show up.



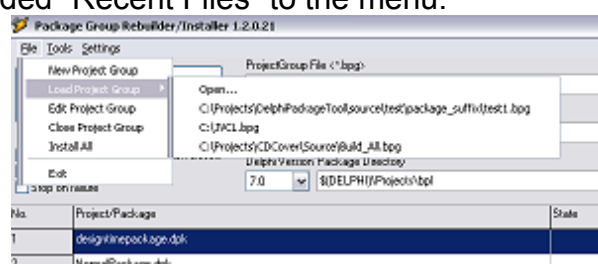
The first selection will remove all 3rd-Party Packages from the Delphi IDE. It removes the registry keys of all Packages which are **not** located in the Default Delphi Binary-Path. (\$DELPHI)\BIN.

The second selection will remove all Packages which are located in the Directory (\$DELPHI)\Projects\bpl. The registry keys will be deleted.

If the check box “Delete the .bpl and .dcp files” is checked, then the according files will be deleted from the hard disk !

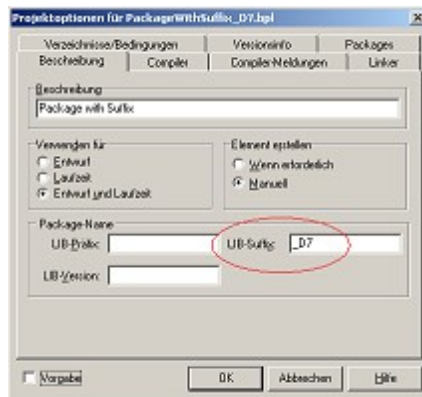
News in Version 1.2.0.21 (25-04.2006)

Added “Recent Files” to the menu.



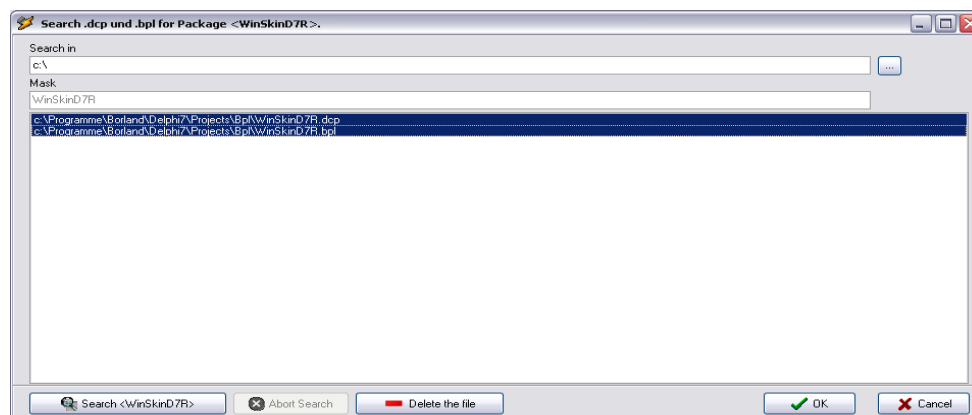
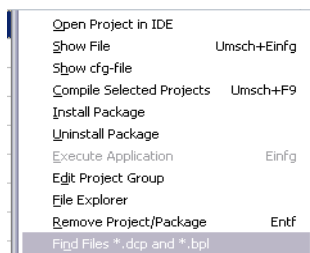
News in Version 1.2.0.20 (20-04.2006)

Added support for Packages which use the suffix feature of Delphi.



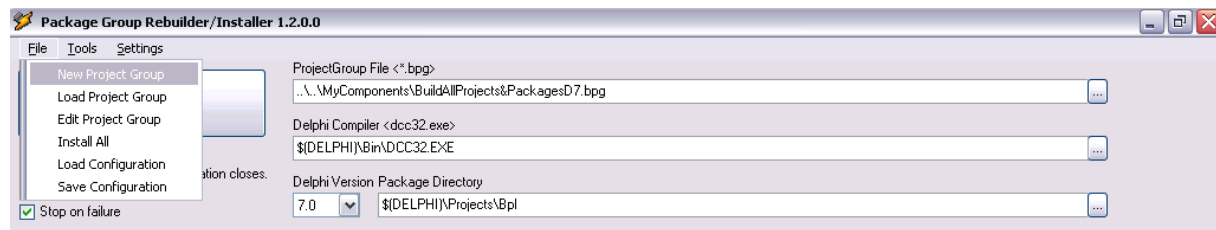
News in Version 1.2.0.16 (18-03.2006)

The Tool has a new dialogue to search and delete the .bpl and .dcp file.
To use this feature choose from the pop-up menu the Entry "Find Files *.dcp and *.bpl".

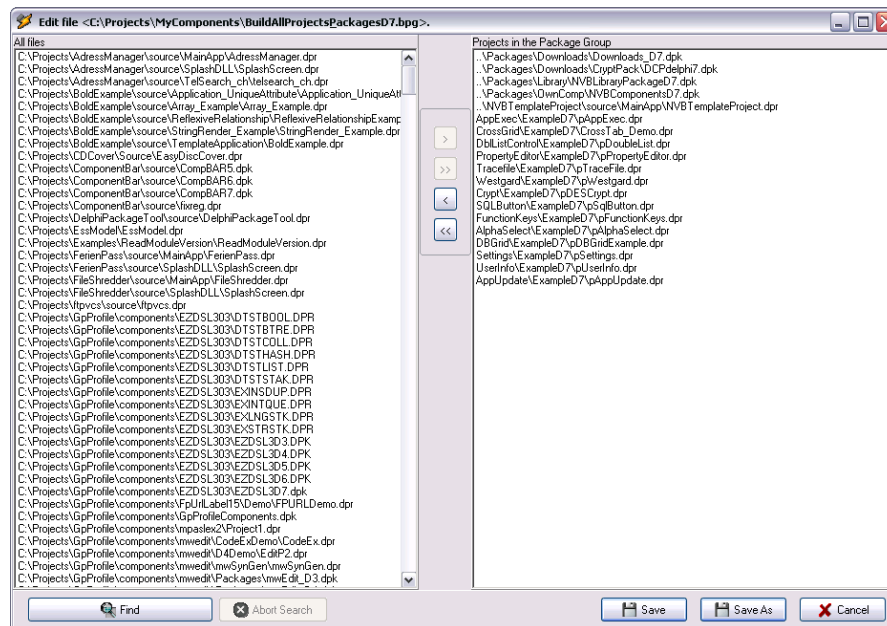


News in Version 1.2.0.0

Since Version 1.2.0.0 the Package Tool is now able to create/re-arrange the Project Group Files. Until now this task had to be done in the Delphi IDE.



The Project Group File <.bpg> can now be arranged very easy by drag & drop.



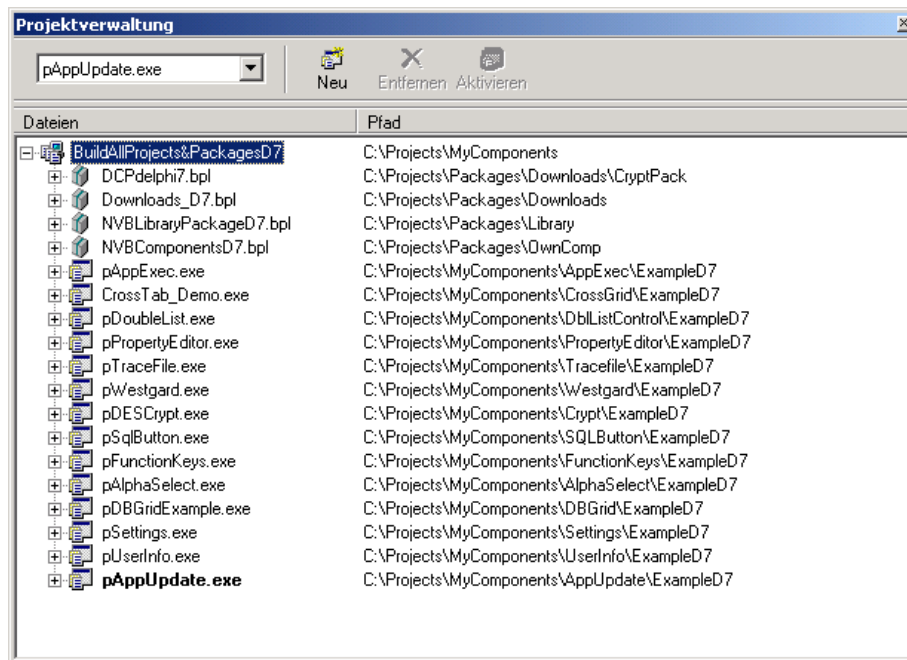
The new Version now stores all path/file names with relative Path. Relative path means relative to the location of the <.bpg> file.

The language of the tool can now be switched between German and English. If anyone wants to translate to French I would be happy ;-)

1.2. The Delphi Way of handling large Systems

A larger System consists of several Packages (runtime,designtime,third-Party, own components) and several Projects.

The Delphi IDE supports this kind of Projects with the Borland Package Group File (.bpg). When pressing CTRL+ALT+F11 the Delphi Project Manager shows up.



The Project Manager let's you add Packages (.dpk) and Projects (.dpr) to this Group.

The function <Project> <Build All Projects> builds all .bpl,.dcp,.exe files.

Of course you have to order these projects in the correct way. First the component packages, then the project's! If you choose "Build All Projects" from the Delphi Main Menu (Projects) then Delphi will compile all Packages and Projects.

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.

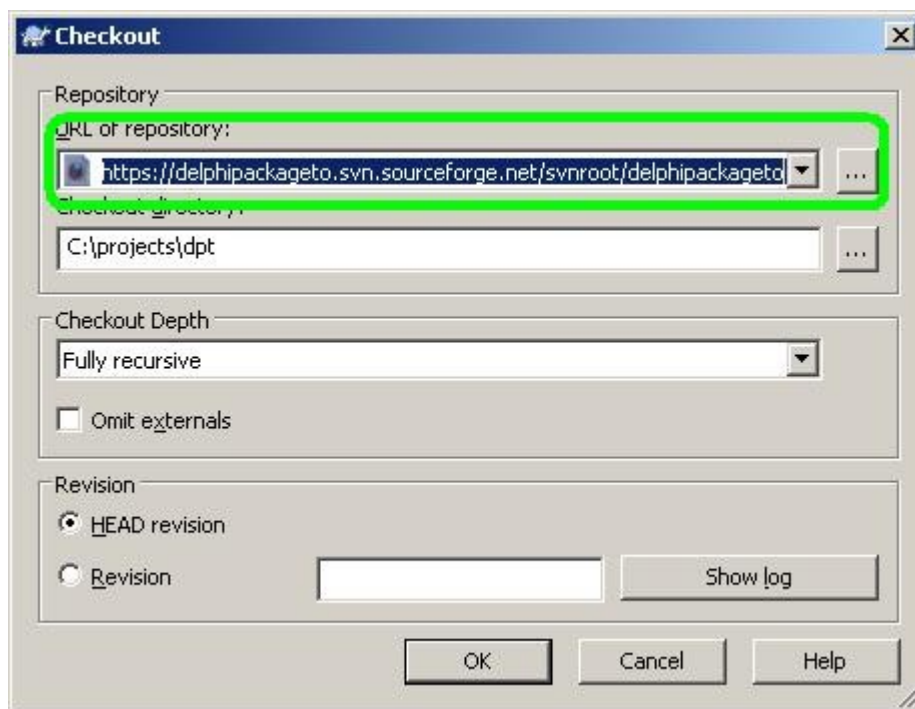
In the Delphi IDE, there is no drag&drop support to define the order of the projects.

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://delhipackageto.svn.sourceforge.net/svnroot/delhipackageto>

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 improve the DelphiPackageTool ?

Well, the past 7 years were not easy with all these company and file changes. There are 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

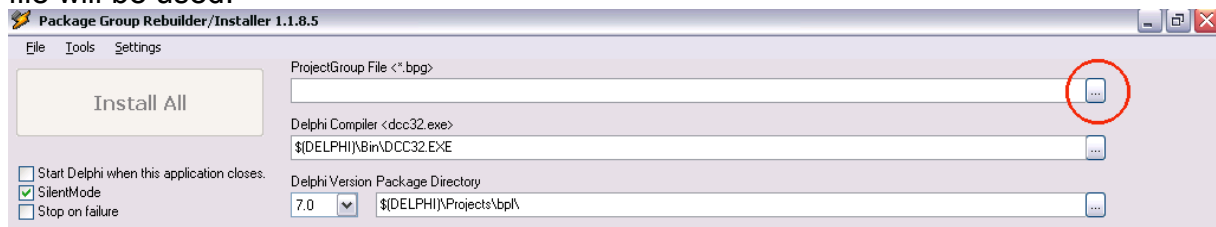
- Same function as <Build All Projects> but without the need to start Delphi IDE. Detects if a Package is a Runtime or Designtime-Package. The Designtime-Package are being installed automatically in the Delphi Component Palette.
- The DelphiPackageTool can be used for Batch-file rebuilds of your Projects.
- 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 ?*

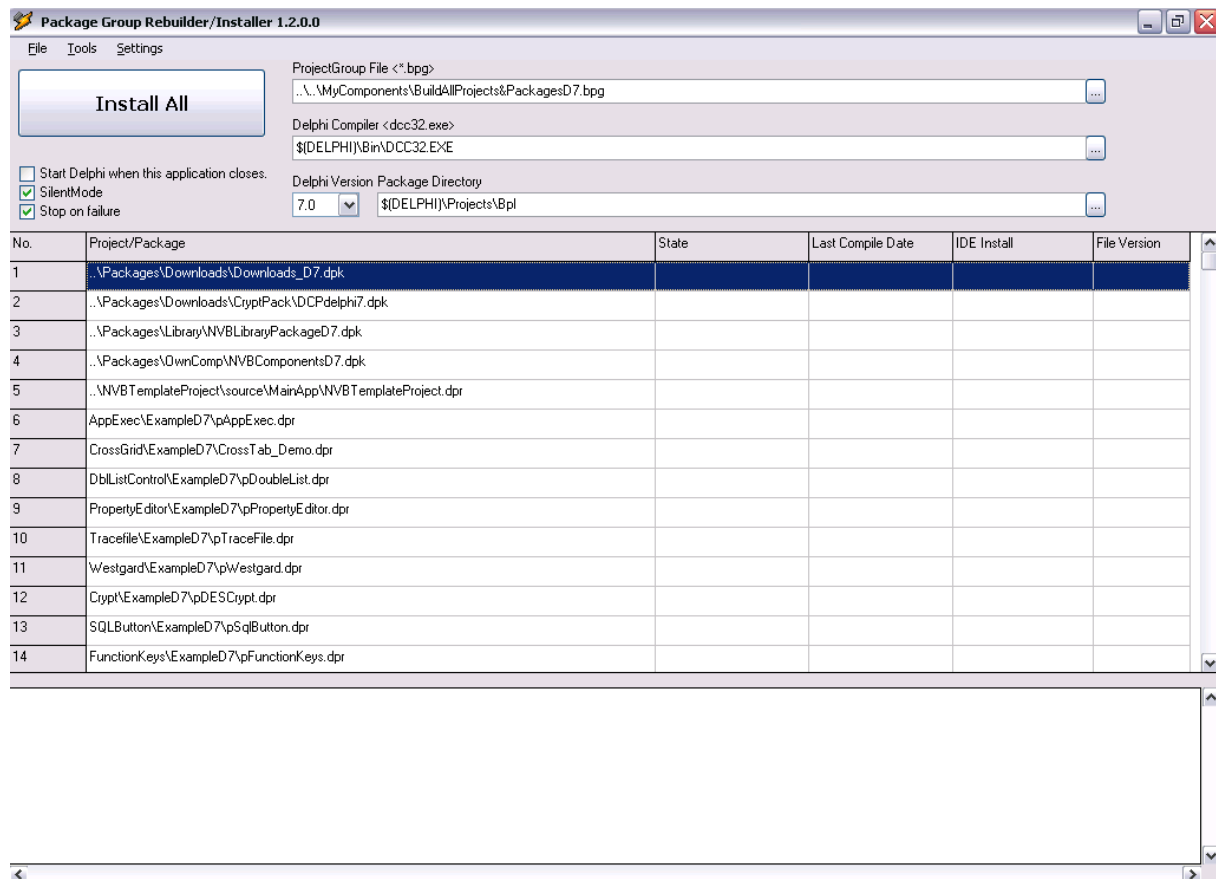
I use this tool in my daily business. The tool is always running on my machine. I use it as starting point for my tasks. So I do not click in the browser on a <.dpr> file to open the project. Instead I use this tool to open projects. Basically you need to start working with the Package Group Files instead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

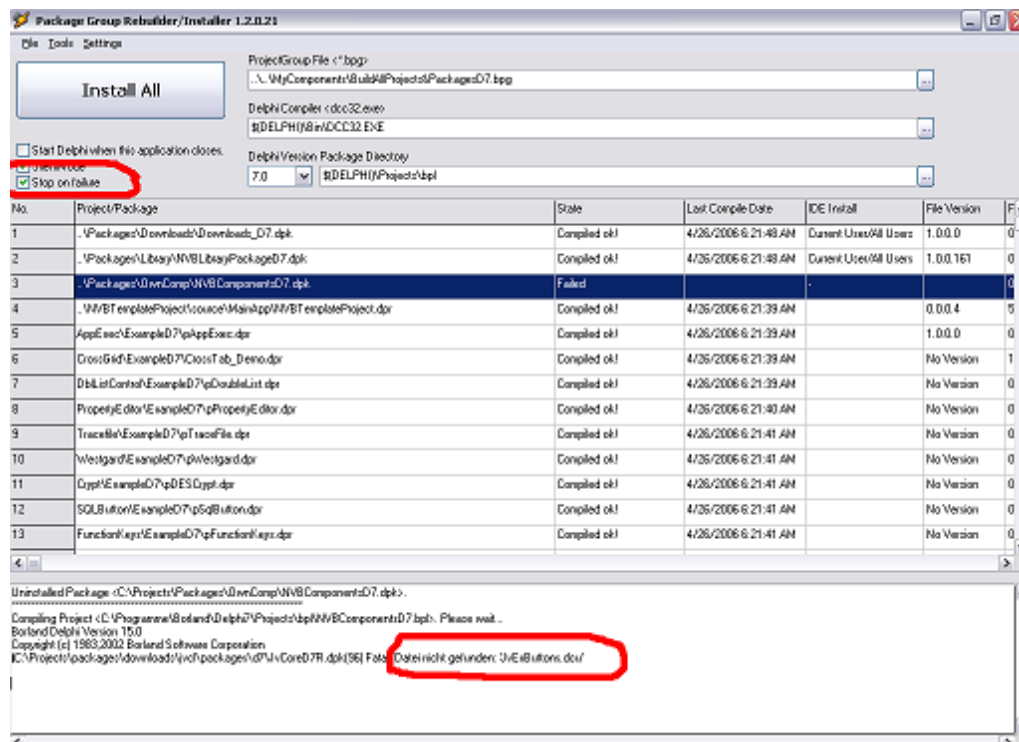
Note: The field <Package Directory> is only used if no .cfg-file is available. If a .cfg-file is in the same directory as the .dpk file, then the output path from the .cfg file will be used.



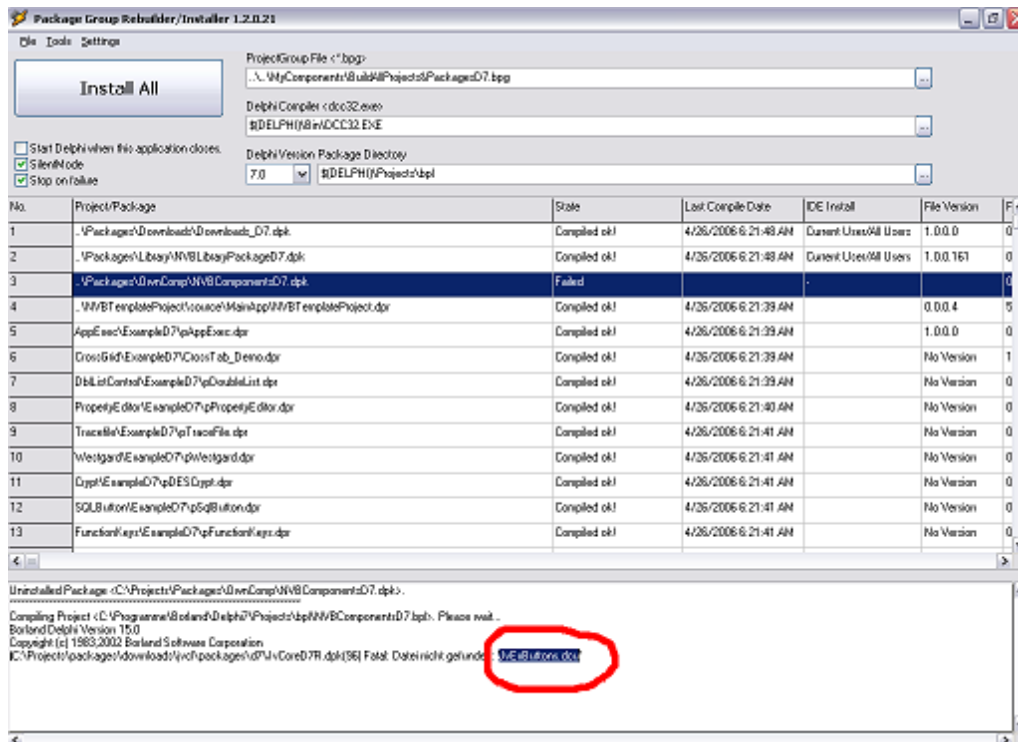
The Packages and Projects are displayed in the grid. Make sure that Delphi is closed now. Otherwise the .bpl files can not be created because the Delphi IDE has loaded them.



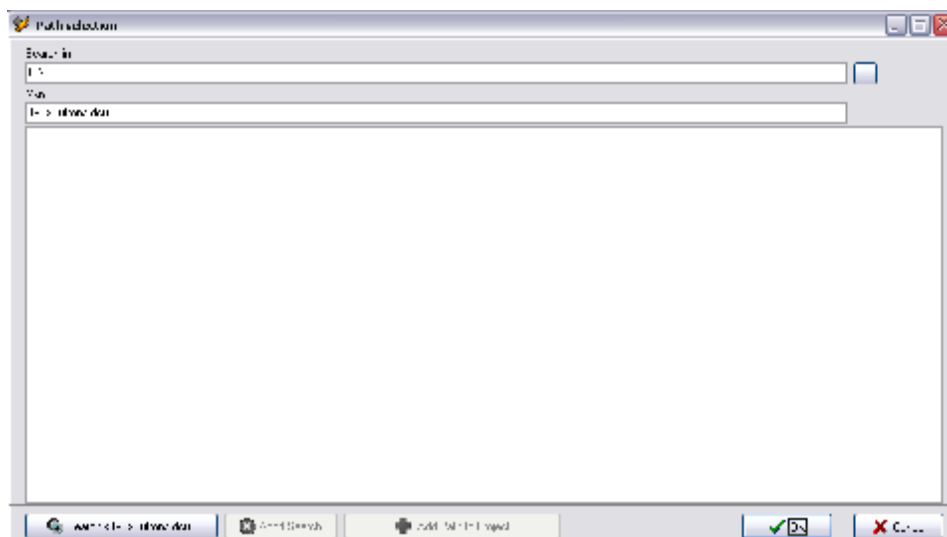
Press button <Install All>. The DelphiPackageTool starts to build all Packages and Projects. If you have tagged <Stop on failure> then the process stops on the first Package/Project that could not be compiled.



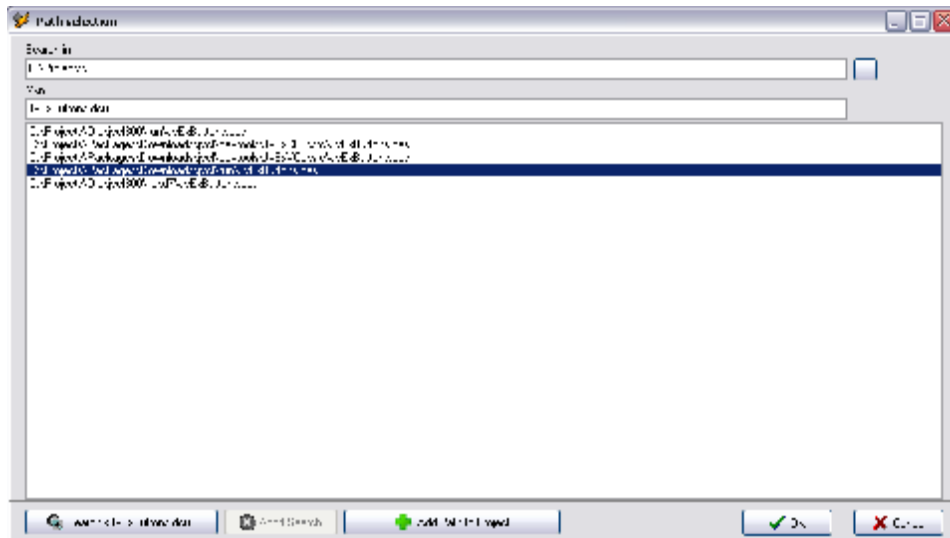
In this example it says “The FileJvExButtons.dcu could not be found”. This is no problem. Just Double-Click onto the word <FileJvExButtons.dcu>.



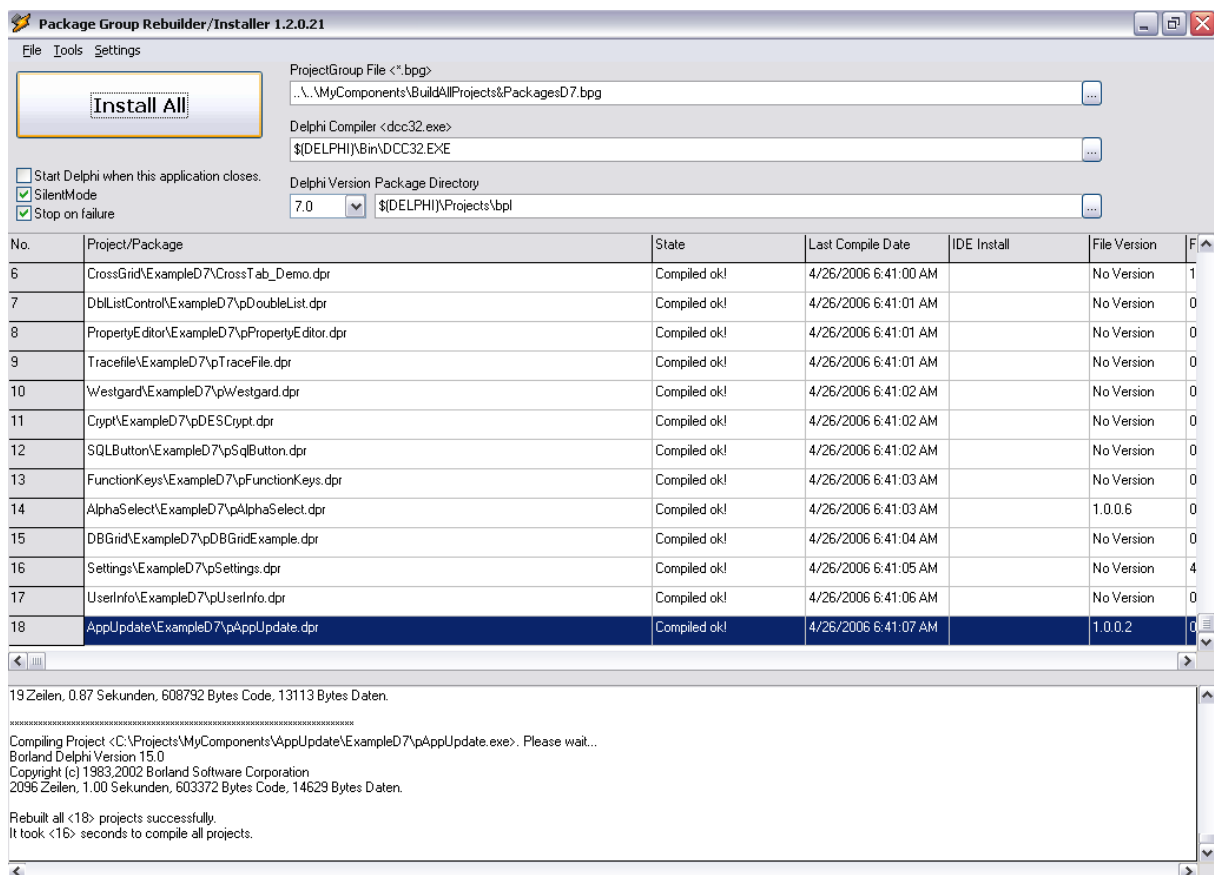
This will show up the search dialogue. Just press button <Search FileJvExButtons.*>. This will search for the units which may help.



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.

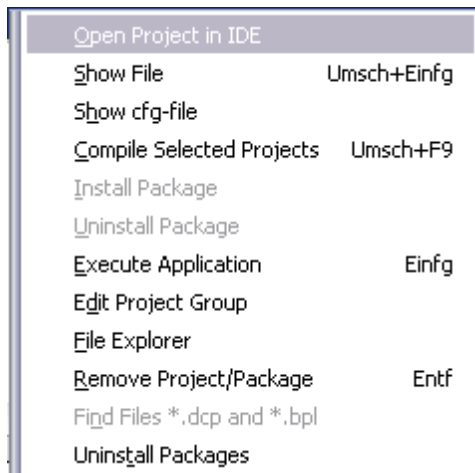


Congratulation, you did it ! All .bpl,.dcp and .exe files have been built now and the Packages have been registered in the Component Palette.

Now it's time to start the Delphi IDE.

Select the project or package you want to open and then do right click.

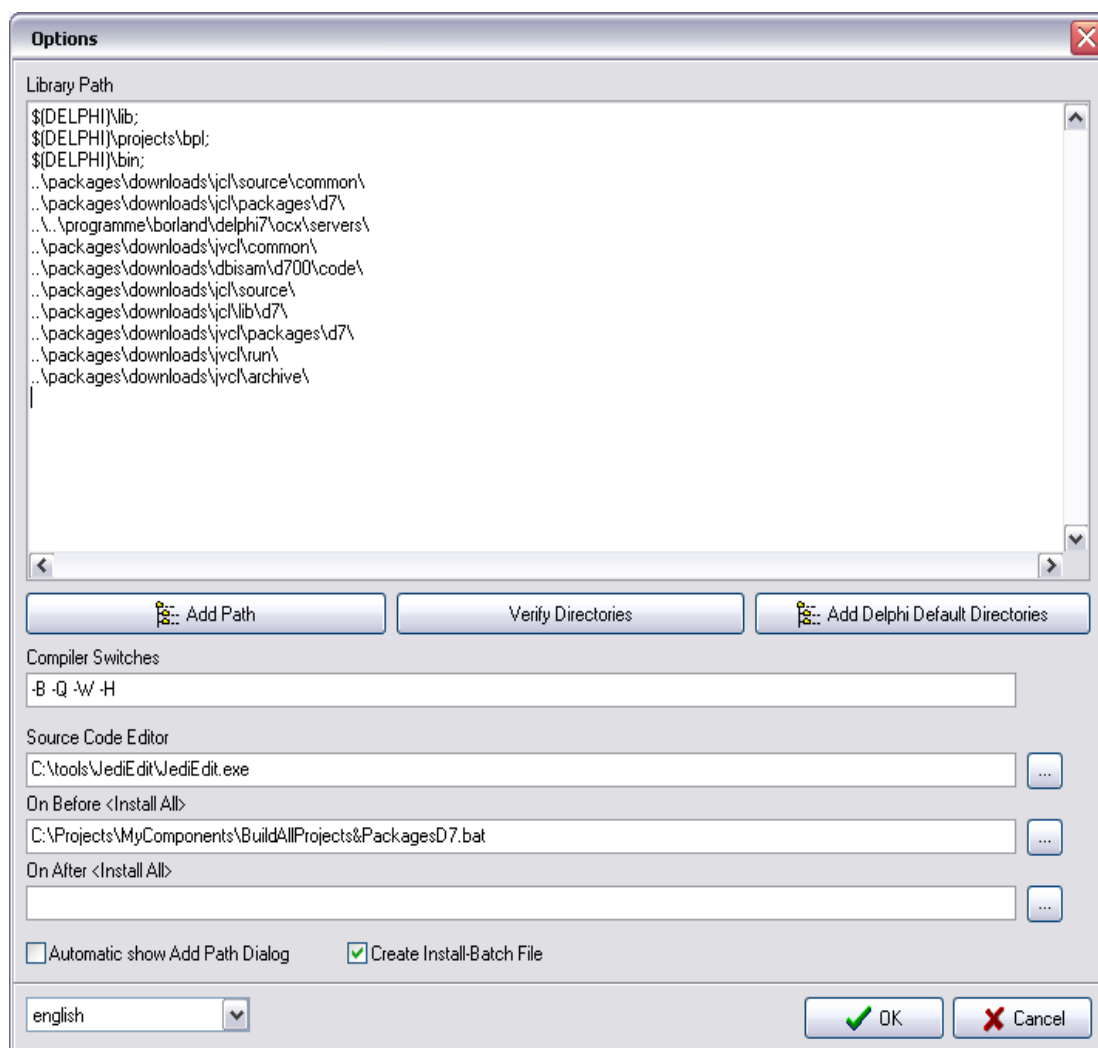
The following pop-up menu shows up.



Choose <Open Project in IDE>. Delphi will be started and the Project will be loaded.

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

2.3.1. 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.

<i>DelphiPackageTool -PC:\Projects\MyComponents\BuildAllProjects&PackagesD7.bpg -Rebuild</i>
--

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

2.3.2. 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*

2.5.1. 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.