

---

Subject: How to import existing MinGW/C++ projects into TheIDE

Posted by [guenthk](#) on Fri, 28 Apr 2006 18:14:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

We had hoped that we could utilize TheIDE as a development environment for our existing C++ project on the Windows and Linux, just as you can import existing source trees in Eclipse or KDevelop, but I couldn't find out how to achieve that.

Our project (see <http://lavape.sf.net>) is based on Qt4 and runs on Windows, Linux, and further Unixes. On Windows we'd like to support also the MinGW g++ compiler, but we have existing custom makefiles that we'd like to use also for MinGW. The project workspace contains a number of subdirectories (belonging to sub-projects having their own makefiles).

Is it possible to import such a project structure into TheIDE?

Many thanks in advance for your help.

Klaus.

---

---

Subject: Re: How to import existing MinGW/C++ projects into TheIDE

Posted by [mirek](#) on Sat, 29 Apr 2006 06:00:32 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

guenthk wrote on Fri, 28 April 2006 14:14 We had hoped that we could utilize TheIDE as a development environment for our existing C++ project on the Windows and Linux, just as you can import existing source trees in Eclipse or KDevelop, but I couldn't find out how to achieve that.

Our project (see <http://lavape.sf.net>) is based on Qt4 and runs on Windows, Linux, and further Unixes. On Windows we'd like to support also the MinGW g++ compiler, but we have existing custom makefiles that we'd like to use also for MinGW. The project workspace contains a number of subdirectories (belonging to sub-projects having their own makefiles).

Is it possible to import such a project structure into TheIDE?

Many thanks in advance for your help.

Klaus.

Basically, you have two options:

- retain current makefiles, but that will turn TheIDE just into a kind of text editor
- convert the project into U++ packages. Usually, this involves simply adding all files to the packages (and then trying to compile until it works). With 'very custom' makefiles you will likely need to play with custom build steps and conditional compiler/library options a lot....

Be warned, U++ build system is quite different from traditional makefiles, which has significant

drawback of being totally incompatible and a couple of significant advantages (up to 4x faster build times in debug mode, modularity, improved portability..).

Mirek

---

---

Subject: Re: How to import existing MinGW/C++ projects into TheIDE

Posted by [gprentice](#) on Sat, 29 Apr 2006 06:24:31 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

guenthk wrote on Sat, 29 April 2006 06:14 We had hoped that we could utilize TheIDE as a development environment for our existing C++ project on the Windows and Linux, just as you can import existing source trees in Eclipse or KDevelop, but I couldn't find out how to achieve that.

Our project (see <http://lavape.sf.net>) is based on Qt4 and runs on Windows, Linux, and further Unixes. On Windows we'd like to support also the MinGW g++ compiler, but we have existing custom makefiles that we'd like to use also for MinGW. The project workspace contains a number of subdirectories (belonging to sub-projects having their own makefiles).

Is it possible to import such a project structure into TheIDE?

Many thanks in advance for your help.

Klaus.

In addition to what Mirek said -

If you have external make files, why do you want to use U++ and why would you want to import a "source tree" into a U++ project. Is it because you want to use U++ code navigation Assist++ ?

You could see this thread.

<http://www.arilect.com/upp/forum/index.php?t=msg&th=159& amp; amp; amp;start=0&>  
and this one

<http://www.arilect.com/upp/forum/index.php?t=msg&th=183& amp; amp; amp;start=0&>

Also read [http://upp.sourceforge.net/app\\$ide\\$PackagesAssembliesAndNests\\$en-us.html](http://upp.sourceforge.net/app$ide$PackagesAssembliesAndNests$en-us.html) to understand packages, nests and assemblies.

If you want to add files to a package that are not within the package folder or subfolders, then in the lower lefthand pane of theIDE, right click and select "insert any file(s)" - then browser to a folder, select all the files you want to add and click OK. After that, check the package .upp file to see how it has added the files you selected.

Graeme

---

---

Subject: Re: How to import existing MinGW/C++ projects into TheIDE

Posted by [guenthk](#) on Sat, 29 Apr 2006 09:24:05 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

gprentice wrote on Sat, 29 April 2006 02:24

In addition to what Mirek said -

If you have external make files, why do you want to use U++ and why would you want to import a "source tree" into a U++ project. Is it because you want to use U++ code navigation Assist++ ?

...

Graeme

On the U++ home page I have read:

---

TheIDE can work with GCC, MinGW and Visual C++ 7.1 or 8.0 compilers (including free Visual C++ Toolkit 2003 and Visual C++ 2005 Express Edition) and contains a full featured debugger. TheIDE can also be used to develop non-U++ applications.

---

From this I have concluded that it should be possible to use TheIDE also for non-U++-based projects, particularly the code analyzer and a GUI interface to the GNU gdb debugger.

If this isn't true (isn't it?) then it doesn't make much sense for us to invest further work into becoming acquainted with U++.

Many thanks for your response,

Klaus.

---

---

Subject: Re: How to import existing MinGW/C++ projects into TheIDE

Posted by [mirek](#) on Sat, 29 Apr 2006 10:09:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

guenthk wrote on Sat, 29 April 2006 05:24

On the U++ home page I have read:

---

TheIDE can work with GCC, MinGW and Visual C++ 7.1 or 8.0 compilers (including free Visual C++ Toolkit 2003 and Visual C++ 2005 Express Edition) and contains a full featured debugger. TheIDE can also be used to develop non-U++ applications.

---

From this I have concluded that it should be possible to use TheIDE also for non-U++-based projects, particularly the code analyzer and a GUI interface to the GNU gdb debugger.

If this isn't true (isn't it?) then it doesn't make much sense for us to invest further work into becoming acquainted with U++.

Many thanks for your response,

Klaus.

"non-U++" applications here means applications not using U++ library.

Actually, situation here is the same as with e.g. MS Visual Studio - its native build system does not use makefiles as well..

Mirek

---

---

Subject: Re: How to import existing MinGW/C++ projects into TheIDE

Posted by [gprentice](#) on Sat, 29 Apr 2006 10:41:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

guenthk wrote on Sat, 29 April 2006 21:24

From this I have concluded that it should be possible to use TheIDE also for non-U++-based projects, particularly the code analyzer and a GUI interface to the GNU gdb debugger.

If this isn't true (isn't it?) then it doesn't make much sense for us to invest further work into becoming acquainted with U++.

Klaus.

I'm sure Mirek will correct me if I'm wrong, but I guess in theIDE, "Run in debugger" will first try to build the application - but since I haven't played with using external makefiles, I'm not sure what happens. If you're building with a customised U++ build and not just running Execute from a macro, then you might be able to use the debugger from theIDE.

I believe the code analyser (Assist++) works with package files only and has a limitation with macros, but it could still be useful - you've probably seen this.

[http://upp.sourceforge.net/app\\$ide\\$Assist\\$en-us.html](http://upp.sourceforge.net/app$ide$Assist$en-us.html)

BTW - don't miss Alt-left and Alt-right with code navigation - I'm not sure if they're documented yet.

Graeme

---

---

Subject: Re: How to import existing MinGW/C++ projects into TheIDE

Posted by [unodgs](#) on Sat, 29 Apr 2006 11:28:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

gprentice wrote on Sat, 29 April 2006 06:41

BTW - don't miss Alt-left and Alt-right with code navigation - I'm not sure if they're documented yet.

Graeme

To remeber everyone ctrl + alt + (right/left) - go left/right according to opened tabs

---

---

Subject: Re: How to import existing MinGW/C++ projects into TheIDE

Posted by [guenthk](#) on Sat, 29 Apr 2006 11:43:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

luzr wrote on Sat, 29 April 2006 06:09

"non-U++" applications here means applications not using U++ library.

Actually, situation here is the same as with e.g. MS Visual Studion - its native build system does not use makefiles as well..

Mirek

We use VC++ Express 2005 for the VC++ version of our software. So we could also use the U++ build system, if we didn't need to change our principle structure of sub-directories/sub-projects and if we could stay with Qt instead of using the U++ libraries.

For us, the main problem seems to be the very rudimentary state of the U++ documentation and the fact that Qt and Qt applications make heavy use of macros in C++, which might disturb the U++ code analyzer.

I haven't been able so far to jump from the implementation of a class method to its declaration: This always brings me just to the beginning of some arbitrary header file that is completely unrelated to the class/method in question. Will this problem disappear when using the U++ build system?

In order to make use of the U++ build system: Would I have to make U++ packages from our existing sub-projects? Will this work without also using U++ built-in packages and libraries?

Is it possible to use the same U++ configuration (\*.upp) files on Windows as well as Linux?

When using the U++ build system: Does U++ provide an easy way to jump from the compiler's error output to the respective erroneous source line by a single (or double) click?

---

---

Subject: Re: How to import existing MinGW/C++ projects into TheIDE

Posted by [mirek](#) on Sat, 29 Apr 2006 14:07:01 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Quote:

For us, the main problem seems to be the very rudimentary state of the U++ documentation and the fact that Qt and Qt applications make heavy use of macros in C++, which might disturb the U++ code analyzer.

Well, that is the problem. At current stage of development, analyzer ignores #include files (because otherwise it would be too slow), therefore knows nothing about macros.

However, there is a workaround - "Ignore list". Try to add macro names to it (it is item in Assist menu) and parser will simply ignore them. Of course, solves just basic issue (just enough to make it work for U++ library...)

Quote:

unrelated to the class/method in question. Will this problem disappear when using the U++ build system?

Actually, it is not about using build system. It is just that only files that are "in project" are scanned using the parser.

Quote:

In order to make use of the U++ build system: Would I have to make U++ packages from our existing sub-projects?

Yes. However, while enforcing directory structure just like in U++ is desirable (because it makes your project portable), you could also use "open any file" and keep files in original folders - however that will put full pathes to the .upp file.

Thinking about it, if your project is not very big (e.g. less than 20 files), you could consider putting it all into single package, arrange so that .upp file is its "root" (because all package files are relative to .upp file location).

Quote:

Will this work without also using U++ built-in packages and libraries?

Yes.

Quote:

Is it possible to use the same U++ configuration (\*.upp) files on Windows as well as Linux?

Of course, that is the point!!!!

Quote:

When using the U++ build system: Does U++ provide an easy way to jump from the compiler's error output to the respective erroneous source line by a single (or double) click?

Yes. Double-click or F4.

Mirek

P.S.: there was another related Qt question a couple of days ago here. Perhaps you should cooperate; I am no Qt expert and just barely suspect that in order to compile, you will have to add custom build steps for "moc"...

P.P.S.: I am really not sure whether this will work... Admittedly, TheIDE is designed with U++ library in mind (just like MS Visual Studio is designed with MFC/.NET in mind).

---