

---

Subject: Re: setting include-path  
Posted by [mr\\_ped](#) on Thu, 08 Nov 2007 20:32:24 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

No, just check the "be verbose" ON, and than recompile the package with your source.

In the status window where normally the compilation progress is shown you will see not just which files and packages are build, but also the command line used to invoke compiler will be printed.

Like this for example:

```
cd /home/ped/UppVA/VATestsAll
VATestsAll.cpp
c++ -c -I"/home/ped/UppVA" -I"/home/ped/upp/uppsrc" -I"/usr/include/freetype2"
-I"/usr/include/gtk-2.0" -I"/usr/include/glib-2.0" -I"/usr/lib/glib-2.0/include" -I"/usr/lib/gtk-2.0/include"
-I"/usr/include/cairo" -I"/usr/include/pango-1.0" -I"/usr/include/atk-1.0" -DflagMAIN -DflagGCC
-DflagDEBUG -DflagSHARED -DflagDEBUG_FULL -DflagBLITZ -DflagLINUX -DflagNOLIB
-DbmYEAR=2007 -DbmMONTH=11 -DbmDAY=8 -DbmHOUR=21 -DbmMINUTE=24
-DbmSECOND=14 -ggdb -g2 -fexceptions D_DEBUG -O0 -x c++
"/home/ped/UppVA/VATestsAll/VATestsAll.cpp" -o
"/home/ped/upp/out/VATestsAll/GCC.Debug_full.Main.Shared/VATestsAll.o"
```

Now look for the string you did insert into "includes" where it is used in compiler invocation.

See my -I"/home/ped/UppVA" .. that's what you are looking for.

If you have your include path somewhere on the line, you only need to make sure there's correct option ahead of it (-I also for MSVC? Check it in help).

You don't really need to know cmd.exe well, just check your compiler help for command line options to understand which are used and why.

If it is missing completely, than you put that include path in some wrong place, or you did hit Cancel by accident, or it is applied only in different mode of compilation or simply something went wrong and you will have to find out what.

In such case it may be of some help to post here the content of your package.upp file which will show where the include path is stored.

It's good to know your tools a bit. You don't need to memorize every compilation option, but you should be aware how the final binary is produced, in what steps (preprocessing, object compilation, linking), and what possible options your compiler has... you should read it at least once in life.

(and maybe you will like some of them, like /Wall)

---