How to enable C++11/C++0x support in Eclipse CDT?

Ask Question

Eclipse 3.7.1 CDT 1.4.1 GCC 4.6.2

This is an example of a piece of C++11 code:

```
auto text = std::unique_ptr<char[]>(new char[len]);
```

The Eclipse editor complains about:

```
Function 'unique_ptr' could not be resolved
```

The Makefile compilation works fine. How to make Eclipse stop complaining about these sort of errors?

c++ eclipse c++11 eclipse-cdt

edited Jun 28 at 13:18
Ciro Santilli 新疆改造中心 六四事件 法轮功
126k 25 487 425

asked Feb 3 '12 at 16:13
Nick
3,453 5 21 33

12 At the very least, it should be `std::unique_ptr<char[]>` – Cubbi Feb 3 '12 at 16:16

Not sure that is correct -- char[] is convertible to char* and unique_ptr<char> is a pointer to char, like char* – Nick Feb 3 '12 at 16:24

1 Wouldn't `unique_ptr<char>` call `delete`, which is wrong since it was created with `new[]`? – Dietrich Epp Feb 3 '12 at 17:44 ✎

5 @Nick: No, he's right. The way you wrote it will cause `delete` to be

`delete[]` correctly. – Nicol Bolas Feb 3 '12 at 17:47

Related: stackoverflow.com/q/8889260/636019 , stackoverflow.com/q/8763937/636019 , stackoverflow.com/q/8564544/636019 , stackoverflow.com/q/8312854/636019 – ildjarn Feb 3 '12 at 19:04

15 Answers

I found this article in the Eclipse forum, just followed those steps and it works for me. I am using Eclipse Indigo 20110615-0604 on Windows with a Cygwin setup.

- Make a new C++ project
- Default options for everything
- Once created, right-click the project and go to "Properties"
- C/C++ Build -> Settings -> Tool Settings -> GCC C++ Compiler -> Miscellaneous -> Other Flags. Put `-std=c++0x` (or for newer compiler version `-std=c++11` at the end . ... instead of GCC C++ Compiler I have also Cygwin compiler
- C/C++ General -> Paths and Symbols -> Symbols -> GNU C++. Click "Add..." and paste `__GXX_EXPERIMENTAL_CXX0X__` (ensure to append and prepend two underscores) into "Name" and leave "Value" blank.
- Hit Apply, do whatever it asks you to do, then hit OK.

There is a description of this in the Eclipse FAQ now as well: Eclipse FAQ/C++11 Features.

Eclipse image setting

edited Apr 21 '16 at 13:12
Adib
369 3 18

answered Feb 3 '12 at 20:33

Johannes Schaub - litb Feb 4 '12 at 21:53

5 Did this remove spurious editor errors? I have no problem with compiling, and new std types are recognised, but I can't get rid of editor syntax errors for range based for loops and rvalue references `&&` . – juanchopanza May 3 '12 at 13:09

7 The natural follow-up question now becomes: can we optimize our workflow by saving these specific C++ Project Settings into a new Eclipse Project template say "C++11 Project"? – Nordlöw May 21 '12 at 21:34

38 Note: for anyone looking for the answer for Juno, this *does not work*. – Christopher Dec 31 '12 at 16:37

3 For kepler/juno, the solution is found in this answer. It's posted in the edit to Johan Lundberg's reply below, but I note it here as well, to make the accepted answer more complete. – Inusable Lumière Nov 27 '13 at 15:45

Instruction For Eclipse CDT 4.4 Luna and 4.5 Mars

First, before creating project, configure Eclipse syntax parser:

`Window -> Preferences -> C/C++ -> Build -> Settings -> Discovery -> CDT GCC Build-in Compiler Settings`

in the text box entitled `Command to get compiler specs` append `-std=c++11`

Now you can create project, configuration depends on what kind of project you created:

For project created as: File -> New -> Project -> C/C++ -> C++ Project

Right click on created project and

`C++ Compiler` -> `Dialect`

Put `-std=c++11` into text box entitled `other dialect flags` or select `ISO C++11` from the `Language standard` drop down.

For CMake project

Generate eclipse project files (inside your project)

```
mkdir build
cd build
cmake -G"Eclipse CDT4 - Unix Makef
```

Then import generated directory to eclipse as standard eclipse project. Right click project and open

`Properties` -> `C/C++ General` -> `Preprocessor Include Paths, Marcos etc.` -> `Providers`

enable `CDT GCC Build-in Compiler Settings` and move it higher than `Contributed PathEntry Containers` (This is important)

Last Common Step

recompile, regenerate `Project` -> `C/C++ Index` and restart Eclipse.

edited Sep 20 '17 at 14:19

answered Jul 3 '14 at 19:12
Trismegistos
2,565 1 14 31

2 works for me just fine – javapowered Sep 17 '14 at 12:28

6 No other answer worked for me on Luna. Thank you. – Charles W Dec 20 '14 at 22:22

5 great, works on CMake based project – liangbright Jan 2 '15 at 23:52

1 Worked perfectly even without needing an eclipse restart. Thank you. – prasannak Feb 1 '15 at 7:10

1 For Eclipse Neon.3, with a C++ Makefile project, the project setting to add `-std=c++11` to was `Project Properties -> C/C++ General -> Preprocessor Include Paths,`

in the global workspace settings as suggesting, but it wasn't copied into the location I describe above when I created a new `Makefile` project. – BeeOnRope May 27 '17 at 23:37

Update 2016:

As of gcc 6 (changes), the default C++ dialect is C++14. That means that unless you explicitly need a newer or older dialect than than, you don't need to do anything with eclipse anymore.

For Luna and Mars

This community wiki section incorporates the answer by Trismegistos;

1. Before creating project, configure Eclipse syntax parser:

Window -> Preferences -> C/C++ -> Build -> Settings -> Discovery -> CDT GCC Build-in Compiler Settings

in the text box entitled Command to get compiler specs append `-std=c++14` **2. Create project, configuration depends on what kind of project you created:**

For project created as: File -> New -> Project -> C/C++ -> C++ Project

Right click on created project and open

Properties -> C/C++ Build -> Settings -> Tool Settings -> GCC C++ Compiler -> Dialect

Put `-std=c++14` into text box entitled other dialect flags or select ISO C++11 from the Language standard drop down.

There's now a new way to solve this without the GXX_EXPERIMENTAL hack.

For most recent versions: (Currently ~~**Juno and Kepler**~~ **Luna**):

`etc. -> tab Providers -> CDT GCC Builtin Compiler Settings () .`

Older versions 2012/2013:

1. Under C/C++ Build (at project settings), find the **Preprocessor Include** Path and go to the **Providers Tab**. Deselect all except CDT GCC Builtin Compiler Settings. Then untag Share settings entries … . **Add the option -std=c++11** to the text box called Command to get compiler specs.
2. Go to paths and symbols. Under Symbols, click **restore defaults**, and then apply.

Notes:

Eclipse is picky about hitting apply, you need to do it every time you leave a settings tab.

[Self-promotion]: I wrote my own more detailed instructions based on the above. http://scrupulousabstractions.tumblr.com/post/36441490955/eclipse-mingw-builds

Thanks to the user `Nobody` at https://stackoverflow.com/a/13635080/1149664

edited May 23 '17 at 11:54
Community ♦
1 1

answered Nov 25 '12 at 7:05
Johan Lundberg
15.8k 1 53 80

3 Thank you but it didn't help me to solve c++11 references. Program compiles correctly, but eclipse cannot resolve them. When I check "array", `#ifndef __GXX_EXPERIMENTAL_CXX0X__` returns true and code below it is not parsed. When I add `__GXX_EXPERIMENTAL_CXX0X__` to project symbols, array gets resolved but my project's references gets messed up. What might be the problem here? – Halil Kaskavalci Dec

random class. Get the iterator. Try to access public variable. Either my workspace is corrupted or there is a bug in Eclipse that cannot parse this. – Halil Kaskavalci Dec 20 '12 at 17:00

1 @HalilKaskavalci I found this same bug. If I set the editor for c++11, iterators can't resolve. If I don't, std::unique_ptr can't resolve. Really annoying, since I use both. – Kian Dec 20 '12 at 22:24

1 @Kian I reported the bug. bugs.eclipse.org/bugs/show_bug.cgi?id=397027 . It is quite annoying and I started to use boost libraries, they don't produce the error :) – Halil Kaskavalci Dec 20 '12 at 23:19

1 works for me! Thank you! (did it by your reference) I'va added -std=c++11 to providers and restore defaults in symbols. I think restore default in symbols saves me. Now unique_ptr is resolving. – Denis Zaikin Oct 16 '14 at 9:22

For the latest (Juno) eclipse cdt the following worked for me, no need to declare `__GXX_EXPERIMENTAL_CXX0X__` on myself. This works for the the CDT indexer and as parameter for the compiler:

"your project name" -> right click -> properties:

C/C++ General -> Preprocessor Include Paths, Macros etc. -> switch to the tab named "Providers":

- for "Configuration" select "Release" (and afterwards "debug")
- switch off all providers and just select "CDT GCC Built-in Compiler Settings"
- uncheck "Share setting entries between projects (global provider)"
- in the "Command to get compiler specs:" add "-std=c++11" without the quotes (may work with quotes too)
- hit apply and close the options
- rebuild the index

Now all the c++11 related stuff should

mingwbuilds project on sourceforge

answered Mar 10 '13 at 11:34

Andreas
380 3 6

still getting same erroneous editor warnings, as in Type 'std::thread' could not be resolved even though it compiles and executes OK – Scott Stensland May 31 '13 at 11:27

2 u need to compile once with the new settings, so the parser will "learn" that the std=c++11 flag is set – Andreas Jun 11 '13 at 9:16

depending on the compiler version it might be necessary to add -std=c++0x instead of -std=c++11 (e.g. debian wheezy) – xmoex Aug 19 '14 at 15:57

While the answers above didn't do the trick, this one did. – Minas Mina Jan 17 '15 at 13:29

I had the same problem on my Eclipse Juno. These steps solved the problem :

- Go to `Project -> Properties -> C/C++ General -> Path and Symbols -> Tab [Symbols]` .
- Add the symbol : **__cplusplus** with the value **201103L**

answered Mar 8 '14 at 17:21

Jerome
835 2 8 22

2 This should be accepted answer since it works also for Makefile projects! – omikron Dec 12 '14 at 10:02

@Jerk31 this is not working for Eclipse Juno running on windows(for me), I still see those complained un-necessary errors, despite the binary is working fine. Please help me!!! – overexchange Jan 6 '15 at 11:37

I'm really sorry I can't help you on Windows I don't have the opportunity to try to execute any C++ project on this OS. I wish you the best and good luck ! – Jerome Jan 7 '15 at 12:32

Doesn't work for me on Mars. – user1205577 Mar 6 '16 at 19:13

unresolved symbol is:

1. Go to Preferences->C/C++->Build->Settings
2. Select the Discovery tab
3. Select CDT GCC Built-in Compiler Settings [Shared]
4. Add the -std=c++11 to the "Command to get the compiler specs:" field such as:

> ${COMMAND} -E -P -v -dD -std=c++11 ${INPUTS}

5. Ok and Rebuild Index for the project.

Adding `-std=c++11` to project `Properties/C/C++ Build->Settings->Tool Settings->GCC C++ Compiler->Miscellaneous->Other` Flags wasn't enough for Kepler, however it was enough for older versions such as Helios.

edited Sep 1 at 13:25
YSelf
2,168 1 9 15

answered Feb 24 '14 at 23:33
user3348915
111 1 2

For me in Neon 4.6.3 it also did the trick !! – Guy Avraham Dec 7 '17 at 12:11

I can't yet comment so am writing my own answer:

It's related to `__GXX_EXPERIMENTAL_CXX0X__` and it's valid for Eclipse Juno and CDT 8.x.

Some parts of this answer are already covered in other answers but I want it to be coherent.

To make it possible to build using stdc++11, one have to add specific flag for compiler. You can do that via project properties. To modify project properties `RMB` and `Project` it will look something like: `-c -fmessage-length=0 -std=c++11` . By adding -stdc++11 flag compiler (GCC) will declare `__GXX_EXPERIMENTAL_CXX0X__` by itself.

At this point you can build project using all the goodness of C++11.

The problem is that Eclipse has it's own parser to check for errors - that's why you're still getting all the nasty errors in Eclipse editor, while at the same time you can build and run project without any. There is a way to solve this problem by explicitly declaring `__GXX_EXPERIMENTAL_CXX0X__` flag for the project, one can do that (just like Carsten Greiner said): C/C++ General -> Paths and Symbols -> Symbols -> GNU C++. Click "Add..." and past `__GXX_EXPERIMENTAL_CXX0X__` (ensure to append and prepend two underscores) into "Name" and leave "Value" blank. And now is the extra part I wanted to cover in comment to the first answer, go to: C/C++ General -> Preprocessor Include Path Macros etc. -> Providers, and Select `CDT Managed Build Setting Entries` then click APPLY and go back to Entries tab, under GNU C++ there should be now CDT Managed Build Setting Entries check if inside there is defined `__GXX_EXPERIMENTAL_CXX0X__` if it is -> APPLY and rebuild index you should be fine at this point.

answered Feb 26 '13 at 20:23

ravwojdyla
2,306 1 10 11

I had several issues too (Ubuntu 13.04 64-bit, g++-4.8, eclipse Juno 3.8.1, CDT 6.0.0). A lot of things are mentioned above, sorry to repeat those, but additionally I had problems utilizing

```
std::thread
```

as part of c++11 (adding -pthread for

```
-std=c++11
```

flag for the GCC and G++ compilers. Click Apply.

For the linker, same window, Miscellaneous, Linker flags, added the

```
-pthread
```

flag. Shared library settings, Shared object name, add the

```
-Wl,--no-as-needed
```

flag too. Click Apply.

C/C++ General -> Paths and symbols -> Symbols TAB, GNU C++ selected, Add the

```
__GXX_EXPERIMENTAL_CXX0X__
```

(no value)

flag. Click Apply.

C/C++ General -> Preprocessor Include paths.. -> Providers tab : check

CDT GCC built-in Compiler Settings

and for "Command to get compiler specs", add the

```
-std=c++11
```

flag. Uncheck Share. Click Apply.

CDT Managages Build Setting Entries, check this too. Uncheck the two others. Click Apply.

Going back to the Entries tab, GNU C++ CDT Managages Build Setting Entries, you should now see your added

```
__GXX_EXPERIMENTAL_CXX0X__
```

entry.

That's it. When coding, typing

```
std::
```

can now auto-complete the thread class for instance, builds should work at runtime.

edited Nov 28 '16 at 1:01
Emile Bergeron
10k 4 38 67

answered Jan 25 '14 at 22:42
Ola Aronsson
61 4

It was the `Project -> Properties -> C/C++ Build -> Settings -> Miscellaneous` step that worked for me. Thanks a bunch :D – Jake88 Feb 14 '14 at 16:19

I don't know if it is only me, the highest ranked solution doesn't work for me, my eclipse version is just normal eclipse platform installed by using sudo apt-get install eclipse in Ubuntu But I found a solution which adopts method together from both the highest ranked solution and the second, what I did to make it work is described as below (Note that the other steps like creating a C++ project etc. is ignored for simplicity)

Once you have created the C++ project

(1) C/C++ General -> Paths and Symbols -> Symbols -> GNU C++. Click "Add..." and paste **GXX_EXPERIMENTAL_CXX0X** (ensure to append and prepend two underscores) into "Name" and leave "Value" blank.

(2) Under C/C++ Build (at project settings), find the Preprocessor Include Path and go to the Providers Tab. Deselect all except CDT GCC Builtin Compiler Settings. Then untag Share settings entries … . Add the option -std=c++11 to the text box called Command to get compiler specs

After performed above 2 and 2 only steps, it works, the eclipse is able to resolve the unique_ptr, I don't know why this solution works, hope that it

This also worked for me (eclipse neon). At first I didn't deselect the other providers, which still caused problems, but deselecting all other providers than "CDT GCC Built-in Compiler Settings" and "CDT User Settings" worked. Thanks! – zpon Sep 7 '16 at 13:11

For me on ***Eclipse Neon*** I followed ***Trismegistos*** answer here above , YET I also added an additional step:

- Go to project --> Properties --> C++ General --> Preprocessor Include paths,Macros etc. --> Providers --> CDT Cross GCC Built-in Compiler Settings, append the flag ***"-std=c++11"***

Hit apply and OK.

Cheers,

Guy.

edited Dec 18 '17 at 5:22

answered Dec 17 '17 at 22:03

Guy Avraham

1,280 2 17 30

- right-click the project and go to "Properties"
- C/C++ Build -> Settings -> Tool Settings -> GCC C++ Compiler -> Miscellaneous -> Other Flags. Put -lm at the end of other flags text box and OK.

answered Jul 25 '13 at 1:21

Sameera Chanaka

1

What version of Eclipse is this? – Braiam Jul 25 '13 at 1:45

Neither the hack nor the cleaner

providing any useful reason why. At least from the command line, I get reproducible results.

answered May 5 '14 at 16:36

Jerry Miller
602 5 10

To get support for C++14 in Eclipse Luna, you could do these steps:

- In `C++ General -> Preprocessor Include -> Providers -> CDT Cross GCC Built-in Compiler Settings` , add "-std=c++14"
- In `C++ Build -> Settings -> Cross G++ Compiler -> Miscellaneous` , add "-std=c++14"

Reindex your project and eventually restart Eclipse. It should work as expected.

answered Nov 20 '14 at 15:43

Jerome
835 2 8 22

Eclipse C/C++ does not recognize the symbol `std::unique_ptr` even though you have included the C++11 memory header in your file.

Assuming you are using the GNU C++ compiler, this is what I did to fix:

Project -> Properties -> C/C++ General -> Preprocessor Include Paths -> GNU C++ -> CDT User Setting Entries

1. Click on the "Add..." button
2. Select "Preprocessor Macro" from the dropdown menu

   ```
   Name: __cplusplus     Value:
   ```

3. Hit Apply, and then OK to go back to your project
4. Then rebuild you C++ index: Projects -> C/C++ Index -> Rebuild

answered Mar 8 '16 at 23:28

Rubens Gomes
358 2 6

I solved it this way on a Mac. I used Homebrew to install the latest version of gcc/g++. They land in /usr/local/bin with includes in /usr/local/include.

I CD'd into /usr/local/bin and made a symlink from g++@7whatever to just g++ cause that @ bit is annoying.

Then I went to MyProject -> Properties -> C/C++ Build -> Settings -> GCC C++ Compiler and changed the command from "g++" to "/usr/local/bin/g++". If you decide not to make the symbolic link, you can be more specific.

Do the same thing for the linker.

Apply and Apply and Close. Let it rebuild the index. For a while, it showed a daunting number of errors, but I think that was while building indexes. While I was figuring out the errors, they all disappeared without further action.

I think without verifying that you could also go into Eclipse -> Properties -> C/C++ -> Core Build Toolchains and edit those with different paths, but I'm not sure what that will do.

answered Apr 3 at 17:37

Joseph Larson
461 3 9

protected by Community ♦
Jun 18 '15 at 17:35

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 reputation on this site (the association bonus does not count).

Would you like to answer one of