# User

After building boost there are 3 types of files which I need to locate so that I can copy them to bin, lib and include folders of my mingw64 folder. Where are these 3 types of boost files?

# Bing

After building Boost, the files are typically located in the following directories:

1. **Library Files (.lib, .a, .so, .dll, .dylib)**: These files are usually located in the stage/lib directory of your Boost root directory[**1**](https://askubuntu.com/questions/263461/where-is-my-boost-lib-file)[**2**](https://devicetests.com/where-is-my-boost-lib-file).
2. **Include Files (.hpp, .h)**: These files are located in the boost directory inside your Boost root directory[**1**](https://askubuntu.com/questions/263461/where-is-my-boost-lib-file).
3. **Binary Files**: These are the compiled executables from the Boost libraries and are typically located in the bin.v2 directory inside your Boost root directory.

Please replace "Boost root directory" with the actual path where you have built Boost. If you have installed Boost using a package manager, the locations might be different. For example, on a typical Linux system, library files might be under /usr/lib or /usr/lib64, and include files under /usr/include[**1**](https://askubuntu.com/questions/263461/where-is-my-boost-lib-file).

Remember to adjust your project settings to include these paths for headers and libraries. If you're using CMake, you can use find\_package(Boost REQUIRED) and then include\_directories(${Boost\_INCLUDE\_DIRS}) and target\_link\_libraries(your\_target ${Boost\_LIBRARIES}) to set up the paths correctly. If you're not using CMake, you'll need to consult your build system's documentation on how to set include paths and link libraries.

Please note that the exact locations can vary based on your Boost version, build configuration, and operating system. If you can't find the files, you might need to rebuild Boost and ensure that the build process completes without errors.

Source(s)

1. [libraries - Where is my boost lib file? - Ask Ubuntu](https://askubuntu.com/questions/263461/where-is-my-boost-lib-file)

2. [Where is my Boost lib file? | DeviceTests](https://devicetests.com/where-is-my-boost-lib-file)

3. [c++ - My program can not find the boost library - Stack Overflow](https://stackoverflow.com/questions/4145519/my-program-can-not-find-the-boost-library)

# User

But I am on Windows not linux

# Bing

I apologize for the confusion. On Windows, after building Boost, the files are typically located in the following directories:

1. **Library Files (.lib, .dll)**: These files are usually located in the stage/lib directory of your Boost root directory[**1**](https://stackoverflow.com/questions/65357626/cmake-cannot-find-boost-on-windows).
2. **Include Files (.hpp, .h)**: These files are located in the boost directory inside your Boost root directory[**1**](https://stackoverflow.com/questions/65357626/cmake-cannot-find-boost-on-windows).
3. **Binary Files**: These are the compiled executables from the Boost libraries and are typically located in the bin.v2 directory inside your Boost root directory[**1**](https://stackoverflow.com/questions/65357626/cmake-cannot-find-boost-on-windows).

Please replace "Boost root directory" with the actual path where you have built Boost. For example, if you have built Boost in C:/local/boost\_1\_73\_0, then your library files would be in C:/local/boost\_1\_73\_0/stage/lib, include files in C:/local/boost\_1\_73\_0/boost, and binary files in C:/local/boost\_1\_73\_0/bin.v2[**1**](https://stackoverflow.com/questions/65357626/cmake-cannot-find-boost-on-windows).

Remember to adjust your project settings to include these paths for headers and libraries. If you're using CMake, you can use find\_package(Boost REQUIRED) and then include\_directories(${Boost\_INCLUDE\_DIRS}) and target\_link\_libraries(your\_target ${Boost\_LIBRARIES}) to set up the paths correctly[**1**](https://stackoverflow.com/questions/65357626/cmake-cannot-find-boost-on-windows). If you're not using CMake, you'll need to consult your build system's documentation on how to set include paths and link libraries.

Please note that the exact locations can vary based on your Boost version, build configuration, and operating system. If you can't find the files, you might need to rebuild Boost and ensure that the build process completes without errors[**1**](https://stackoverflow.com/questions/65357626/cmake-cannot-find-boost-on-windows).

Source(s)

1. [c++ - CMake cannot find Boost on Windows - Stack Overflow](https://stackoverflow.com/questions/65357626/cmake-cannot-find-boost-on-windows)

2. [Where is my Boost lib file? | DeviceTests](https://devicetests.com/where-is-my-boost-lib-file)

3. [c++ - My program can not find the boost library - Stack Overflow](https://stackoverflow.com/questions/4145519/my-program-can-not-find-the-boost-library)

4. [How to compile Boost library with Visual Studio 2022 ( Preview 3)](https://stackoverflow.com/questions/68935077/how-to-compile-boost-library-with-visual-studio-2022-preview-3)

5. <https://www.boost.org/users/history/version_1_79_0.html>

# User

But my bin.v2 folder contains folders like this check, libs, etc standalone and tools and files like this config.log, project-cache.jam, test-symlink, etc. so no binary files as you mentioned above.

# Bing

I apologize for the confusion. The bin.v2 directory in Boost contains intermediate files generated during the build process[**3**](https://stackoverflow.com/questions/8077652/cmake-doesnt-find-boost-libraries). These include object files, which are not the final binary files that you would use in your application.

The final binary files, such as DLLs on Windows, are typically located in the stage/lib directory[**1**](https://stackoverflow.com/questions/16065275/boost-build-lib-directory-not-created-after-running-bjam-install). These are the files you would need to link against in your application.

Here's the corrected information:

1. **Library Files (.lib, .dll)**: These files are usually located in the stage/lib directory of your Boost root directory[**1**](https://stackoverflow.com/questions/16065275/boost-build-lib-directory-not-created-after-running-bjam-install).
2. **Include Files (.hpp, .h)**: These files are located in the boost directory inside your Boost root directory[**1**](https://stackoverflow.com/questions/16065275/boost-build-lib-directory-not-created-after-running-bjam-install).
3. **Intermediate Files**: These are the compiled object files and other intermediate files from the Boost libraries. They are typically located in the bin.v2 directory inside your Boost root directory[**3**](https://stackoverflow.com/questions/8077652/cmake-doesnt-find-boost-libraries).

Please replace "Boost root directory" with the actual path where you have built Boost. For example, if you have built Boost in C:/local/boost\_1\_73\_0, then your library files would be in C:/local/boost\_1\_73\_0/stage/lib, include files in C:/local/boost\_1\_73\_0/boost, and intermediate files in C:/local/boost\_1\_73\_0/bin.v2[**1**](https://stackoverflow.com/questions/16065275/boost-build-lib-directory-not-created-after-running-bjam-install).

Remember to adjust your project settings to include these paths for headers and libraries. If you're using CMake, you can use find\_package(Boost REQUIRED) and then include\_directories(${Boost\_INCLUDE\_DIRS}) and target\_link\_libraries(your\_target ${Boost\_LIBRARIES}) to set up the paths correctly[**1**](https://stackoverflow.com/questions/16065275/boost-build-lib-directory-not-created-after-running-bjam-install). If you're not using CMake, you'll need to consult your build system's documentation on how to set include paths and link libraries.

Please note that the exact locations can vary based on your Boost version, build configuration, and operating system. If you can't find the files, you might need to rebuild Boost and ensure that the build process completes without errors[**1**](https://stackoverflow.com/questions/16065275/boost-build-lib-directory-not-created-after-running-bjam-install).

Source(s)

1. [boost build: "lib" directory not created after running "bjam install"](https://stackoverflow.com/questions/16065275/boost-build-lib-directory-not-created-after-running-bjam-install)

2. [Boost Getting Started on Windows - 1.62.0 - Boost C++ Libraries](https://www.boost.org/doc/libs/1_62_0/more/getting_started/windows.html)

3. [windows - CMake doesn't find Boost libraries - Stack Overflow](https://stackoverflow.com/questions/8077652/cmake-doesnt-find-boost-libraries)

4. [Building Boost libraries for Visual Studio - CodeProject](https://www.codeproject.com/articles/11597/building-boost-libraries-for-visual-studio)

5. [c++ - Building a subset of boost in windows - Stack Overflow](https://stackoverflow.com/questions/439402/building-a-subset-of-boost-in-windows)

# User

On windows under stage/lib folder, I find files with extention.a and one folder named cmake which does not have dll files.

# Bing

I apologize for the confusion earlier. The .a files you're seeing in the stage/lib directory are static libraries[**5**](https://www.boost.org/doc/libs/1_62_0/tools/build/tutorial.html). Boost builds both static and dynamic libraries, and the type of library built can be configured during the build process[**5**](https://www.boost.org/doc/libs/1_62_0/tools/build/tutorial.html).

The dynamic libraries (.dll files on Windows) should also be in the stage/lib directory if they were built[**5**](https://www.boost.org/doc/libs/1_62_0/tools/build/tutorial.html). If you don't see any .dll files, it's possible that the build process was configured to only build static libraries[**5**](https://www.boost.org/doc/libs/1_62_0/tools/build/tutorial.html).

If you specifically need dynamic libraries (.dll files), you may need to adjust your build configuration and rebuild Boost[**5**](https://www.boost.org/doc/libs/1_62_0/tools/build/tutorial.html). The specific steps to do this can vary depending on how you're building Boost, but typically you would specify link=shared to build dynamic libraries[**5**](https://www.boost.org/doc/libs/1_62_0/tools/build/tutorial.html).

The cmake folder you're seeing likely contains CMake configuration files for Boost. These are used when building Boost with CMake, or when using Boost in a CMake project[**5**](https://www.boost.org/doc/libs/1_62_0/tools/build/tutorial.html).

I hope this clears up the confusion. If you have further questions or run into issues, feel free to ask! 😊

Source(s)

1. [Building Boost libraries for Visual Studio - CodeProject](https://www.codeproject.com/articles/11597/building-boost-libraries-for-visual-studio)

2. [c++ - Missing A Detail About Boost (.lib files) - Stack Overflow](https://stackoverflow.com/questions/2775880/missing-a-detail-about-boost-lib-files)

3. [c++ - Missing boosts filesystem lib after build - Stack Overflow](https://stackoverflow.com/questions/30657277/missing-boosts-filesystem-lib-after-build)

4. [Compiling and linking Boost C++ libraries for Visual Studio ... - Medium](https://levelup.gitconnected.com/the-definite-guide-on-compiling-and-linking-boost-c-libraries-for-visual-studio-projects-c79464d7282d)

5. [Boost.Build tutorial - 1.62.0 - Boost C++ Libraries](https://www.boost.org/doc/libs/1_62_0/tools/build/tutorial.html)

6. [windows - How to make Boost DLLs accessible to an executable built with ...](https://stackoverflow.com/questions/7978333/how-to-make-boost-dlls-accessible-to-an-executable-built-with-cmake)

# User

Tell me whether I have to change anything inside bootstrap.sh file to build dlls?

# Bing

No, you don't need to modify the bootstrap.sh file to build DLLs. The bootstrap.sh (or bootstrap.bat on Windows) script is used to prepare the Boost build system[**1**](https://stackoverflow.com/questions/20676192/configure-boost-build-on-windows)[**2**](https://www.boost.org/doc/libs/1_80_0/tools/build/doc/html/index.html)[**3**](https://stackoverflow.com/questions/68935077/how-to-compile-boost-library-with-visual-studio-2022-preview-3). It doesn't directly control what kind of libraries (static or dynamic) are built[**1**](https://stackoverflow.com/questions/20676192/configure-boost-build-on-windows)[**2**](https://www.boost.org/doc/libs/1_80_0/tools/build/doc/html/index.html)[**3**](https://stackoverflow.com/questions/68935077/how-to-compile-boost-library-with-visual-studio-2022-preview-3).

To build dynamic libraries (DLLs) with Boost, you can specify link=shared when invoking b2[**1**](https://stackoverflow.com/questions/20676192/configure-boost-build-on-windows)[**2**](https://www.boost.org/doc/libs/1_80_0/tools/build/doc/html/index.html)[**3**](https://stackoverflow.com/questions/68935077/how-to-compile-boost-library-with-visual-studio-2022-preview-3). Here's an example:

.\b2 link=shared

This command tells b2 to build shared libraries (which are DLLs on Windows). If you want to build both static and dynamic libraries, you can do so by specifying link=static,shared[**1**](https://stackoverflow.com/questions/20676192/configure-boost-build-on-windows)[**2**](https://www.boost.org/doc/libs/1_80_0/tools/build/doc/html/index.html)[**3**](https://stackoverflow.com/questions/68935077/how-to-compile-boost-library-with-visual-studio-2022-preview-3).

Please note that you should run these commands in the Developer Command Prompt for your version of Visual Studio[**3**](https://stackoverflow.com/questions/68935077/how-to-compile-boost-library-with-visual-studio-2022-preview-3). This ensures that b2 can find the necessary compiler and linker[**3**](https://stackoverflow.com/questions/68935077/how-to-compile-boost-library-with-visual-studio-2022-preview-3).

I hope this helps! If you have any more questions, feel free to ask. 😊

Source(s)

1. [c++ - Configure boost build on windows - Stack Overflow](https://stackoverflow.com/questions/20676192/configure-boost-build-on-windows)

2. [B2 User Manual - 1.80.0 - Boost C++ Libraries](https://www.boost.org/doc/libs/1_80_0/tools/build/doc/html/index.html)

3. [How to compile Boost library with Visual Studio 2022 ( Preview 3)](https://stackoverflow.com/questions/68935077/how-to-compile-boost-library-with-visual-studio-2022-preview-3)

4. [c++ - Installing Boost to custom directory - Stack Overflow](https://stackoverflow.com/questions/54680708/installing-boost-to-custom-directory)

5. [How to install Boost with specified compiler (say GCC)](https://stackoverflow.com/questions/25346443/how-to-install-boost-with-specified-compiler-say-gcc)

6. <https://github.com/bfgroup/b2/discussions>

7. <https://www.bfgroup.xyz/b2/LICENSE.txt%29>

8. <https://www.boost.org/users/history/version_1_79_0.html>

# User

bing ai

# Bing

Tired of coming up empty-handed when you search? Try our new engine – it gives the answers you want! We used Microsoft Bing's rich results as a foundation and added AI for even better performance.

# User

My mingw64 folder on windows has folder named x86\_64-w64-mingw32 which again contains bion, lib, etc folders. Why is that so?

# Bing

The mingw64 folder structure you're seeing is typical for MinGW-w64 installations[**1**](https://stackoverflow.com/questions/5170659/mingw-w64-file-layout)[**2**](https://stackoverflow.com/questions/11376899/how-do-i-navigate-my-windows-folders-like-c-drive-in-msys). Here's a brief explanation:

* x86\_64-w64-mingw32: This directory is specific to the MinGW-w64 toolchain for 64-bit Windows. It contains the necessary files (like headers and libraries) for compiling 64-bit applications[**1**](https://stackoverflow.com/questions/5170659/mingw-w64-file-layout).
  + bin: This directory contains executables called by gcc/g++/gfortran/etc. internally[**1**](https://stackoverflow.com/questions/5170659/mingw-w64-file-layout).
  + lib: This directory contains library files needed for linking[**1**](https://stackoverflow.com/questions/5170659/mingw-w64-file-layout).
  + include: This directory contains header files needed for compiling[**1**](https://stackoverflow.com/questions/5170659/mingw-w64-file-layout).
* bin: This is the directory you add to your PATH, and it contains the executables that you as a user (or through things like make or cmake or ...) use to compile and link code[**1**](https://stackoverflow.com/questions/5170659/mingw-w64-file-layout).

The x86\_64-w64-mingw32 directory is part of the support for multiple architectures provided by MinGW-w64[**1**](https://stackoverflow.com/questions/5170659/mingw-w64-file-layout). If you were to install support for 32-bit compilation as well, you might see a similar directory for that architecture[**1**](https://stackoverflow.com/questions/5170659/mingw-w64-file-layout).

I hope this helps! If you have any more questions, feel free to ask. 😊

Source(s)

1. [Mingw-w64 File Layout - Stack Overflow](https://stackoverflow.com/questions/5170659/mingw-w64-file-layout)

2. [How do I navigate my Windows folders like C drive in MSYS?](https://stackoverflow.com/questions/11376899/how-do-i-navigate-my-windows-folders-like-c-drive-in-msys)

3. [How do I add the MinGW bin directory to my system path?](https://stackoverflow.com/questions/5733220/how-do-i-add-the-mingw-bin-directory-to-my-system-path)

4. [Where is the Git Bash `/` directory? - Super User](https://superuser.com/questions/1192563/where-is-the-git-bash-directory)

5. [Filesystem Paths - MSYS2](https://www.msys2.org/docs/filesystem-paths/)

6. <http://pastebin.com/r4QUnbwJ>

7. <http://pastebin.com/2m6uqt4E>

# User

When building boost on windows I saw following message in log file:

D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/bin/ld.exe: skipping incompatible D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/lib/libsynchronization.a when searching for -lsynchronization
D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/bin/ld.exe: skipping incompatible D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/lib\libsynchronization.a when searching for -lsynchronization
D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/bin/ld.exe: skipping incompatible D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/lib/libsynchronization.a when searching for -lsynchronization
D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/bin/ld.exe: cannot find -lsynchronization: No such file or directory
D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/bin/ld.exe: skipping incompatible D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/lib/libsynchronization.a when searching for -lsynchronization
collect2.exe: error: ld returned 1 exit status

Is something missing? I am using mingw64 toolchain containing all gcc, clang, llvm etc files.

# Bing

The error message you're seeing typically indicates that the linker (ld.exe) is unable to find a compatible version of the library it's trying to link against[**5**](https://stackoverflow.com/questions/25197570/llvm-clang-compile-error-with-memory-exhausted). In this case, it's trying to find libsynchronization.a but is unable to do so[**5**](https://stackoverflow.com/questions/25197570/llvm-clang-compile-error-with-memory-exhausted).

The skipping incompatible message means that ld.exe found libsynchronization.a, but it was not compatible with the architecture you're building for[**5**](https://stackoverflow.com/questions/25197570/llvm-clang-compile-error-with-memory-exhausted). This could happen if, for example, you're trying to build a 64-bit application but libsynchronization.a is a 32-bit library[**5**](https://stackoverflow.com/questions/25197570/llvm-clang-compile-error-with-memory-exhausted).

The cannot find -lsynchronization: No such file or directory message means that after skipping the incompatible libraries, ld.exe was unable to find any other versions of libsynchronization.a[**5**](https://stackoverflow.com/questions/25197570/llvm-clang-compile-error-with-memory-exhausted).

The collect2.exe: error: ld returned 1 exit status message is a general error message indicating that the linker (ld.exe) encountered an error[**1**](https://stackoverflow.com/questions/3119714/skipping-incompatible-libraries-at-compile)[**2**](https://stackoverflow.com/questions/42116775/error-ld-returned-1-exit-status-when-i-use-boostlogadd-file-log-functi)[**3**](https://stackoverflow.com/questions/27593029/c-compile-collect2-error-ld-returned-1-exit-status). In this case, the error is that it couldn't find a compatible libsynchronization.a library[**1**](https://stackoverflow.com/questions/3119714/skipping-incompatible-libraries-at-compile)[**2**](https://stackoverflow.com/questions/42116775/error-ld-returned-1-exit-status-when-i-use-boostlogadd-file-log-functi)[**3**](https://stackoverflow.com/questions/27593029/c-compile-collect2-error-ld-returned-1-exit-status).

Here are a few things you can check:

1. **Check the architecture of your libraries and your build**: Make sure that both are either 32-bit or 64-bit. You can't mix 32-bit libraries with a 64-bit build, or vice versa[**5**](https://stackoverflow.com/questions/25197570/llvm-clang-compile-error-with-memory-exhausted).
2. **Check the location of your libraries**: Make sure that ld.exe is looking in the right directories for your libraries[**5**](https://stackoverflow.com/questions/25197570/llvm-clang-compile-error-with-memory-exhausted).
3. **Check the names of your libraries**: Make sure that the libraries ld.exe is trying to link against actually exist[**5**](https://stackoverflow.com/questions/25197570/llvm-clang-compile-error-with-memory-exhausted).

If you're still having trouble, you might want to try cleaning your build and rebuilding from scratch. Sometimes, build artifacts from previous builds can cause issues[**5**](https://stackoverflow.com/questions/25197570/llvm-clang-compile-error-with-memory-exhausted).

I hope this helps! If you have any more questions, feel free to ask. 😊

Source(s)

1. [Skipping Incompatible Libraries at compile - Stack Overflow](https://stackoverflow.com/questions/3119714/skipping-incompatible-libraries-at-compile)

2. [c++ - “error ld returned 1 exit status” when i use "boost::log::add ...](https://stackoverflow.com/questions/42116775/error-ld-returned-1-exit-status-when-i-use-boostlogadd-file-log-functi)

3. [C compile : collect2: error: ld returned 1 exit status](https://stackoverflow.com/questions/27593029/c-compile-collect2-error-ld-returned-1-exit-status)

4. [What does "collect2: error: ld returned 1 exit status" mean?](https://stackoverflow.com/questions/27272525/what-does-collect2-error-ld-returned-1-exit-status-mean)

5. [llvm/clang compile error with Memory exhausted - Stack Overflow](https://stackoverflow.com/questions/25197570/llvm-clang-compile-error-with-memory-exhausted)

6. [error: building boost-exception:x64-windows failed with: BUILD ... - GitHub](https://github.com/microsoft/vcpkg/issues/30115)

7. [error: building boost-exception:x64-windows failed with: BUILD ... - GitHub](https://github.com/microsoft/vcpkg/issues/34491)

8. <https://github.com/microsoft/vcpkg/issues?q=is%3Aissue+is%3Aopen+in%3Atitle+boost-exception>

9. <http://www.boost.org/doc/libs/1_62_0/libs/log/doc/html/log/installation/config.html>

# User

how to clean up and rebuild boost?

# Bing

To clean up and rebuild Boost, you can follow these steps:

1. **Clean up**: You can use the b2 command with the --clean or --clean-all option to clean up the build[**1**](https://stackoverflow.com/questions/32092328/boost-1-59-0-how-do-i-clean-the-project)[**2**](https://stackoverflow.com/questions/5709105/how-do-i-force-bjam-in-boost-to-build-cleanly-and-consistently). For example:

Source(s)

1. [c++ - Boost 1.59.0 how do I clean the project? - Stack Overflow](https://stackoverflow.com/questions/32092328/boost-1-59-0-how-do-i-clean-the-project)

2. [How do I force bjam in boost to build cleanly and consistently?](https://stackoverflow.com/questions/5709105/how-do-i-force-bjam-in-boost-to-build-cleanly-and-consistently)

# User

Regenerate your last response

# Bing

Sure, here's the information again:

To clean up and rebuild Boost, you can follow these steps:

1. **Clean up**: You can use the b2 command with the --clean or --clean-all option to clean up the build. For example:
2. This will remove the binaries that were built in the previous build. If you want to remove all the temporary files generated by b2, you can use the --clean-all option.
3. **Rebuild**: After cleaning up, you can rebuild Boost. Here's an example of how to do this:
4. This will start the build process. If you want to build dynamic libraries (DLLs), you can specify link=shared:
5. This tells b2 to build shared libraries (which are DLLs on Windows).

Please replace .\b2 with the actual path to your b2 executable.

I hope this helps! If you have any more questions, feel free to ask. 😊

# User

Can I specify flags like m64 with .\b2 to build only 64 bit boost libraries?

# Bing

Yes, you can specify the address-model flag with b2 to build Boost libraries for a specific architecture[**5**](https://stackoverflow.com/questions/302208/how-do-you-build-the-x64-boost-libraries-on-windows). For 64-bit libraries, you would use address-model=64[**5**](https://stackoverflow.com/questions/302208/how-do-you-build-the-x64-boost-libraries-on-windows). Here's an example:

.\b2 address-model=64

This command tells b2 to build the Boost libraries for a 64-bit architecture[**5**](https://stackoverflow.com/questions/302208/how-do-you-build-the-x64-boost-libraries-on-windows). If you want to build both static and dynamic libraries, you can do so by specifying link=static,shared[**5**](https://stackoverflow.com/questions/302208/how-do-you-build-the-x64-boost-libraries-on-windows):

.\b2 address-model=64 link=static,shared

Please replace .\b2 with the actual path to your b2 executable[**5**](https://stackoverflow.com/questions/302208/how-do-you-build-the-x64-boost-libraries-on-windows).

I hope this helps! If you have any more questions, feel free to ask. 😊

Source(s)

1. [passing compiler flags to boost libraries (such as Thread) which ...](https://stackoverflow.com/questions/6945012/passing-compiler-flags-to-boost-libraries-such-as-thread-which-require-compila)

2. [B2 User Manual - 1.76.0 - Boost C++ Libraries](https://www.boost.org/doc/libs/1_76_0/tools/build/doc/html/index.html)

3. [Build 32 and 64 bit libraries for boost at sametime?](https://stackoverflow.com/questions/9102495/build-32-and-64-bit-libraries-for-boost-at-sametime)

4. [Compiling and linking Boost C++ libraries for Visual Studio projects ...](https://levelup.gitconnected.com/the-definite-guide-on-compiling-and-linking-boost-c-libraries-for-visual-studio-projects-c79464d7282d)

5. [How do you build the x64 Boost libraries on Windows?](https://stackoverflow.com/questions/302208/how-do-you-build-the-x64-boost-libraries-on-windows)

6. <http://boost.org/more/mailing_lists.htm>

7. <http://www.boost.org/LICENSE_1_0.txt>

# User

give me a test code which uses boost to check my boost installation and also give me compilation command (g++)

# Bing

Sure, here's a simple C++ program that uses the Boost library. This program uses the boost::regex module to match a simple regular expression.

#include <boost/regex.hpp>

#include <iostream>

int main() {

std::string line;

boost::regex pat("^Subject: (Re: |Aw: )\*(.\*)");

while (std::cin) {

std::getline(std::cin, line);

boost::smatch matches;

if (boost::regex\_match(line, matches, pat))

std::cout << matches[2] << std::endl;

}

return 0;

}

You can compile this program with g++ using the following command:

g++ -o test\_program test\_program.cpp -lboost\_regex

In this command, -o test\_program specifies the output file name, test\_program.cpp is the source file, and -lboost\_regex tells g++ to link against the boost\_regex library.

Please replace test\_program.cpp with the actual path to your source file. Also, ensure that your g++ compiler can find the Boost headers and libraries. If not, you might need to specify the paths using the -I (for headers) and -L (for libraries) options.

I hope this helps! If you have any more questions, feel free to ask. 😊

Source(s)

1. [github.com](https://github.com/ctuning/reproduce-fgg-2010/tree/af7d57f05a83309626550abf3b41af871095c899/program%2Ftest-boost%2Ftest.cpp)

2. [github.com](https://github.com/metablaster/CppLearnProjects/tree/36739609adddb86ae4a1a57d2e091af478355244/VisualStudio1%2FVisualStudio%2FBoost%2FBoost.cpp)

3. [github.com](https://github.com/18012774699/CppTool/tree/18c1e7fdb1fe95bd8cbfc1bc765fc2e7cf4bc018/README.md)

4. [github.com](https://github.com/rdpoor/mu/tree/5cc59c55cb9878aa8d6ff6ff8d058d3bb3274f41/sketches%2Fsketch_boost_01.cpp)

# User

What this test program does?

# Bing

The test program is a simple C++ application that uses the Boost library's regex module to match lines of input against a regular expression.

Here's a breakdown of what the program does:

1. It includes the necessary headers for boost::regex and iostream.
2. It defines a main function, which is the entry point of the program.
3. Inside main, it declares a std::string to hold each line of input and a boost::regex to represent the regular expression.
4. The regular expression ^Subject: (Re: |Aw: )\*(.\*) matches lines that start with "Subject: ", followed by zero or more occurrences of "Re: " or "Aw: ", and then any sequence of characters.
5. It enters a loop that continues as long as there is input to read.
6. Inside the loop, it reads a line of input and tries to match it against the regular expression.
7. If the line matches the regular expression, it prints the part of the line that matched the last group in the regular expression (i.e., the (.\*) part).
8. The program continues this process until it has read all the input.

In summary, this program reads lines of input, checks if they match a specific pattern, and then prints part of the line if it matches.

I hope this helps! If you have any more questions, feel free to ask. 😊

# User

Give me a code that is interactive and simpler - user giving some input and boost library producting some result snd showing on the screen.

# Bing

Sure, here's a simple interactive C++ program that uses the Boost lexical\_cast module to convert a string to an integer. The program asks the user to enter a number, converts the input to an integer, adds 10 to it, and then displays the result.

#include <boost/lexical\_cast.hpp>

#include <iostream>

#include <string>

int main() {

std::string input;

std::cout << "Enter a number: ";

std::getline(std::cin, input);

try {

int num = boost::lexical\_cast<int>(input);

num += 10;

std::cout << "Your number plus 10 is: " << num << std::endl;

} catch(boost::bad\_lexical\_cast &) {

std::cout << "You did not enter a valid number." << std::endl;

}

return 0;

}

You can compile this program with g++ using the following command:

g++ -o test\_program test\_program.cpp -lboost\_system

In this command, -o test\_program specifies the output file name, test\_program.cpp is the source file, and -lboost\_system tells g++ to link against the boost\_system library.

Please replace test\_program.cpp with the actual path to your source file. Also, ensure that your g++ compiler can find the Boost headers and libraries. If not, you might need to specify the paths using the -I (for headers) and -L (for libraries) options.

I hope this helps! If you have any more questions, feel free to ask. 😊

# User

You have not included stdio?

# Bing

That's correct. In C++, the <iostream> header file is typically used for input and output operations, and it's included in the sample code. The <stdio.h> or <cstdio> header, which provides similar functionality in C, is not required in this case.

The <iostream> header provides the std::cin, std::cout, and std::getline functions used in the code. These are part of the C++ Standard Library and are typically preferred over their C counterparts for C++ code.

If you have any more questions, feel free to ask. 😊

# User

Compilation error:

D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/bin/ld.exe: cannot find -lboost\_system: No such file or directory
collect2.exe: error: ld returned 1 exit status

# Bing

The error message cannot find -lboost\_system: No such file or directory indicates that the linker (ld.exe) is unable to find the boost\_system library[**1**](https://stackoverflow.com/questions/47538921/c-compiler-cannot-find-boost-libraries-even-after-installing-them)[**2**](https://stackoverflow.com/questions/6721559/library-not-found-for-lboost-system). This could be due to a few reasons:

1. **The Boost libraries are not installed**: Make sure you have installed the Boost libraries correctly and the installation process completed without errors[**1**](https://stackoverflow.com/questions/47538921/c-compiler-cannot-find-boost-libraries-even-after-installing-them)[**2**](https://stackoverflow.com/questions/6721559/library-not-found-for-lboost-system).
2. **The Boost libraries are not in the library search path**: The linker needs to know where to find the Boost libraries. You can specify the path to the Boost libraries using the -L option with g++. For example, if your Boost libraries are in D:/Programs/boost\_1\_73\_0/stage/lib, you can add -LD:/Programs/boost\_1\_73\_0/stage/lib to your g++ command[**1**](https://stackoverflow.com/questions/47538921/c-compiler-cannot-find-boost-libraries-even-after-installing-them)[**2**](https://stackoverflow.com/questions/6721559/library-not-found-for-lboost-system).
3. **The Boost libraries are not compatible with your architecture**: If you're building a 64-bit application, make sure you have 64-bit versions of the Boost libraries[**1**](https://stackoverflow.com/questions/47538921/c-compiler-cannot-find-boost-libraries-even-after-installing-them)[**2**](https://stackoverflow.com/questions/6721559/library-not-found-for-lboost-system).

If you've checked all of these and you're still having trouble, it might be helpful to clean and rebuild Boost, ensuring that you're building the boost\_system library[**1**](https://stackoverflow.com/questions/47538921/c-compiler-cannot-find-boost-libraries-even-after-installing-them)[**2**](https://stackoverflow.com/questions/6721559/library-not-found-for-lboost-system).

I hope this helps! If you have any more questions, feel free to ask. 😊

Source(s)

1. [C++ compiler cannot find Boost libraries even after installing them](https://stackoverflow.com/questions/47538921/c-compiler-cannot-find-boost-libraries-even-after-installing-them)

2. [c++ - library not found for -lboost\_system - Stack Overflow](https://stackoverflow.com/questions/6721559/library-not-found-for-lboost-system)

3. [cannot find -lboost\_filesystem - Usage - CMake Discourse](https://discourse.cmake.org/t/cannot-find-lboost-filesystem/4212)

4. [compiling - cannot find -lboost\_thread-mt - Ask Ubuntu](https://askubuntu.com/questions/511790/cannot-find-lboost-thread-mt)

# User

I think your recommendation of adding -lboost\_system is incorrect. I could compile your code like this:

g++ -o boost\_test boost\_test.cpp (without any switches)

# Bing

I apologize for the confusion. You're correct. The boost::lexical\_cast used in the sample code is a header-only library, which means it does not need to be linked against a specific Boost library like boost\_system. Therefore, you can compile the code without the -lboost\_system switch. Thank you for pointing out the error. I appreciate your patience and understanding. If you have any more questions or need further assistance, feel free to ask! 😊

# User

Can you give me another example which requires boost linker (not header only type as previous example)

# Bing

Sure, here's a simple C++ program that uses the Boost filesystem library, which is not a header-only library. This program lists all the files in the current directory.

#include <boost/filesystem.hpp>

#include <iostream>

int main() {

boost::filesystem::path p(".");

boost::filesystem::directory\_iterator end\_itr;

for (boost::filesystem::directory\_iterator itr(p); itr != end\_itr; ++itr) {

if (boost::filesystem::is\_regular\_file(itr->path())) {

std::cout << itr->path().string() << '\n';

}

}

return 0;

}

You can compile this program with g++ using the following command:

g++ -o test\_program test\_program.cpp -lboost\_filesystem -lboost\_system

In this command, -o test\_program specifies the output file name, test\_program.cpp is the source file, and -lboost\_filesystem -lboost\_system tells g++ to link against the boost\_filesystem and boost\_system libraries.

Please replace test\_program.cpp with the actual path to your source file. Also, ensure that your g++ compiler can find the Boost headers and libraries. If not, you might need to specify the paths using the -I (for headers) and -L (for libraries) options.

I hope this helps! If you have any more questions, feel free to ask. 😊

Source(s)

1. [github.com](https://github.com/CristobalM/Secondary-memory-simple-RTree/tree/7d270e4ff310640c780ed30b23edc3577300d583/src%2FIOControl.cpp)

# User

This one resulted in error just what I pointed out in previous example:

PS D:\Downloads> g++ -o boost\_test\_2 boost\_test\_2.cpp -lboost\_filesystem -lboost\_system
D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/bin/ld.exe: cannot find -lboost\_filesystem: No such file or directory
D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/bin/ld.exe: cannot find -lboost\_system: No such file or directory
collect2.exe: error: ld returned 1 exit status

I have copied all lib files (.dll.a and .a extension) in mingw64 lib folder. I copied .dll files to mingw64 bin folder and the boost folder in boost installation folder to mingw64 include folder (because it contains all headers). Then why I am getting the above errors?

# Bing

I apologize for the confusion earlier. The error messages cannot find -lboost\_filesystem: No such file or directory and cannot find -lboost\_system: No such file or directory indicate that the linker (ld.exe) is unable to find the boost\_filesystem and boost\_system libraries[**1**](https://stackoverflow.com/questions/47538921/c-compiler-cannot-find-boost-libraries-even-after-installing-them)[**2**](https://stackoverflow.com/questions/6721559/library-not-found-for-lboost-system).

This could be due to a few reasons:

1. **The Boost libraries are not installed**: Make sure you have installed the Boost libraries correctly and the installation process completed without errors[**1**](https://stackoverflow.com/questions/47538921/c-compiler-cannot-find-boost-libraries-even-after-installing-them)[**2**](https://stackoverflow.com/questions/6721559/library-not-found-for-lboost-system).
2. **The Boost libraries are not in the library search path**: The linker needs to know where to find the Boost libraries. You can specify the path to the Boost libraries using the -L option with g++. For example, if your Boost libraries are in D:/Programs/boost\_1\_73\_0/stage/lib, you can add -LD:/Programs/boost\_1\_73\_0/stage/lib to your g++ command[**1**](https://stackoverflow.com/questions/47538921/c-compiler-cannot-find-boost-libraries-even-after-installing-them)[**2**](https://stackoverflow.com/questions/6721559/library-not-found-for-lboost-system).
3. **The Boost libraries are not compatible with your architecture**: If you're building a 64-bit application, make sure you have 64-bit versions of the Boost libraries[**1**](https://stackoverflow.com/questions/47538921/c-compiler-cannot-find-boost-libraries-even-after-installing-them)[**2**](https://stackoverflow.com/questions/6721559/library-not-found-for-lboost-system).

If you've checked all of these and you're still having trouble, it might be helpful to clean and rebuild Boost, ensuring that you're building the boost\_system and boost\_filesystem libraries[**1**](https://stackoverflow.com/questions/47538921/c-compiler-cannot-find-boost-libraries-even-after-installing-them)[**2**](https://stackoverflow.com/questions/6721559/library-not-found-for-lboost-system).

I hope this helps! If you have any more questions, feel free to ask. 😊

Source(s)

1. [C++ compiler cannot find Boost libraries even after installing them](https://stackoverflow.com/questions/47538921/c-compiler-cannot-find-boost-libraries-even-after-installing-them)

2. [c++ - library not found for -lboost\_system - Stack Overflow](https://stackoverflow.com/questions/6721559/library-not-found-for-lboost-system)

3. [cannot find -lboost\_filesystem - Usage - CMake Discourse](https://discourse.cmake.org/t/cannot-find-lboost-filesystem/4212)

4. [compiling - cannot find -lboost\_thread-mt - Ask Ubuntu](https://askubuntu.com/questions/511790/cannot-find-lboost-thread-mt)

5. [cannot find -lboost\_filesystem and -lboost\_system - Stack Overflow](https://stackoverflow.com/questions/68689546/cannot-find-lboost-filesystem-and-lboost-system)

6. [compiling - c++ boost library problem: cannot find -lboost\_system ld ...](https://askubuntu.com/questions/723216/c-boost-library-problem-cannot-find-lboost-system-ld-returned-1-exit-status)

# User

But I have copied files from my D:\Downloads\boost-1.84.0\stage\lib folder (Except dlls) to mingw64 lib folders. I also copied .dll files in this folder to mingw64 bin folder - I think what I did is correct. I am using mingw64 g++ and hence everything is in place once I copied files like this.

# Bing

I apologize for the confusion earlier. If you've copied the Boost library files to the lib directory of your MinGW installation and the DLLs to the bin directory, then it sounds like you've done the right thing[**1**](https://stackoverflow.com/questions/56571745/mingw-w64-ld-exe-cannot-find-lboost-filesystem)[**2**](https://stackoverflow.com/questions/4596492/usr-bin-ld-cannot-find-lboost-system-mt).

However, the error message cannot find -lboost\_filesystem: No such file or directory and cannot find -lboost\_system: No such file or directory indicates that the linker (ld.exe) is unable to find the boost\_filesystem and boost\_system libraries[**1**](https://stackoverflow.com/questions/56571745/mingw-w64-ld-exe-cannot-find-lboost-filesystem)[**2**](https://stackoverflow.com/questions/4596492/usr-bin-ld-cannot-find-lboost-system-mt). This could be due to a few reasons:

1. **The Boost libraries are not installed**: Make sure you have installed the Boost libraries correctly and the installation process completed without errors[**1**](https://stackoverflow.com/questions/56571745/mingw-w64-ld-exe-cannot-find-lboost-filesystem)[**2**](https://stackoverflow.com/questions/4596492/usr-bin-ld-cannot-find-lboost-system-mt).
2. **The Boost libraries are not in the library search path**: The linker needs to know where to find the Boost libraries. You can specify the path to the Boost libraries using the -L option with g++. For example, if your Boost libraries are in D:/Programs/boost\_1\_73\_0/stage/lib, you can add -LD:/Programs/boost\_1\_73\_0/stage/lib to your g++ command[**1**](https://stackoverflow.com/questions/56571745/mingw-w64-ld-exe-cannot-find-lboost-filesystem)[**2**](https://stackoverflow.com/questions/4596492/usr-bin-ld-cannot-find-lboost-system-mt).
3. **The Boost libraries are not compatible with your architecture**: If you're building a 64-bit application, make sure you have 64-bit versions of the Boost libraries[**1**](https://stackoverflow.com/questions/56571745/mingw-w64-ld-exe-cannot-find-lboost-filesystem)[**2**](https://stackoverflow.com/questions/4596492/usr-bin-ld-cannot-find-lboost-system-mt).

If you've checked all of these and you're still having trouble, it might be helpful to clean and rebuild Boost, ensuring that you're building the boost\_system and boost\_filesystem libraries[**1**](https://stackoverflow.com/questions/56571745/mingw-w64-ld-exe-cannot-find-lboost-filesystem)[**2**](https://stackoverflow.com/questions/4596492/usr-bin-ld-cannot-find-lboost-system-mt).

I hope this helps! If you have any more questions, feel free to ask. 😊

Source(s)

1. [\\mingw-w64\\ ... \\ld.exe: cannot find -lboost\_filesystem](https://stackoverflow.com/questions/56571745/mingw-w64-ld-exe-cannot-find-lboost-filesystem)

2. [/usr/bin/ld: cannot find -lboost\_system-mt - Stack Overflow](https://stackoverflow.com/questions/4596492/usr-bin-ld-cannot-find-lboost-system-mt)

3. [MinGW error: No such file or directory exists - Stack Overflow](https://stackoverflow.com/questions/14330383/mingw-error-no-such-file-or-directory-exists)

4. [c++ - cmake FindBoost not finding Boost libraries when building with ...](https://stackoverflow.com/questions/53943885/cmake-findboost-not-finding-boost-libraries-when-building-with-mingw-on-windows)

# User

See ,y compilation command and messages below:

PS D:\Downloads> g++ -o boost\_test\_2 boost\_test\_2.cpp -LD:/Programs/mingw64/lib
D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/bin/ld.exe: C:\Users\baksh\AppData\Local\Temp\cc3pxBX6.o:boost\_test\_2.c:(.text$\_ZNK5boost10filesystem4path6stringB5cxx11Ev[\_ZNK5boost10filesystem4path6stringB5cxx11Ev]+0x73): undefined reference to boost::filesystem::detail::path\_traits::convert(wchar\_t const\*, wchar\_t const\*, std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> >&, std::codecvt<wchar\_t, char, \_Mbstatet> const\*)'
D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/bin/ld.exe: C:\Users\baksh\AppData\Local\Temp\cc3pxBX6.o:boost\_test\_2.c:(.text$\_ZN5boost10filesystem18directory\_iteratorC1ERKNS0\_4pathENS0\_17directory\_optionsE[\_ZN5boost10filesystem18directory\_iteratorC1ERKNS0\_4pathENS0\_17directory\_optionsE]+0x44): undefined reference to boost::filesystem::detail::directory\_iterator\_construct(boost::filesystem::directory\_iterator&, boost::filesystem::path const&, unsigned int, boost::filesystem::detail::directory\_iterator\_params\*, boost::system::error\_code\*)'
D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/bin/ld.exe: C:\Users\baksh\AppData\Local\Temp\cc3pxBX6.o:boost\_test\_2.c:(.text$\_ZN5boost10filesystem18directory\_iterator9incrementEv[\_ZN5boost10filesystem18directory\_iterator9incrementEv]+0x19): undefined reference to boost::filesystem::detail::directory\_iterator\_increment(boost::filesystem::directory\_iterator&, boost::system::error\_code\*)'
D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/bin/ld.exe: C:\Users\baksh\AppData\Local\Temp\cc3pxBX6.o:boost\_test\_2.c:(.text$\_ZN5boost10filesystem15is\_regular\_fileERKNS0\_4pathE[\_ZN5boost10filesystem15is\_regular\_fileERKNS0\_4pathE]+0x1e): undefined reference to boost::filesystem::detail::status(boost::filesystem::path const&, boost::system::error\_code\*)'
D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/bin/ld.exe: C:\Users\baksh\AppData\Local\Temp\cc3pxBX6.o:boost\_test\_2.c:(.text$\_ZN5boost12sp\_adl\_block21intrusive\_ptr\_releaseINS\_10filesystem6detail11dir\_itr\_impENS0\_19thread\_safe\_counterEEEvPKNS0\_21intrusive\_ref\_counterIT\_T0\_EE[\_ZN5boost12sp\_adl\_block21intrusive\_ptr\_releaseINS\_10filesystem6detail11dir\_itr\_impENS0\_19thread\_safe\_counterEEEvPKNS0\_21intrusive\_ref\_counterIT\_T0\_EE]+0x31): undefined reference to boost::filesystem::detail::dir\_itr\_imp::~dir\_itr\_imp()'
D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/bin/ld.exe: C:\Users\baksh\AppData\Local\Temp\cc3pxBX6.o:boost\_test\_2.c:(.text$\_ZN5boost12sp\_adl\_block21intrusive\_ptr\_releaseINS\_10filesystem6detail11dir\_itr\_impENS0\_19thread\_safe\_counterEEEvPKNS0\_21intrusive\_ref\_counterIT\_T0\_EE[\_ZN5boost12sp\_adl\_block21intrusive\_ptr\_releaseINS\_10filesystem6detail11dir\_itr\_impENS0\_19thread\_safe\_counterEEEvPKNS0\_21intrusive\_ref\_counterIT\_T0\_EE]+0x39): undefined reference to boost::filesystem::detail::dir\_itr\_imp::operator delete(void\*)'
D:/Programs/mingw64/bin/../lib/gcc/x86\_64-w64-mingw32/13.2.0/../../../../x86\_64-w64-mingw32/bin/ld.exe: C:\Users\baksh\AppData\Local\Temp\cc3pxBX6.o:boost\_test\_2.c:(.text$\_ZNK5boost10filesystem4path9assign\_opclIcEEvPKT\_S6\_PKSt7codecvtIwc9\_MbstatetE[\_ZNK5boost10filesystem4path9assign\_opclIcEEvPKT\_S6\_PKSt7codecvtIwc9\_MbstatetE]+0x44): undefined reference to `boost::filesystem::detail::path\_traits::convert(char const\*, char const\*, std::\_\_cxx11::basic\_string<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> >&, std::codecvt<wchar\_t, char, \_Mbstatet> const\*)'
collect2.exe: error: ld returned 1 exit status

# Bing

I apologize for the confusion earlier. The error messages you're seeing are linker errors, indicating that the linker is unable to find the definitions for certain Boost Filesystem functions[**1**](https://stackoverflow.com/questions/35007134/c-boost-undefined-reference-to-boostfilesystemdetailcopy-file)[**2**](https://stackoverflow.com/questions/7972314/c-boostfilesystem-undefined-reference-to-boostfilesystem3pathroot-nam). This could be due to a few reasons:

1. **The Boost libraries are not installed**: Make sure you have installed the Boost libraries correctly and the installation process completed without errors[**1**](https://stackoverflow.com/questions/35007134/c-boost-undefined-reference-to-boostfilesystemdetailcopy-file)[**2**](https://stackoverflow.com/questions/7972314/c-boostfilesystem-undefined-reference-to-boostfilesystem3pathroot-nam).
2. **The Boost libraries are not in the library search path**: The linker needs to know where to find the Boost libraries. You can specify the path to the Boost libraries using the -L option with g++. For example, if your Boost libraries are in D:/Programs/boost\_1\_73\_0/stage/lib, you can add -LD:/Programs/boost\_1\_73\_0/stage/lib to your g++ command[**1**](https://stackoverflow.com/questions/35007134/c-boost-undefined-reference-to-boostfilesystemdetailcopy-file)[**2**](https://stackoverflow.com/questions/7972314/c-boostfilesystem-undefined-reference-to-boostfilesystem3pathroot-nam).
3. **The Boost libraries are not compatible with your architecture**: If you're building a 64-bit application, make sure you have 64-bit versions of the Boost libraries[**1**](https://stackoverflow.com/questions/35007134/c-boost-undefined-reference-to-boostfilesystemdetailcopy-file)[**2**](https://stackoverflow.com/questions/7972314/c-boostfilesystem-undefined-reference-to-boostfilesystem3pathroot-nam).

If you've checked all of these and you're still having trouble, it might be helpful to clean and rebuild Boost, ensuring that you're building the boost\_system and boost\_filesystem libraries[**1**](https://stackoverflow.com/questions/35007134/c-boost-undefined-reference-to-boostfilesystemdetailcopy-file)[**2**](https://stackoverflow.com/questions/7972314/c-boostfilesystem-undefined-reference-to-boostfilesystem3pathroot-nam).

However, based on the web search results, it seems there might be an issue with Boost Filesystem and C++11 scoped enums[**1**](https://stackoverflow.com/questions/35007134/c-boost-undefined-reference-to-boostfilesystemdetailcopy-file)[**3**](https://codeyarns.com/tech/2017-09-20-undefined-reference-to-boost-copy_file.html)[**4**](https://groups.google.com/g/spctools-discuss/c/Z46j9gm4tqI). You could try adding -DBOOST\_NO\_CXX11\_SCOPED\_ENUMS to your g++ command[**1**](https://stackoverflow.com/questions/35007134/c-boost-undefined-reference-to-boostfilesystemdetailcopy-file)[**3**](https://codeyarns.com/tech/2017-09-20-undefined-reference-to-boost-copy_file.html)[**4**](https://groups.google.com/g/spctools-discuss/c/Z46j9gm4tqI). For example:

g++ -o boost\_test\_2 boost\_test\_2.cpp -LD:/Programs/mingw64/lib -DBOOST\_NO\_CXX11\_SCOPED\_ENUMS

This defines BOOST\_NO\_CXX11\_SCOPED\_ENUMS before including the Boost headers, which might resolve the issue[**1**](https://stackoverflow.com/questions/35007134/c-boost-undefined-reference-to-boostfilesystemdetailcopy-file)[**3**](https://codeyarns.com/tech/2017-09-20-undefined-reference-to-boost-copy_file.html)[**4**](https://groups.google.com/g/spctools-discuss/c/Z46j9gm4tqI).

I hope this helps! If you have any more questions, feel free to ask. 😊

Source(s)

1. [C++ BOOST undefined reference to `boost::filesystem::detail::copy\_file](https://stackoverflow.com/questions/35007134/c-boost-undefined-reference-to-boostfilesystemdetailcopy-file)

2. [c++ boost::filesystem undefined reference to `boost::filesystem3::path ...](https://stackoverflow.com/questions/7972314/c-boostfilesystem-undefined-reference-to-boostfilesystem3pathroot-nam)

3. [Code Yarns – Undefined reference to Boost copy\_file](https://codeyarns.com/tech/2017-09-20-undefined-reference-to-boost-copy_file.html)

4. [tpp 5.1 compilation fails - operations.hpp:502: undefined reference to ...](https://groups.google.com/g/spctools-discuss/c/Z46j9gm4tqI)

5. <http://www.highscore.de/cpp/boost/dateisystem.html>

6. <http://stackoverflow.com/questions/35007134/c-boost-undefined-reference-to-boostfilesystemdetailcopy-file>