

Compile MYSTRAN: Code::Blocks + gFortran

Instructions provided by Brian Esp (www.espcomposites.com and www.structruafea.com)

This document describes how to compile MYSTRAN with the Code::Blocks IDE and gFortran. This is the preferred build process for MYSTRAN. *The instructions are for Windows.* Please read the document thoroughly; some of the details can be subtle.

It will take about 15-30 minutes to go through the steps in this document, provided there are no errors. If there are errors, see the Troubleshooting section.

Download 1 – gFortran

gFortran (GNU Fortran) is an open source Fortran compiler. This document presents two options for downloading gFortran.

OPTION 1: (Currently preferred)

You can download a self extracting gFortran version from the following link. *This is a more direct approach than the one in OPTION 2.*

<http://www.equation.com/servlet/equation.cmd?fa=fortran>

Version 10.1.0 (64 bit) was tested and compiles with MYSTRAN (image with highlighted option below).

Three Last Releases	Self-Extracting File		Note
	32-bit	64-bit	
8.3.0	gcc-8.3.0-32.exe	gcc-8.3.0-64.exe	Including <i>gdb</i> ; <i>make</i> ; OpenMP; <i>icl</i> ; <i>neuLoop</i> ; <i>lalpe2</i>
9.2.0	gcc-9.2.0-32.exe	gcc-9.2.0-64.exe	Including <i>gdb</i> ; <i>make</i> ; OpenMP; <i>icl</i> ; <i>neuLoop</i> ; <i>lalpe2</i>
10.1.0	gcc-10.1.0-32.exe	gcc-10.1.0-64.exe	Including <i>gdb</i> ; <i>make</i> ; OpenMP; <i>icl</i> ; <i>neuLoop</i> ; <i>lalpe2</i>

OPTION 2: (Previously preferred)

The intent of OPTION 2 is **simply to install gFortran** using a **DIFFERENT** version of Code:Blocks than what you will use to compile MYSTRAN. The gfortran version from Code::Blocks 20.03 (following link) is 8.1.0 (2018) and can be determined by using "gfortran --version" in a CMD window.

If you don't have gFortran installed, select the proper Code::Blocks download (codeblocksXmingw-setup.exe). The entire objective of this download is to install gFortran. There is a section option as well, which is discussed later. By default, this Code::Blocks version will likely install gfortran.exe in the "C:\Program Files\CodeBlocks\MinGW\bin" directory.

Select the FossHUB or Sourceforge.net link associated with:
codeblocksXmingw-setup.exe



File	Date	Download from
codeblocks-20.03-setup.exe	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03-setup-nonadmin.exe	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03-nosetup.zip	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03mingw-setup.exe	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03mingw-nosetup.zip	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup.exe	02 Apr 2020	FossHUB or Sourceforge.net

The Code:Blocks download link for Windows:

<http://www.codeblocks.org/downloads/26>

Now Install Code::Blocks

The above Code::Blocks version is only used to install gFortran (you will not compile MYSTRAN with it). To compile MYSTRAN, you will need to use a separate Code::Blocks version (Download 2), which has as a patch.

Download 2

Code::Blocks 20.03 does not support projects with many files. This is because at the end it sends a string to the Fortran compiler via CMD line. If the string is too long (about 35k characters), it will give an error with Code:Blocks 20.03.

The developers of Code::Blocks have implemented a patch, but it is not currently part of the 20.03 build. However, a modified version of the “nightly build” can *be downloaded via the link below*, which contains a patch that will allow MYSTRAN to be compiled even if there are many files:

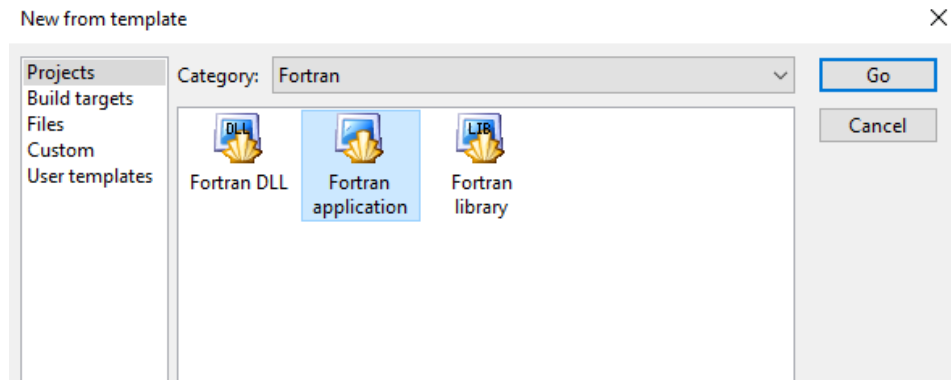
Download 2 Link:

https://www.mystran.com/misc/CB_6_12_2020.zip

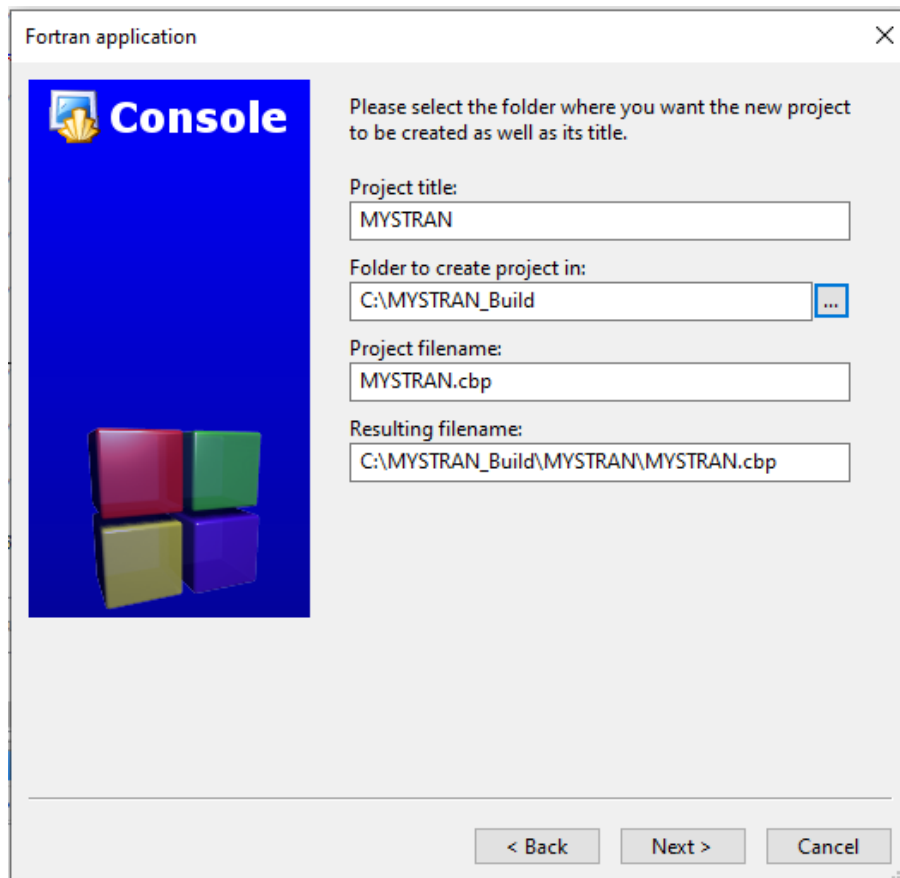
Unzip **Download 2** to a directory (this will contain codeblocks.exe in the root of the ZIP file). ***Now, run “codeblocks.exe” from the directory you extracted Download 2 to. You can think of this as a sort of “portable version” that does not require a full install. DO NOT RUN it from installation from Download 1 (if you used OPTION 1 - that will not have the required patch).***

NEXT: Creating the MYSTRAN Project

File→**New Project** and use the dropdown to select Fortran. Then select Fortran application and Go.

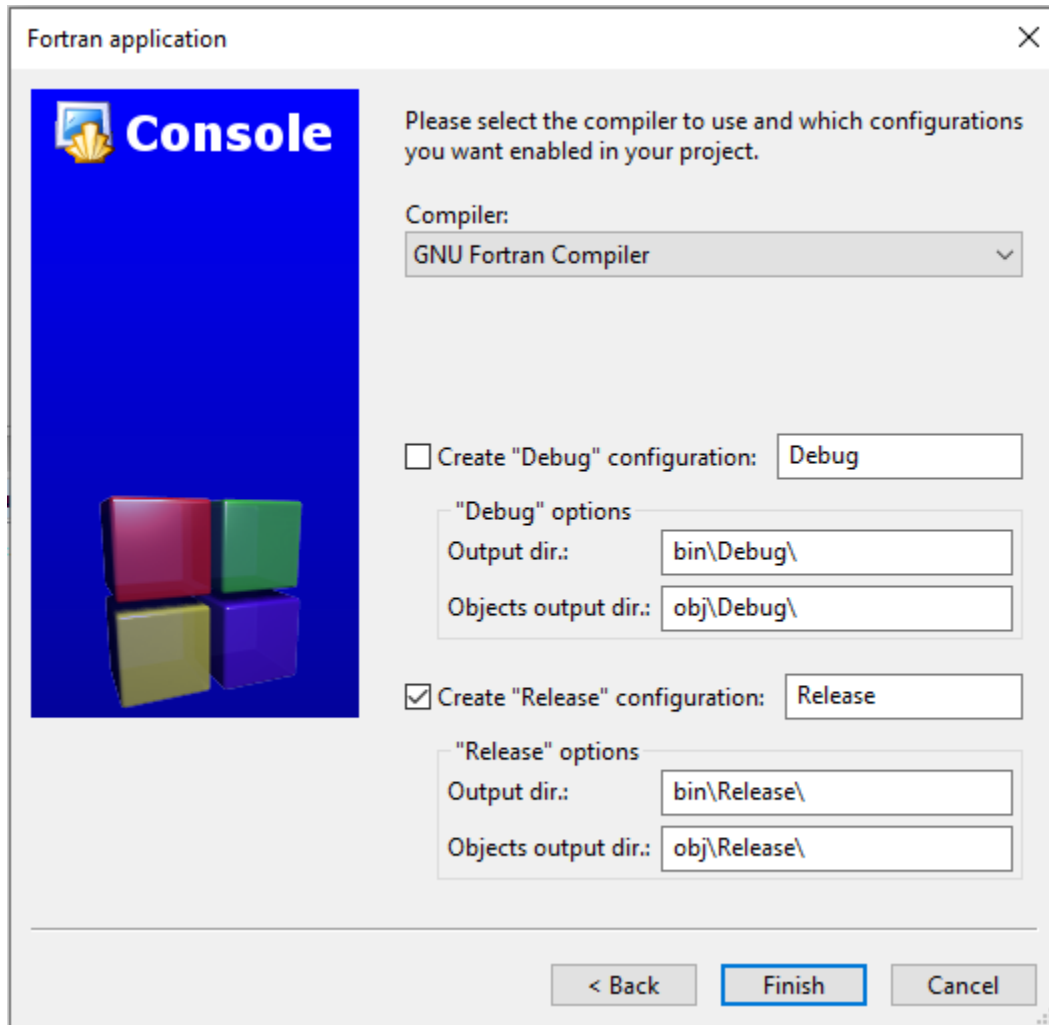


In the next screen, below is an example, but you can adjust the entries as desired. It is recommended to use “MYSTRAN” as the project name. Note that the folder to create the project in is not the same as the folder where the source code resides.



In the following menu, ensure **“GNU Fortran Compiler”** is selected. You can choose whether to use a Debug configuration or not, but you should have **“Create Release Configuration”** checked.

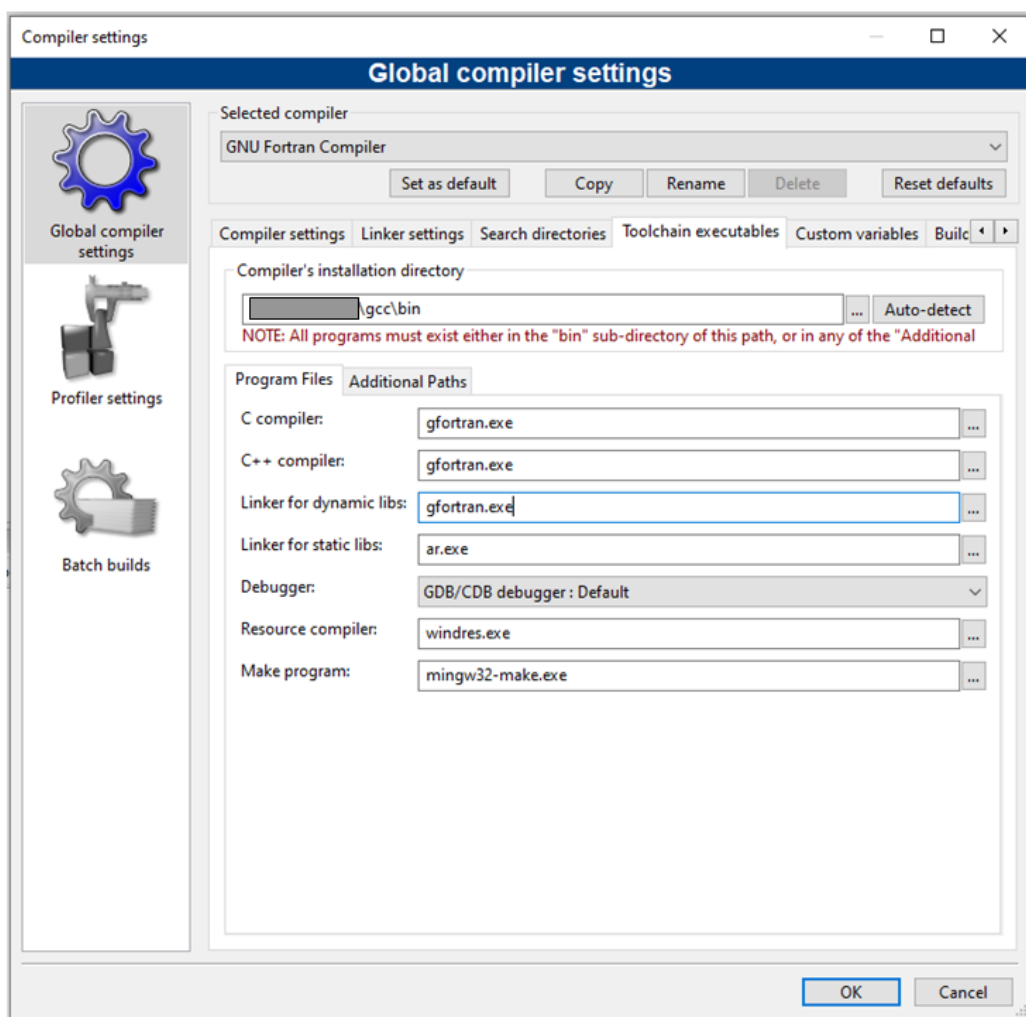
The Release configuration allows you to make the MYSTRAN.EXE. For the remainder of this document, only the **“Release”** option will be considered so as to reduce confusion.



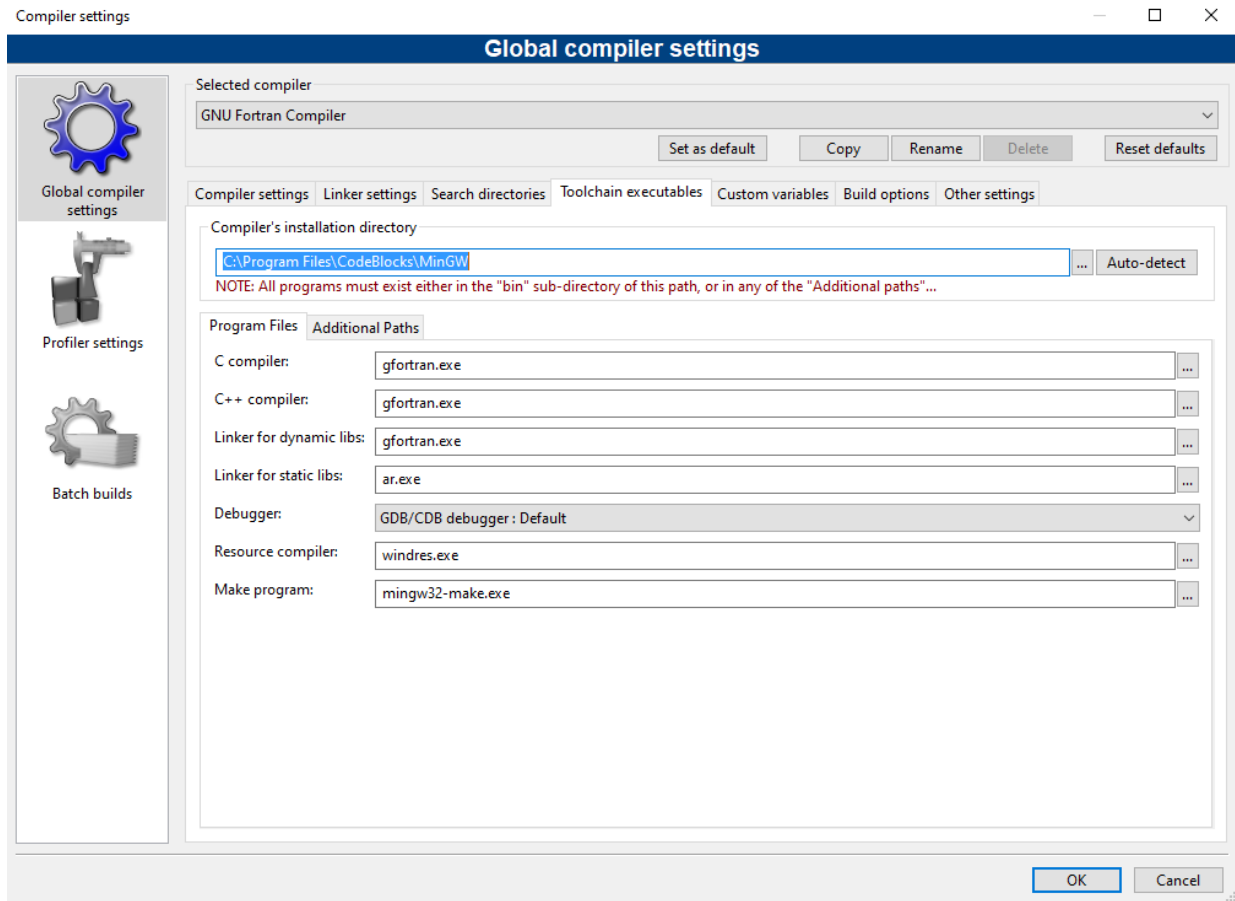
Next, ensure the compiler paths are correct. The path shown in the following screenshot is the default path is you chose Download 1 – OPTION 1. If gfortran.exe is in a different directory, then select that directory. For example, if you used Download 1 – OPTION 2, you would need to select a different path.

Settings→Compiler and select the **“Toolchain executable”** tab.

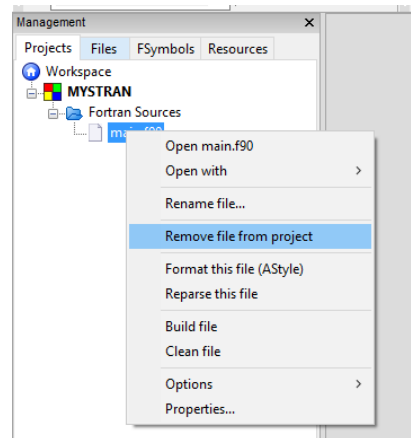
Make sure **“GNU Fortran Compiler”** is selected at the top. Also, don’t be concerned that it says **“C compiler”** and **“C++ Compiler”** in the label fields (these are just labels).



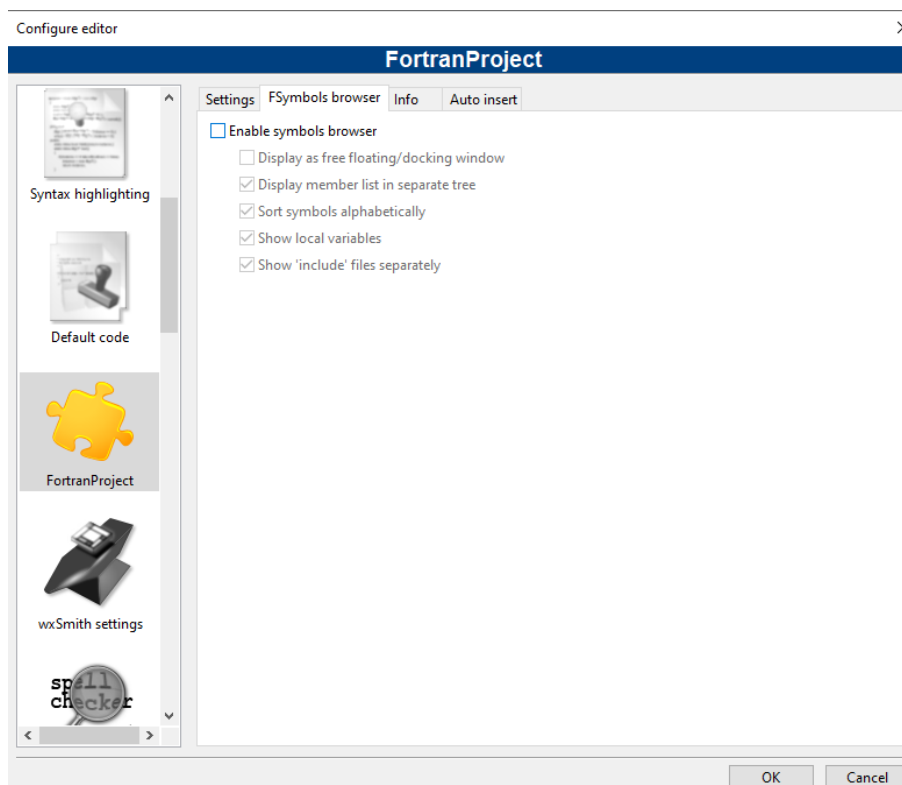
This is the default path if you used Download 1 - OPTION 2 to establish the Fortran compiler:



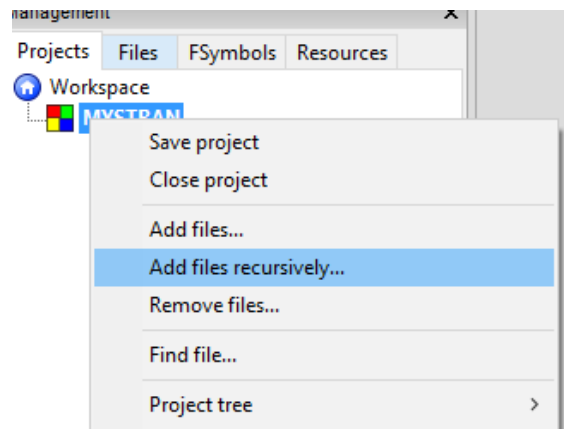
Next, remove the dummy main.f90 file



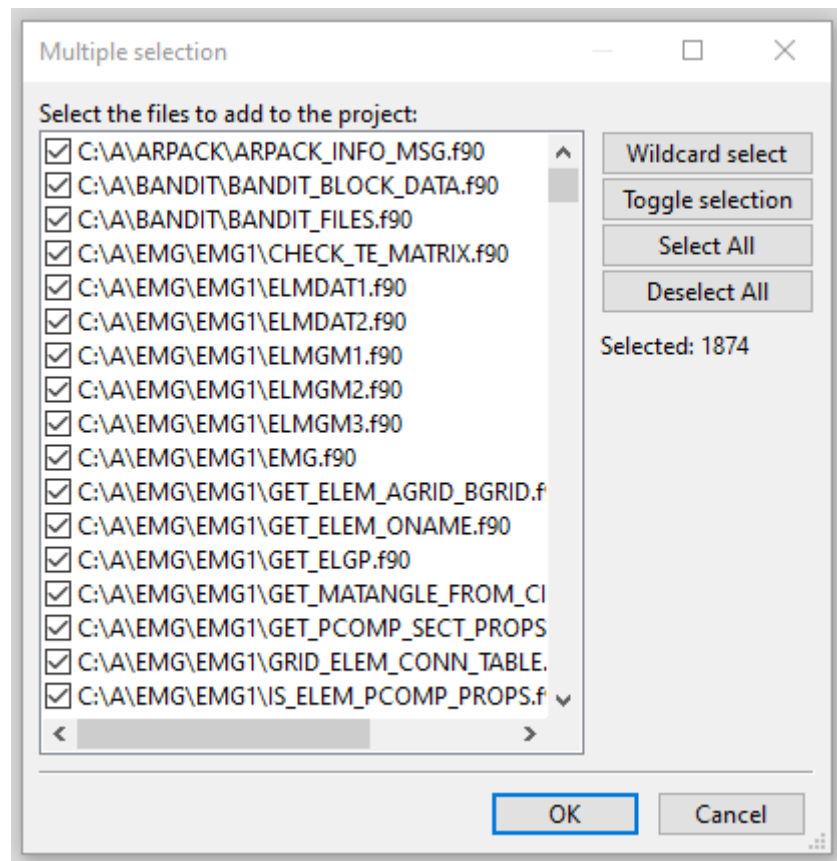
Before importing the files, use **Settings→Editor→Fortran Project→FSymbols Browser** and Disable the Symbols Browser. “Fortran Project” is located on the vertical slider menu. ***If you do not do not do this, it will take a very long time (10-40 minute) to add files in the next step.*** The long read in time problem happens because the symbol browser is updated on every added file. If desired, Symbols can be reactivated after reading in the files.



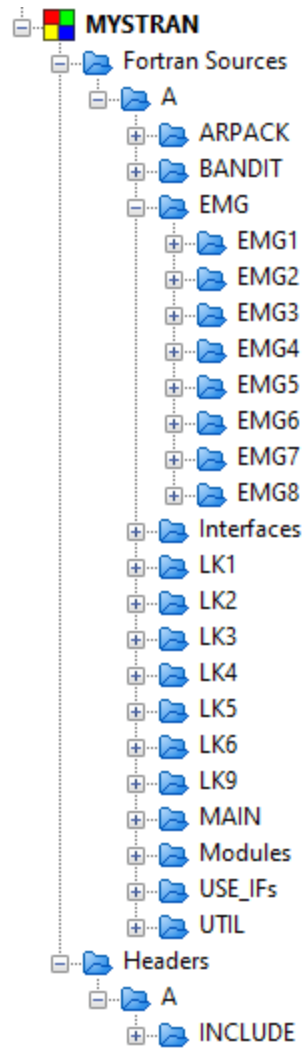
For this example, the files from MYSTRAN ZIP package (directory of “a-Windows-version”) are copied to the “C:\A” directory for convenience. To import the source code, right click on MYSTRAN and add the source code via “Add files recursively”. ***If this process takes more than a minute or two, see the prior step.***



You will see a menu like this (which should have about 1800+ files). Click OK.



After adding the files, the tree will look something like this after it is expanded (where C:\A is where the source files are located). C:\A is just an example path for the location where the source code is on the computer.



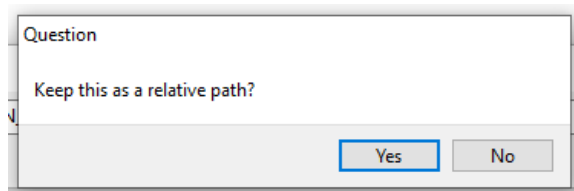
The two INCLUDE files must be addressed (debug.h and stat.h).

Go to **Project**→**Build Options**

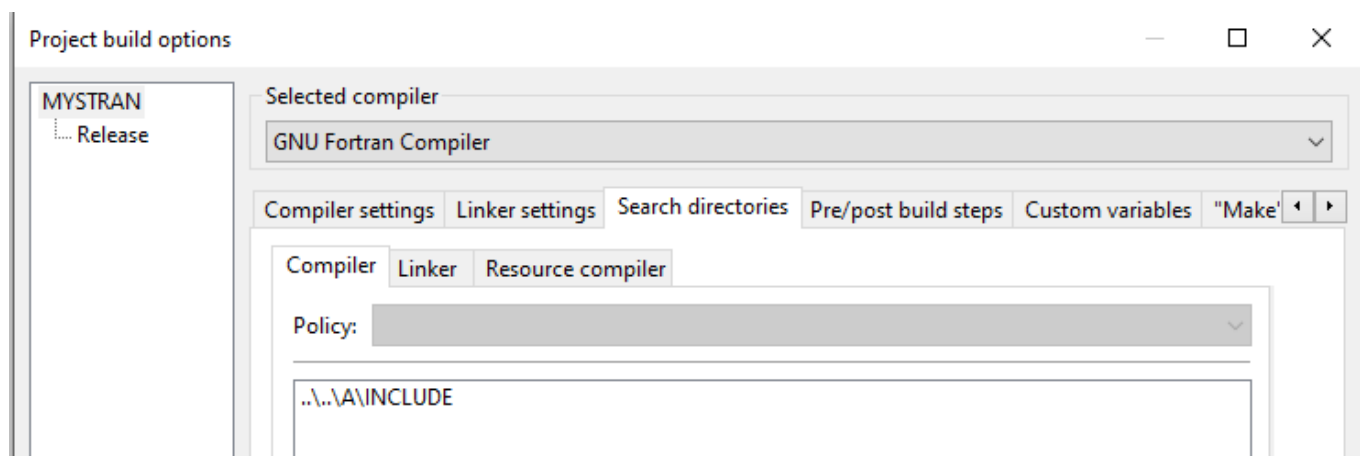
Make sure that **MYSTRAN** is selected on the upper left hand side. This will automatically cause the Release (below it) to have the same option.

Make sure **“GNU Fortran Compiler”** is selected

In the **“Search directories”** tab, click **“Add”** and select the path where the **“INCLUDE”** folder resides from the source files. Answer **“Yes”** when it asks if you want to keep it as a relative path. Then select **“OK”**.



For this example, the selected path on the computer is **“C:\A\INCLUDE”**, which is converted to **“..\..\A\INCLUDE”** in the following image. You will see the path as shown below:



The next step will allow MYSTRAN.EXE to be a static file and not have dependencies to external DLL files.

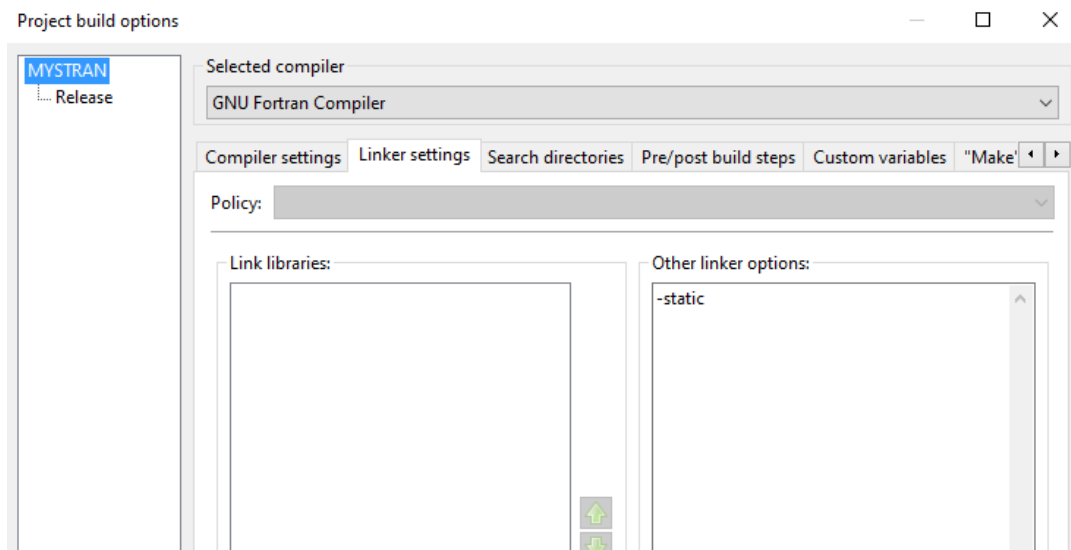
Go to **Project→Build Options**

Make sure that **MYSTRAN** is selected on the upper left hand side. This will automatically cause the Release (below it) to have the same option.

Make sure **“GNU Fortran Compiler”** is selected

Select the **“Linker settings”** tab

Manually type in **“-static”** in the **“Other linker options:”** area



If the above process is not performed, libwinpthread-1.dll, libgcc-s-seh-1.dll, libquadmath-0.dll will need to be added to the MYSTRAN.EXE folder. These files are in subdirectories of C:\Program Files\CodeBlocks\MinGW, provided you installed the CodeBlocks version with MinGW. Also, if you did not perform the above process and you have environment variables set to the DLL files (may have occurred from prior efforts), you may not need to have the DLL files in the same directory as MYSTRAN.EXE. However, if you place the MYSTRAN.EXE on another computer, it may not run. So it is highly recommended to use the above process to static link the DLL files.

NOTE: Code::Blocks appears to have a bug. After you successfully build MYSTRAN.EXE and close C::B, then reopen, it may not build again. You may see an error like the following image. If this occurs, select Settings→Compiler and select the “Toolchain executable” tab. This will “refresh” the location of the Fortran compiler.

Finally, use **Build→Rebuild**

There will be various warnings (in blue) that can be disregarded.

In the event that there was an error, see the “Troubleshooting” section below.

If there was no error, **MYSTRAN.EXE** will then be present in the **bin\Release** subfolder where you created the project (unless you changed the default path at the beginning or in **Build Properties→Build Targets→Output Filename**).

Within the MYSTRAN ZIP package on GitHub, there are various example DAT files (in the “3 – Runs” directory). You can test the build via command line by typing “mystran X.dat” where X.dat is the sample run and mystran.exe and X.dat are in the same directory.

NOTE: After saving the project, you can add the Symbols Browser back if you want via: **Settings→Editor→Fortran Project→FSymbols Browser** and Disable the Symbols Browser.

Troubleshooting

Problem 1. When building, if you get an error that says something like "...PENTIUM II...", this likely means the Fortran compiler is not linked properly. There are some possible reasons for this:

A – The compiler path and executable is not properly set in Toolchain Executables. The process for this was discussed earlier in this document.

B – See the **"Awareness"** section of this document (after this section) to make sure you do not have the compiler linked under "Debug", but not at the "Release" level or the top "MYSTRAN" level.

C – There appears to be a bug in Code::Blocks, which sometimes causes a linking error. I have noticed this when opening a saved project and trying to build it, without making any other modifications. In my case, the simple fix was to just go into the "Toolchain Executables" menu and then leave the menu. This seems to refresh it.

D – Ensure that the Fortran plugin is installed. It was discovered that if this plugin was not installed, there could be an error.

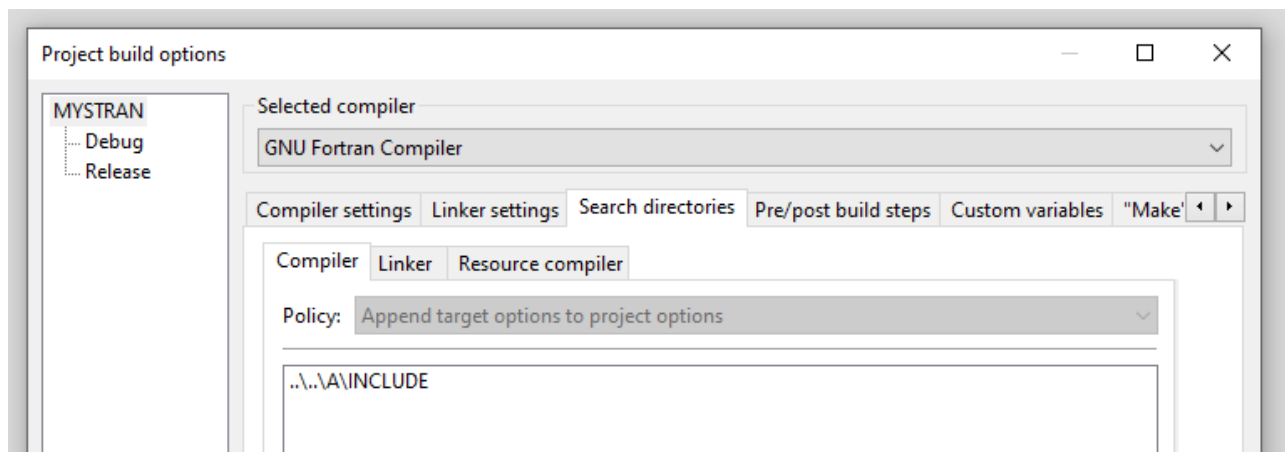
E – Save the project and then reopen Code::Blocks. Use **Build→Rebuild**. If that does not work, use option C again as way to "refresh" it.

F – In **Project→Build Options**, on the top right, select Release (below MYSTRAN). Ensure this says "GNU Fortran Compiler". It is possible to accidentally override this to another compiler, even if the compiler under MYSTRAN is "GNU Fortran Compiler". In other words, be sure that you did not override the compiler path.

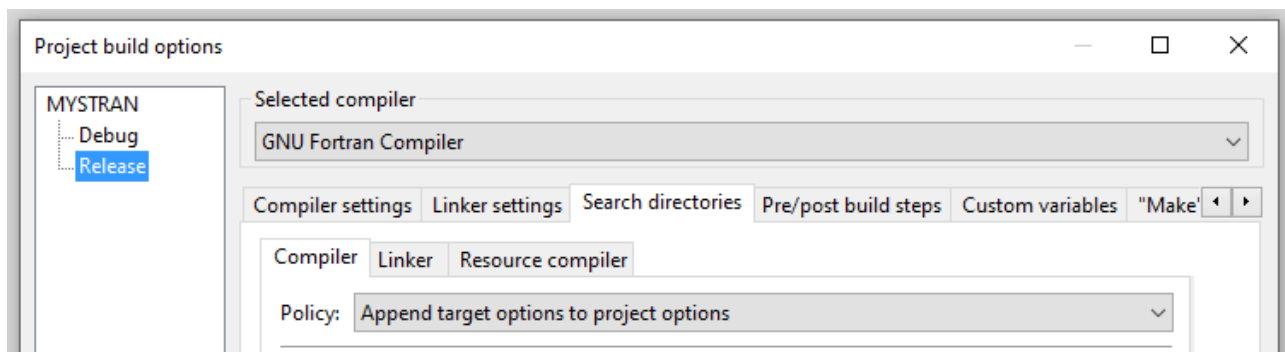
Problem 2. If there is an error related to debug.h or stat.h (the INCLUDE files), ensure that the INCLUDE paths are set as stated prior in this document.

Awareness

When in **Project→Build Options**, you can select the top level, which is MYSTRAN in this case (as shown below). This is relevant to “Search Directories” and “Other Linker Options” settings. The options chosen here will apply to both “Debug” and “Release” configurations (a “Debug” configuration was part of this project). For example, in the image below, the C:\A\INCLUDE directory will be used for both Debug and Release, even if nothing is entered in the specific Debug and Release configurations when selecting them individually.



You can also “fine tune” the Debug and Release configurations by selecting them individually. This is not necessary for MYSTRAN, but is demonstrated to reduce the chance of inadvertent entries. For example, if you added the INCLUDE path to the Debug configuration (but not at the top level MYSTRAN or the Release configuration), you would get an error when attempting to Build/Rebuild.



Makefile

It is possible to generate a Makefile in Code::Blocks. However, the Makefile can only be used on Linux.

Rather, it is recommended to use the process with CMake (discussed in the MYSTRAN forum).