Cygwin version

The most recent version of the Cygwin DLL is **2.10.0**. (in 2018)

Installing Cygwin

Install Cygwin by running setup-x86_64.exe (64-bit installation) or setup-x86.exe (32-bit installation)

Use the setup program to perform a fresh install or to update an existing installation.

Configuring NetBeans IDE 8.0 for C/C++/Fortran

- If you want to work with C, C++, or Fortran programs in the NetBeans IDE, your NetBeans installation must include the C/C++ plugin.
- Although the plugin is named C/C++, it also includes support for Fortran programming.
- Third party C/C++ compilers, make utilities, and debuggers are also required.

If you have NetBeans IDE 8.0 but do not know if you have the C/C++ plugin

To determine if your NetBeans IDE includes the C/C++ plugin already :

- Select File > New Project. If C/C++ is listed as one of the Categories, you have the C/C++ plugin module. You should skip to the section Installing and Setting Up the Compilers and Tools.

If you have NetBeans IDE 8.0 without the C/C++ plugin

If your NetBeans IDE does not show a C/C++ project category when you select File > New Project, complete the following steps to add the C/C++ plugin module to the IDE.

1. If your network uses a proxy, choose Tools > Options > General in the IDE, select Manual Proxy Settings, enter the HTTP Proxy and Port for your proxy, and click OK.
2. Choose Tools > Plugins.
3. In the Plugins dialog box, click the Available Plugins tab, and scroll to the C/C++ category.
4. Select the C/C++ checkbox and click Install to start the NetBeans IDE Installer.
5. In the NetBeans IDE Installer, click Next.
6. Read the license agreement, select the checkbox to accept the terms of the license agreement, and click Next. Click Install.
7. After the installation completes, select either Restart IDE Now or Restart IDE Later and click Finish.

Installing and Setting Up the Compilers and Tools

Microsoft Windows : The NetBeans C/C++ module has been tested with compilers from Cygwin and MinGW. If you install both Cygwin and MinGW, be careful to keep their installation locations completely separate and do not mix tools from Cygwin and MinGW in one tool collection in the IDE.

Cygwin Compilers and Tools

The NetBeans C/C++ module has been tested with the following compilers and tools from Cygwin.com.

Software or Resource	Version Tested	Description
cygwin1.dll	1.7.7, 1.7.9	Cygwin Linux-like environment for Windows
gcc	4.3.4	Cygwin C compiler
g++	4.3.4	Cygwin C++ compiler
gdb	6.8	Cygwin GNU debugger
make	3.81	Cygwin make utility

If you already have the Cygwin gcc and g++ compilers, GNU make, and gdb debugger installed on your Windows system and your path is set up correctly to find them, make sure that you have the correct versions.
To check the versions of your Cygwin compilers and tools:
Check the version of Cygwin environment by typing the following commands at a Windows command prompt: `C:\> cygcheck -c cygwin`
Check the versions of the Cygwin gcc and g++ compilers, make, and gdb by typing the following commands at a Windows command prompt:

```
C:\> gcc –version          C:\> g++ --version     C:\> make –version          C:\> gdb --version
```

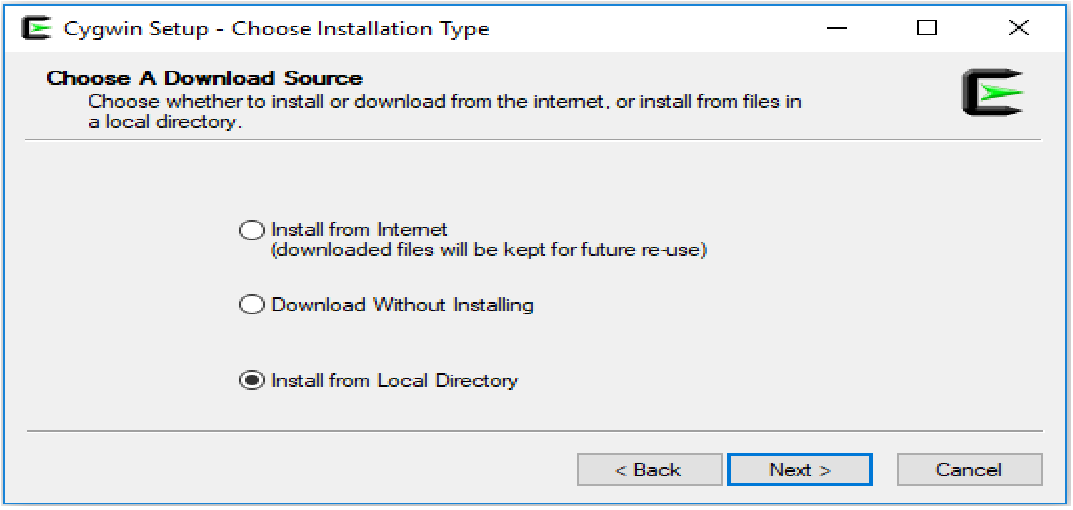
If you have the correct versions, then no further setup is necessary.

To install the GNU gcc and g++ compilers, make, and gdb debugger from cygwin.com:

1. Download the Cygwin setup-x86.exe (32-bit installation) or setup-x86_64.exe (64-bit installation) program by clicking Install Cygwin in the left navigation bar, or by clicking the direct setup-x86.exe or setup-x86_64.exe link in Cygwin site.
2. Run the downloaded Cygwin installer. Accept the defaults until you reach the Select Your Internet Connection page. Select the option on this page that is best for you (direct connection) . Click Next.
3. On the Choose Download Site page, choose a download site you think might be relatively close to you. Click Next.
4. On the Select Packages page you select the packages to download. We can search for Cygwin, base, gcc, g++, gdb and make one by one and download. Click the + next to Devel to expand the development tools category. You may want to resize the window so you can see more of it at one time.
5. Select each package you want to download by clicking the Skip label next to it, which reveals the version number of the package to download. At a minimum, select
   - gcc-core: C compiler
   - gcc-g++: C++ compiler
   - gdb: The GNU Debugger
   - make: the GNU version of the 'make' utility

Packages that are required by the packages you select are automatically selected as well.

6. Click Next to connect to the download site and download the packages you selected, and click Finish when the installation is complete.
7. If we choose install directly from internet it will install the Cygwin (D:\cygwin) and the package is also downloaded in a separate folder (D:\cygwinDownload). We can use this local download for installing the required packages one by one.



Cygwin Setup - Choose Installation Type

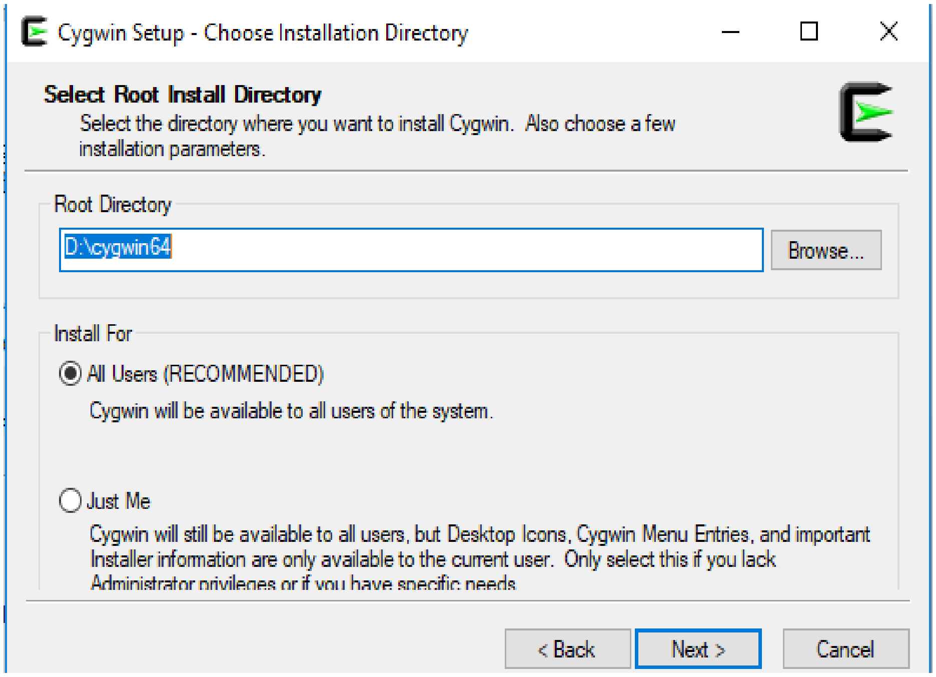
Choose A Download Source

Choose whether to install or download from the internet, or install from files in a local directory.

○ Install from Internet
(downloaded files will be kept for future re-use)

○ Download Without Installing

● Install from Local Directory



Cygwin Setup - Choose Installation Directory

Select Root Install Directory

Select the directory where you want to install Cygwin. Also choose a few installation parameters.

Root Directory: D:\cygwin64

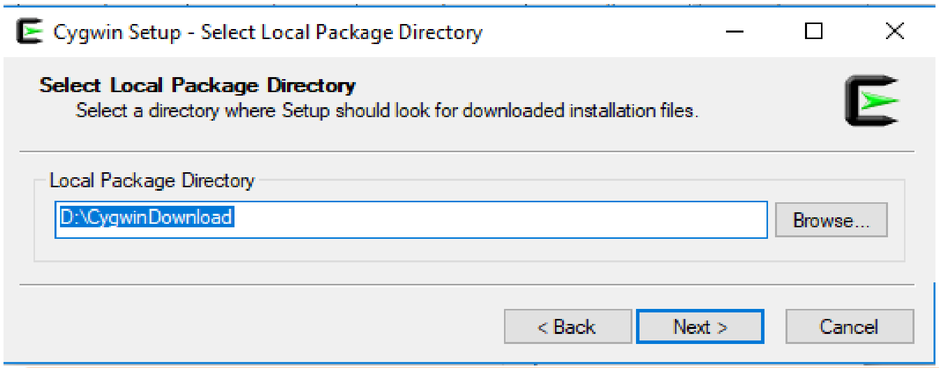
Install For

● All Users (RECOMMENDED)
Cygwin will be available to all users of the system.

○ Just Me
Cygwin will still be available to all users, but Desktop Icons, Cygwin Menu Entries, and important Installer information are only available to the current user. Only select this if you lack Administrator privileges or if you have specific needs.

Let's assume that Cygwin is already downloaded in a folder D:\cygwinDownload and we want to install the selected packages in D:\cygwin

When we start installing install the Cygwin base packages. If we accept the defaults and click next it will automatically install the base packages. After that we need to install gcc, make and gdb.

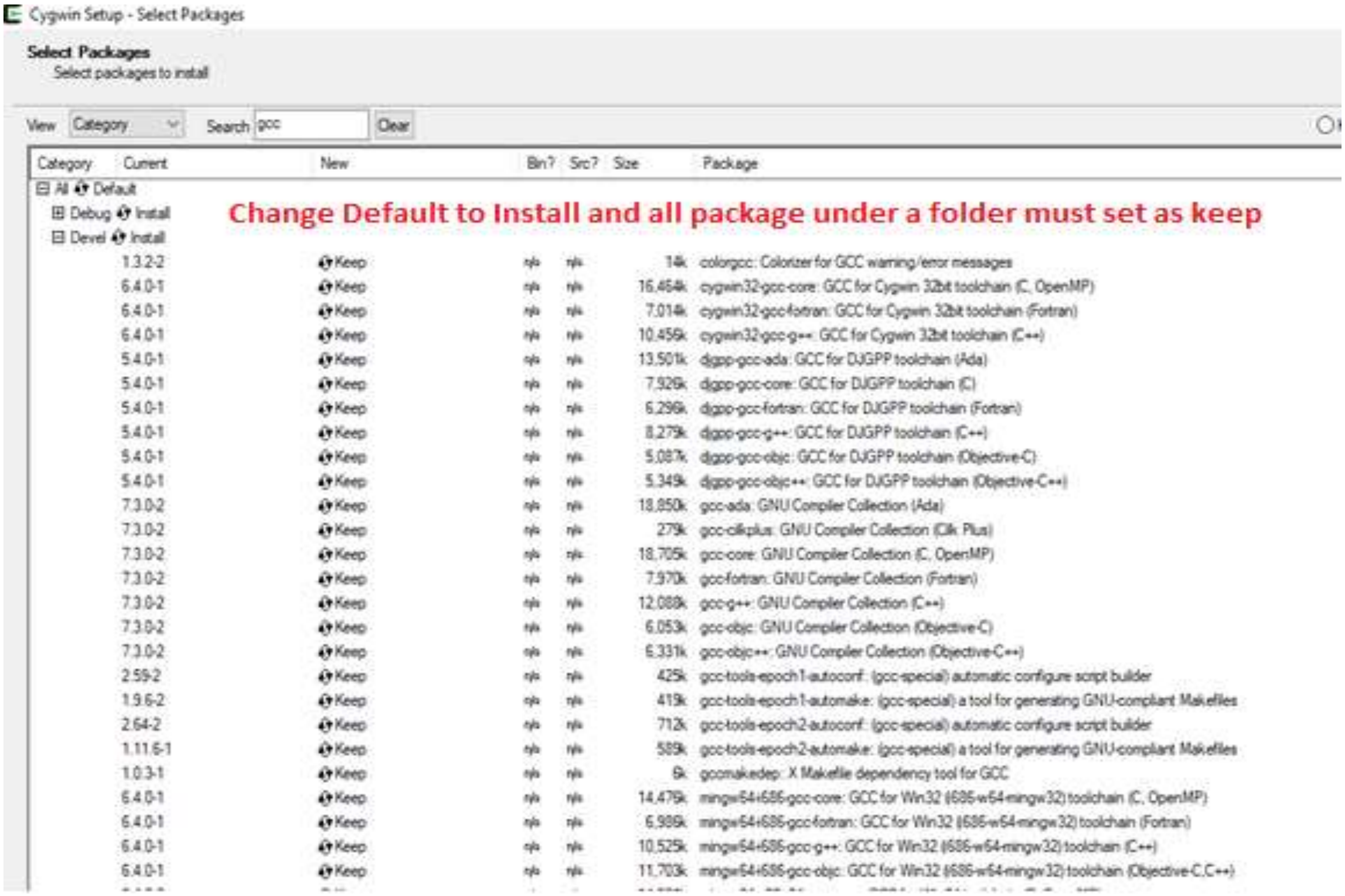


Cygwin Setup - Select Local Package Directory

Select Local Package Directory
Select a directory where Setup should look for downloaded installation files.

Local Package Directory: D:\CygwinDownload

In this screen select **Category** option in in View and then search for **gcc.** All the files needed for gcc will appear in different folder. Now change the default option to install against each folder. Make sure all files under that folder should set as Keep.



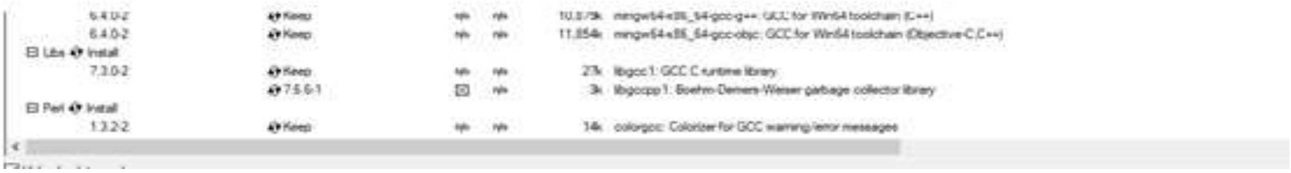
Cygwin Setup - Select Packages

Select Packages
Select packages to install

View: Category   Search: gcc

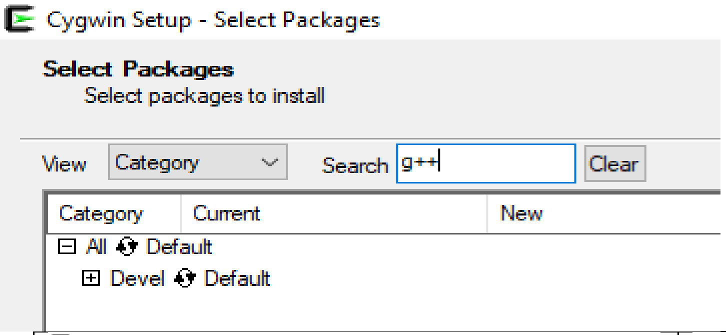
Change Default to Install and all package under a folder must set as keep

Category	Current	New	Bin?	Src?	Size	Package
⊟ All ⟳ Default						
⊞ Debug ⟳ Install						
⊟ Devel ⟳ Install						
	1.3.2-2	⟳ Keep	n/a	n/a	14k	colorgcc: Colorizer for GCC warning/error messages
	6.4.0-1	⟳ Keep	n/a	n/a	16,464k	cygwin32-gcc-core: GCC for Cygwin 32bit toolchain (C, OpenMP)
	6.4.0-1	⟳ Keep	n/a	n/a	7,014k	cygwin32-gcc-fortran: GCC for Cygwin 32bit toolchain (Fortran)
	6.4.0-1	⟳ Keep	n/a	n/a	10,456k	cygwin32-gcc-g++: GCC for Cygwin 32bit toolchain (C++)
	5.4.0-1	⟳ Keep	n/a	n/a	13,501k	djgpp-gcc-ada: GCC for DJGPP toolchain (Ada)
	5.4.0-1	⟳ Keep	n/a	n/a	7,926k	djgpp-gcc-core: GCC for DJGPP toolchain (C)
	5.4.0-1	⟳ Keep	n/a	n/a	6,296k	djgpp-gcc-fortran: GCC for DJGPP toolchain (Fortran)
	5.4.0-1	⟳ Keep	n/a	n/a	8,279k	djgpp-gcc-g++: GCC for DJGPP toolchain (C++)
	5.4.0-1	⟳ Keep	n/a	n/a	5,087k	djgpp-gcc-objc: GCC for DJGPP toolchain (Objective-C)
	5.4.0-1	⟳ Keep	n/a	n/a	5,349k	djgpp-gcc-objc++: GCC for DJGPP toolchain (Objective-C++)
	7.3.0-2	⟳ Keep	n/a	n/a	18,850k	gcc-ada: GNU Compiler Collection (Ada)
	7.3.0-2	⟳ Keep	n/a	n/a	279k	gcc-cilkplus: GNU Compiler Collection (Cilk Plus)
	7.3.0-2	⟳ Keep	n/a	n/a	18,705k	gcc-core: GNU Compiler Collection (C, OpenMP)
	7.3.0-2	⟳ Keep	n/a	n/a	7,970k	gcc-fortran: GNU Compiler Collection (Fortran)
	7.3.0-2	⟳ Keep	n/a	n/a	12,088k	gcc-g++: GNU Compiler Collection (C++)
	7.3.0-2	⟳ Keep	n/a	n/a	6,053k	gcc-objc: GNU Compiler Collection (Objective-C)
	7.3.0-2	⟳ Keep	n/a	n/a	6,331k	gcc-objc++: GNU Compiler Collection (Objective-C++)
	2.59-2	⟳ Keep	n/a	n/a	425k	gcc-tools-epoch1-autoconf: (gcc-special) automatic configure script builder
	1.9.6-2	⟳ Keep	n/a	n/a	419k	gcc-tools-epoch1-automake: (gcc-special) a tool for generating GNU-compliant Makefiles
	2.64-2	⟳ Keep	n/a	n/a	712k	gcc-tools-epoch2-autoconf: (gcc-special) automatic configure script builder
	1.11.6-1	⟳ Keep	n/a	n/a	589k	gcc-tools-epoch2-automake: (gcc-special) a tool for generating GNU-compliant Makefiles
	1.0.3-1	⟳ Keep	n/a	n/a	6k	gccmakedep: X Makefile dependency tool for GCC
	6.4.0-1	⟳ Keep	n/a	n/a	14,479k	mingw64-i686-gcc-core: GCC for Win32 (i686-w64-mingw32) toolchain (C, OpenMP)
	6.4.0-1	⟳ Keep	n/a	n/a	6,986k	mingw64-i686-gcc-fortran: GCC for Win32 (i686-w64-mingw32) toolchain (Fortran)
	6.4.0-1	⟳ Keep	n/a	n/a	10,525k	mingw64-i686-gcc-g++: GCC for Win32 (i686-w64-mingw32) toolchain (C++)
	6.4.0-1	⟳ Keep	n/a	n/a	11,703k	mingw64-i686-gcc-objc: GCC for Win32 (i686-w64-mingw32) toolchain (Objective-C,C++)



Category	Current	New	Bin?	Src?	Size	Package
	6.4.0-2	⟳ Keep	n/a	n/a	10,879k	mingw64-x86_64-gcc-g++: GCC for Win64 toolchain (C++)
	6.4.0-2	⟳ Keep	n/a	n/a	11,854k	mingw64-x86_64-gcc-objc: GCC for Win64 toolchain (Objective-C,C++)
⊟ Libs ⟳ Install						
	7.3.0-2	⟳ Keep	n/a	n/a	27k	libgcc1: GCC C runtime library
		⟳ 7.5.6-1	☒	n/a	3k	libgccpp1: Boehm-Demers-Weiser garbage collector library
⊟ Perl ⟳ Install						
	1.3.2-2	⟳ Keep	n/a	n/a	14k	colorgcc: Colorizer for GCC warning/error messages

Similarly search for **g++, gdb** and **make** and set the install option as before.

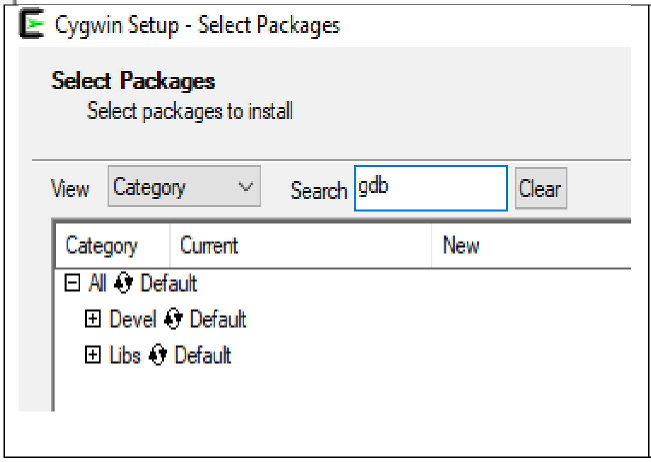


Cygwin Setup - Select Packages

Select Packages
Select packages to install

View: Category Search: g++

Category	Current	New
⊟ All ⟳ Default		
⊞ Devel ⟳ Default		

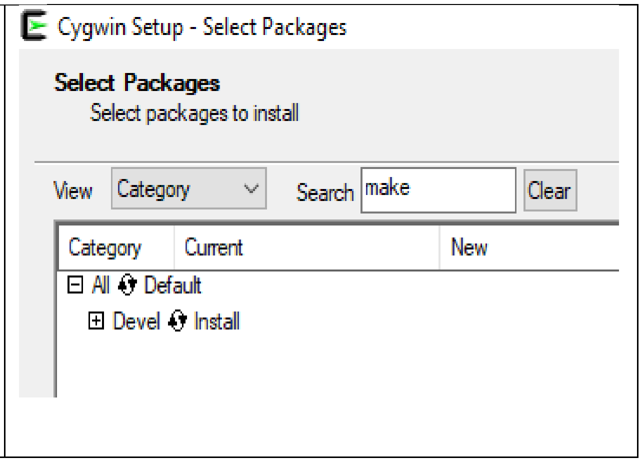


Cygwin Setup - Select Packages

Select Packages
Select packages to install

View: Category Search: gdb

Category	Current	New
⊟ All ⟳ Default		
⊞ Devel ⟳ Default		
⊞ Libs ⟳ Default		



Cygwin Setup - Select Packages

Select Packages
Select packages to install

View: Category Search: make

Category	Current	New
⊟ All ⟳ Default		
⊞ Devel ⟳ Install		

In this way we select one by one package and install them

8. Add the Cygwin lib and bin folders to your **system environment variables**. In my case it is D:/Cygwin/bin and D:/Cygwin/lib.

9. After installation we can check : C:\> cygcheck -c cygwin
   Check the versions of the Cygwin gcc and g++ compilers, make, and gdb by typing the following commands at a Windows command prompt:
   C:\> gcc –version C:\> g++ --version C:\> make –version C:\> gdb --version

Verifying the Installation

To verify that the installation is correct, start NetBeans IDE, build a sample project, and run it in the gdb debugger.

- Double-click the NetBeans IDE icon on the desktop.

To build a sample project and run it in the debugger:

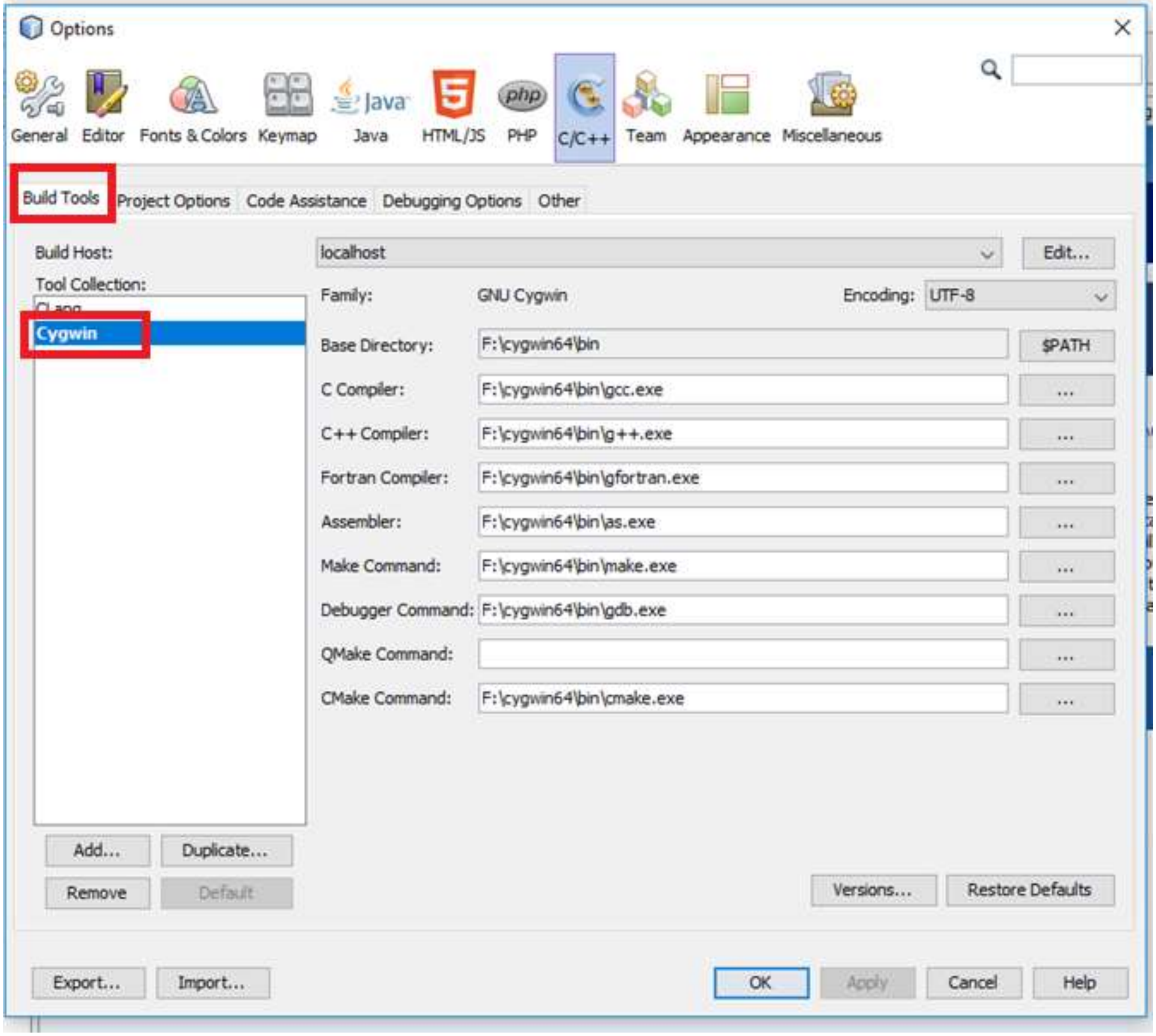
1. Open the New Project wizard by choosing File > New Project.

2. In the Categories panel on the Choose Project page of the wizard, expand the Samples category and select the C/C++ subcategory.
3. In the Projects panel, select the Welcome project. Click Next.
4. On the Project Name and Location page, click Finish.
5. In the Projects window, right-click the Welcome_1 project and choose Build. If your compilers and make utility are installed correctly and the path to them is set, build output is displayed in the Output window and the project builds successfully.
6. Expand the Source Files node of the project and double-click the welcome.cc file to open it in the Source Editor.
7. Set a breakpoint by clicking in the left margin of the Source Editor window next to any line.
8. Right-click the project and choose Debug. If the gdb debugger is installed correctly and the path to it is set, gdb starts up, the Debugger tabs are displayed, and the Welcome application runs and stops at the breakpoint.
9. Choose Debug > Continue to run the application to completion.
10. If the project doesn't build or debugger doesn't work, see the next section.

Troubleshooting Tool Issues

1. Select Tools > Options and click C/C++ in the top panel of the Options dialog box.
2. In the Build Tools tab, look at the Tool Collection list to see if your tool collection is listed.
3. Select the tool collection if it is listed, and check the paths to the tools. If the Base Directory path is incorrect, you may have made a mistake when setting your path environment variable. Refer back to the instructions for setting the path in the section for your platform in this document, and correct the path if necessary.
4. If the tool collection is not listed, click Restore Defaults. This will cause the IDE to rescan your environment to look for tools and the tool collection should be found if the path environment variable is correct.
5. If you are still having an issue, try adding a new tool collection and specifying the path to the tools, as follows:
6. Click **Add** below the Tool Collection list.
    - Browse to the directory that contains the executables for the compilers, debugger, and make utility. This is usually the bin directory.
    - Select the directory and click Open. The IDE should be able to determine the family of the tools and update the other fields in the dialog box appropriately. If not, an error is displayed.
    - If the tools are identified correctly, click OK in the Add Tool Collection dialog box. The new collection should be displayed in the list.
    - Select the new tool collection and click Default to make sure this tool collection is used automatically for new projects.
      If you cannot solve the problem, ask for help at the NetBeans C/C++ User Forum.

In my machine at home the settings are shown below :



Build Host: localhost

Tool Collection:

CLang
Cygwin

Family: GNU Cygwin

Encoding: UTF-8

Base Directory: F:\cygwin64\bin

C Compiler: F:\cygwin64\bin\gcc.exe

C++ Compiler: F:\cygwin64\bin\g++.exe

Fortran Compiler: F:\cygwin64\bin\gfortran.exe

Assembler: F:\cygwin64\bin\as.exe

Make Command: F:\cygwin64\bin\make.exe

Debugger Command: F:\cygwin64\bin\gdb.exe

QMake Command:

CMake Command: F:\cygwin64\bin\cmake.exe