How to build the GNU Compiler Collection

The following instructions shall guide you through the build of the GNU Compiler Collection (GCC).

Note

These instructions were tested on **Ubuntu 12.04** and may differ from those required to build GCC for your operating system. Please refer also to the official GCC installation instructions.

For previous versions of the Ubuntu OS system, it is highly probable that the required version of the compiler, i.e., version 4.3 if you use MATLAB R2009b, is available in their repository. In this case, the installation may be performed in the straightforward manner through the `Synaptic Package Manager`. Make sure that the `libstdc++` and `g++` packages corresponding to the desired version are also installed.

In the most recent versions, among them *Ubuntu 12.04*, there is no package for version 4.3, however. Thus, GCC has to be build sources.

1. Install the build dependencies:

   ```
   sudo apt-get build-dep gcc-4.5
   ```

   The following may be required to avoid linking problems:

   ```
   export LD_LIBRARY_PATH=/usr/lib/x86_64-linux-gnu
   ```

   Alternatively, create a symbolic link such as:

   ```
   sudo ln -s /usr/lib/x86_64-linux-gnu /usr/lib64
   ```

2. Obtain the GCC source distribution package and extract the files:

   ```
   cd
   MIRROR=ftp://ftp.mirrorservice.org/sites/sourceware.org
   wget ${MIRROR}/pub/gcc/releases/gcc-4.3.4/gcc-4.3.4.tar.bz2
   tar -xjvf gcc-4.3.4.tar.bz2
   rm -f gcc-4.3.4.tar.bz2 (optional)
   ```

3. Make a separate build directory and change to it:

   ```
   mkdir gcc-4.3.4-build
   cd gcc-4.3.4-build
   ```

4. Configure the build and set the installation prefix to a different location:

   ```
   ../gcc-4.3.4/configure --prefix=/opt/gcc-4.3.4/
   ```

5. Once configured, build GCC using GNU Make:

   ```
   make [-j<#cores>]
   ```

6. To install the built executables, run:

   ```
   make install
   ```

7. Once the installation is done, you may remove the source and build files again:

   ```
   cd ..
   rm -rf gcc-4.3.4
   rm -rf gcc-4.3.4-build
   ```

Note

On Mac OS, use `curl -O` instead of `wget` in step 1 or simply download the sources using the browser of your choice instead.

Now, there should be two different versions of the GNU Compiler Collection being installed. When you build any of the other build dependencies of GONDOLA as well as GONDOLA itself, make sure that the correct version of the compiler is used. In CMake, toggle therefore to the advanced view and check the value of the `CMAKE_C_COMPILER` and `CMAKE_CXX_COMPILER` variables. Correct the paths to point to the right version if necessary. Note that after this change, CMake must reconfigure everything from scratch and all other settings will be lost. Therefore, set these values first. To avoid having to set these values after the first initialization of CMake, call `ccmake` as follows:

```
ccmake -DCMAKE_C_COMPILER:FILEPATH=/opt/gcc/4.3.4/gcc \
    -DCMAKE_CXX_COMPILER:FILEPATH=/opt/gcc-4.3.4/g++ \
    [options] <source dir>
```

Note

If you encounter the following error during the compilation of GONDOLA using GCC version 4.3.4, replace the file */opt/gcc-4.3.4/libjava/prims.cc* by the one available here. For more information, see the bug report.

```
./.libs/libgcj.so: undefined reference to `__cxa_call_unexpected'
```