Installing Alternate GCC Compiler on Fedora

Mauriat Miranda (http://www.mjmwired.net/contact/)

Other Guides/Resources

Published: 6 March 2006 (*updated: 6 March 2006*)

Post To: del.icio.us, furl, digg, spurl, gmail this, Blog this

Installing Alternate GCC Compiler on Fedora

This guide shows how to installed an alternate GCC compiler from *source* using Fedora Core. By using these steps you should be able to configure and install any GCC compiler you might require. You can *upgrade your GCC compiler* or *downgrade your GCC compiler* if you require.

NOTE:
Make sure you understand all of these steps completely. This is NOT meant for everyone. Please be advised there MAY BE errors. Comments and/or suggestions appreciated.

Problem

Certain releases of Linux distributions like Fedora Core ship with versions of the GCC compiler that may not be compatible with other software. In some cases the compiler is very new (GCC4.1 in FC5) and in other cases the compiler may be non standard (GCC2.96 in RH7.3). Usually there should not be any major problems. However many open source applications will NOT properly compile using specific versions of GCC. Additionally many applications or code bases may not fully utilize/support newer versions of GCC. Specific developers may require installing their own specific compiler. *So if you **do not need** to compile, then this guide is unnecessary.*

Having Multiple Compilers

Multiple compilers is typically reserved for developers or those withthe need to compile for separate architectures (i.e. "*cross compiling*"). Redhat and Fedora have shipped with alternate compilers (both FC4 and FC5 shipped with a GCC4.x as well as GCC3.2).

Linux is convenient that you can have as many compilers as you like.

Required Software

You must have at least one compiler installed in Fedora. This varies in installation from different versions of Fedora. However most installations will install one version of GCC by default. In the newer releases (for example FC4) this includes: Development Tools in the FC4 Install Process -OR- you can go to: *System Settings > Add/Remove Applications > Development > Development Tools*. The following is an example on FC4:

```
[mirandam@charon fc4]$ rpm -q gcc
gcc-4.0.0-8
```

Required Files

To install from source obtain GCC from any [GCC mirror sites](). It is **your personal selection** in choice of the version of GCC. Upon writing this draft `GCC 3.4.4` is available. Try to find a mirror with the full files NOT the `diff` files. The following is only an *example*:

```
gcc-3.4.4.tar.bz2               26920 KB        05/19/2005      10:50:00 AM

gcc-ada-3.4.4.tar.bz2           3380 KB         05/19/2005      10:50:00 AM
gcc-core-3.4.4.tar.bz2          12846 KB        05/19/2005      10:52:00 AM
gcc-g++-3.4.4.tar.bz2           2417 KB         05/19/2005      10:50:00 AM
gcc-g77-3.4.4.tar.bz2           882 KB          05/19/2005      10:51:00 AM
gcc-java-3.4.4.tar.bz2          4588 KB         05/19/2005      10:51:00 AM
gcc-objc-3.4.4.tar.bz2          208 KB          05/19/2005      10:51:00 AM
gcc-testsuite-3.4.4.tar.bz2     2575 KB         05/19/2005      10:51:00 AM
```

Select only which components you wish to install. You will need the `gcc-core` and any extra language you use (ada, g++, etc.) -OR- download `gcc-3.4.4` which will contain ALL of the languages above. ... For this example, I only require `gcc-core` and `gcc-g++`.

Setting Up Sources and Directories

GCC recommends you do NOT build in the same directory as your source files are. So simply create a separate `build` directory. To start, I have downloaded and unzipped my selected `tar.bz2` file from above into the directory: `gcc`.

```
[mirandam@charon gcc]$ pwd
/home/mirandam/gcc

[mirandam@charon gcc]$ bzip2 -cd gcc-core-3.4.4.tar.bz2 | tar xvf -
[mirandam@charon gcc]$ bzip2 -cd gcc-g++-3.4.4.tar.bz2 | tar xvf -

-OR (the whole thing)-
[mirandam@charon gcc]$ bzip2 -cd gcc-3.4.4.tar.bz2 | tar xvf -

[mirandam@charon gcc]$ mkdir build        (this is our build directory)
[mirandam@charon gcc]$ cd build           (important step)
```

Our build directory above is `/home/mirandam/gcc/`**`build`**`/`. All our source code for GCC-3.4.4 is located in `/home/mirandam/gcc/`**`gcc-3.4.4/`**.

Configuring Options

Extensive options are listed on the [GCC configuration page](#) provided on gnu.org. **Most importantly the options must be set so as to NOT conflict or interfere with the GCC already installed on Fedora Core installation.**

The following options are some recommendations and observations taken from Fedora's configuration of GCC.

- `prefix` is used to set where GCC will be installed. This is typically `/usr/local`, but I want to keep it completely separate in `/opt/gcc34`. If I ever plan to remove it, I will simply remove the directory. The compiled `gcc,g++,`etc. will be in `/opt/gcc34/bin/`.
- `program-suffix` is used to attach an extra character or string to every GCC application that gets compiled. I have used `34`. So my compiler will be `gcc34`, `g++34`, etc. This is done so I avoid any confusion with the existing GCC installed (or even GCC 3.2 in FC4). You can also use `program-prefix` instead (ex: `mygcc`).
- `enable-languages` allows you to control which compilers you want built. If you select all (c,c++,objc,java,f95,ada) make sure you either have all the language added or you used the full `gcc tag.bz2` package. (I only care for c and c++.)
- The last set of options (in `italics`) are meant entirely for compatibility with Fedora.

```
[mirandam@charon build]$ pwd
/home/mirandam/gcc/build

[mirandam@charon build]$ /home/mirandam/gcc/gcc-3.4.4/configure \
        --prefix=/opt/gcc34 \
        --program-suffix=34 \
        --enable-languages=c,c++ \
        --enable-shared --enable-threads=posix --disable-checking \
        --with-system-zlib --enable-__cxa_atexit --disable-libunwind-exceptions
```

Compile and Install

Once done configuring above, you just need to compile and install. Everything has been compiled as a user. Make sure to only use when installing.

```
[mirandam@charon build]$ make              (this compiles, takes 10-30min)

[mirandam@charon build]$ su                (logs you in as root, to install)
Password:

[root@charon build]# pwd
/home/mirandam/gcc/build

[root@charon build]# make install
```

Now **create** a text file named: `gcc34.sh` and in it add the following.

```
#!/bin/sh
GCC34_BIN=/opt/gcc34/bin
PATH=$GCC34_BIN:$PATH
export PATH
```

As root, place the file into `/etc/profile.d/` and make sure it has execute permissions for all.

```
[root@charon ~]# ls -la /etc/profile.d/gcc34.sh
-rw-r--r--  1 root root 66 Jun 15 21:38 /etc/profile.d/gcc34.sh

[root@charon ~]# chmod 755 /etc/profile.d/gcc34.sh

[root@charon ~]# ls -la /etc/profile.d/gcc34.sh
-rwxr-xr-x  1 root root 66 Jun 15 21:38 /etc/profile.d/gcc34.sh
```

You should logout and log back in, or just run a new terminal or xterm and `gcc34` should work from any prompt.

Using New GCC

If you installed as above, running `gcc34` or any other app you installed should be directly accessible. You can verify with the following steps:

```
[mirandam@charon ~]$ which gcc34
/opt/gcc34/bin/gcc34
[mirandam@charon ~]$ which g++34
/opt/gcc34/bin/g++34

[mirandam@charon ~]$ gcc34 -v
Reading specs from /opt/gcc34/lib/gcc/i686-pc-linux-gnu/3.4.4/specs
Configured with: /home/mirandam/gcc/gcc-3.4.4/configure --prefix=/opt/gcc34
```

```
--program-suffix=34 --enable-languages=c,c++ --enable-shared --enable-threads=po
--disable-checking --with-system-zlib --enable-__cxa_atexit
--disable-libunwind-exceptions
Thread model: posix
gcc version 3.4.4
```

There are many ways to use an alternate compiler. The 3 methods I commonly use: 1. Environment Variable, 2. Configure Support, and 3. Makefile Support.

Environment Variables

Most softwares support the `cc` and `cxx` environment variables. First assign them, then run `configure` and/or `make`. Example:

```
# export CC=gcc34
# export CXX=g++34
# ./configure
```

Configure Support

If the software is using the standard GNU automake and configure, then there is a chance it supports other compilers by passing in a setting to the `configure` script. First run `configure --help` to see if it mentions anything. The following example is from MPlayer:

```
# ./configure --help
# ./configure --cc=gcc34
```

Makefile Support

Sometimes the software may just come with a `Makefile`. Open the `Makefile` and look inside to see if there are variables that specify the compiler. You can either edit those variables or set them at compile time. For example:

```
(in Makefile)

C_COMPILER = cc
CPLUSPLUS_COMPILER = c++
```

Then using the `Makefile`, you can run:

```
# make CPLUSPLUS_COMPILER=g++34
```

Circuit City DVDs

Get The Hottest New Movies! Free Shipping on Orders $24 & Up.

www.CircuitCity.com

More information:

- GCC home page
- GCC mirror sites
- GNU official setup of GCC
- Installing GCC: Configuration
- Linux with Doubts :: Using Alternate Compilers
- http://www.krischik.com/Ada/gnat-3_4.html
- Italian Translation: Installing Alternate GCC Compiler on Fedora

Comments, suggestions, questions or any feedback welcome for this page or any of my Resources. Please use the contact link.

Help Out: If you found this guide or any Resource helpful, please consider supporting this site by recommending this page to others or linking to this page. I appreciate all the support I receive. Thank you in advance.

- If you wish to donate to pay for my hosting:

  Make a Donation

- If you wish to thank me via Amazon.com wishlist.