

Building a GNU Toolset For Use In Embedded PowerPC™ Cross Development

1 Abstract

This application note describes the process of creating a PowerPC cross development toolset using the GNU GCC compiler package and GNU binary utilities package. The GNU GCC compiler package consists of a C/C++ compiler, Fortran compiler, and Java compiler. The GNU binary utilities package consists of an assembler, linker, archiver and various object file manipulation utilities. The GNU GCC compiler and GNU binary utilities can be used for developing embedded PowerPC applications.

2 Introduction

The GNU GCC compiler package and GNU binary utilities package provide tools that can be used in a cross development environment. The GNU GCC compiler package and GNU binary tools package can be obtained from a number of WEB and FTP sites on the Internet. Both packages contain source code and build scripts and must be compiled for a specific host and target platform. Both packages are distributed under the GNU General Public License, Version 2. The text of this license is located on the <http://www.fsf.org/licenses/info/GPLv2.html> WEB site.

The GNU tools are designed to work with any UNIX® operating system, for example AIX, Solaris™, and Linux. In order to compile and use the GNU tools with the Windows operating system the Cygwin software is required. The cygwin software provides UNIX operating system emulation for Windows systems as well as many programs that are commonly included with UNIX operating systems.

The following list shows the major steps required to compile the GNU GCC compiler package and the GNU binary tools package:

- Download and Install the Cygwin software (required only for the Windows operating system)
- Download, configure, compile, and install the GNU binary utilities package
- Download, configure, compile, and install the GNU GCC compiler package

The process of building the GNU GCC compiler package and GNU binary utilities package requires about 1.2 GByte of space. This includes the space required for the Cygwin software as well as the space required for all the packages.

3 Downloading and Installing The Cygwin Software

If the GNU tools are to be used with the Windows operating system, the Cygwin software must be installed. Only 32-bit Windows systems are supported by the Cygwin software (Windows CE is not supported). The Cygwin software is installed by the Cygwin net release setup program. The Cygwin setup program can be downloaded from the <http://www.cygwin.com> WEB site. On the <http://www.cygwin.com> WEB site click on the Install Cygwin Now icon and save the **setup.exe** (<http://www.cygwin.com/setup.exe>) file.

The Cygwin setup program installs the Cygwin software. The second screen in the Cygwin setup program shows three basic installation options: **install from Internet**, **download from Internet**, or **install from local directory**.

Building a GNU Toolset For Use In Embedded PowerPC Cross Development

The **install from Internet** option downloads the Cygwin software packages to a local directory before starting the install process. This option can be used for a computer that has a persistent connection to the Internet. This option is recommended for most users.

The **download from the Internet** option downloads the Cygwin software packages to a local directory but does not perform the install process. This option can be used if the downloaded files are to be installed on several computers. After downloading the files, the **install from local directory** option must be chosen to actually install the Cygwin software packages.

The **install from local directory** option installs the Cygwin software packages that were previously downloaded using the **download from the Internet** option. This option assumes that the Cygwin software packages reside in a local directory on a computer running the setup program.

The third screen in the setup program allows for the selection of the root installation directory. This directory will become the / (root) directory in the Cygwin installation. The default value of **c:\cygwin** can be used in most cases. The “**Install For**” selection should be set to “**All Users**” unless the setup program is being executed by a user without administrator privileges. If the “**Install For**” selection is not set to “**All Users**” some programs (**sshd**, **crond**) will not function correctly. The “**Default Text File Type**” selection should be set to UNIX. This option chooses /n as the line end delimiter instead of /r/n.

The fourth screen in the setup program allows the user to select the local directory where the software packages will be stored. Any directory except for the root installation directory can be selected.

The next screen allows user to select an Internet connection method. The “**Use IE5 Setting**” selection allows for the setup program to use proxy server and auto-configuration script's that were used to setup the Internet Explorer 5 program. The Use “**HTTP/FTP proxy**” selection allows for manual proxy selection. This option does not support password authentication for proxy services. The correct option depends on the type of connection to the Internet.

The sixth screen in the setup program allows user to choose a download site. Multiple sites can be selected by holding the control key and selecting a site.

Before the Select Packages screen is shown the setup program downloads a file listing all the software packages that are available on the chosen mirror site. The Select Package screen allows the user to choose the software packages that will be downloaded or installed. The software packages are organized into several categories. A single software package can belong to multiple categories. By default only packages in the **base category** will be installed. In order to compile the GNU tools all the packages in the **base category** must be selected. In addition to the packages in the **base category** many packages in the **devel category** must also be installed. If there is sufficient space on the target computer all the packages in the **devel category** should be installed. A vim package in **editors category** and more packages in the **text category** can also be installed in order to provide more complete UNIX environment. A perl package in **interpreters category** must also be installed. The texinfo package in the **text category** must also be selected so that the GNU tools documentation can be converted to a user readable format. Selecting the next button on the Select Packages screen will start the download or installation process. The following screen will show the installation or download progress. If the chosen action was to install Cygwin software before the installation is complete, the user can choose to install shortcuts to the **bash** shell on the desktop or the start menu. If shortcuts are not created, the **bash** shell can be started by executing the **cygwin.bat** file in the Cygwin root directory. During the last step in the installation process the setup program will execute a number of post-install scripts. Each script will execute in a separate window. When all of the scripts finish executing the setup program will display a window announcing the completion of the installation process.

Information on the installed packages is kept in the **/etc/setup** directory of the Cygwin installation. If this directory is removed the setup program will assume that the Cygwin software was not installed.

4 Downloading, Configuring, Compiling, and Installing The GNU Binary Utilities Package

The source for the following programs is included in the GNU binary utilities package:

- **addr2line** (converts addresses into filenames and line numbers)
- **ar** (utility for creating and manipulating archive files)
- **as** (the GNU assembler)
- **c++filt** (filter to demangle encoded C++ symbols)
- **ld** (the GNU linker)
- **nm** (lists symbols from object files)
- **objcopy** (copies and translates object files)
- **objdump** (displays information from object files)
- **runlib** (generates an index to the contents of an archive)
- **readelf** (displays information from any ELF format object file)
- **size** (lists the section sizes of an object or archive file)
- **strings** (lists printable strings from files)
- **strip** (discards symbols in executable files)

The GNU binary utilities package can be downloaded from the <http://www.gnu.org> WEB site. On the main GNU WEB page choose The Free Software Directory link. On the following page choose **All GNU Packages** link. The next page will display a list of the available GNU packages. On this page chose the **binutils** link. The following page will have a link to a source code file. This file is sometimes referred to as a Source Tarball. Click on this link and save the file in a local directory, for example **c:\data\gcc**. The downloaded file contains the latest version of the GNU binary utilities package.

There are a number of GNU mirror sites where the GNU software can be obtained. For example the <http://www.ibiblio.org/pub/gnu/> site which offers a number of GNU packages. The mirror sites often have a number of version of the GNU packages.

This application note assumes that the latest version of the GNU binary utilities package is 2.14. In this case the file containing the GNU binary utilities package will be called **binutils-2.14.tar.gz** (<http://www.ibiblio.org/pub/gnu/binutils/binutils-2.14.tar.gz>).

The **gz** file name extension indicates that the file was compressed with GNU **gzip** program. The **tar** file name extension indicates that the file was created using a **tar** program. The following steps can be used to configure, compile and install the GNU binary utilities package.

1. Start the Cygwin **bash** shell.
2. Change to the directory where the GNU binary package was downloaded (for example **c:\data\gcc** by executing the **cd /cygdrive/c/data/gcc** command). The directory where the GNU binary package is built and the directory where the package is downloaded can not contain a space character.
3. Execute the **gunzip binutils-2.14.tar.gz** command. This command will uncompress the file and create a **binutils-2.14.tar** file. The compressed file will be removed by the **gunzip** command (use **bunzip2** command when the source file ends with the **bz2** suffix).

Building a GNU Toolset For Use In Embedded PowerPC Cross Development

4. Execute the **tar -xvf binutils-2.14.tar** command. This command will extract all the source files from a single file. After this process is complete a **binutils-2.14** directory should have been created. This directory will contain all the files required to build the GNU binary utilities.
5. Make a new directory called **binutils-build** (at the same level as **binutils-2.14** directory) by executing the **mkdir binutils-build** command and change to the newly created directory using the **cd binutils-build** command. It is recommended that the build process be performed in a directory that is different from the directory where all the source files are located.
6. Execute **../binutils-2.14/configure --target=powerpc-ibm-eabi i686-ibm-cygwin** command. This command will take considerable time to execute and will create many files in the **binutils-build** directory. The newly created files will be used in the build process. The first parameter passed to the configure command specifies the target architecture for the GNU binary utilities. In this case the PowerPC architecture is specified. The second parameter identifies the host system used to build the GNU binary tools package and must accurately describe such system. For example when building under the Solaris operating system; the second parameter can be set to **sparc*-solaris**. When building under AIX the second parameter can be set to **powerpc-ibm-aix** or **rs6000-ibm-aix** (the AIX operating system version should be specified after the **aix** keyword, for example **aix4.3**). When building under Linux the second parameter can be set to **i686-ibm-linux**. When building binary utilities for a 64-bit PowerPC architecture the target should be set to **powerpc64-ibm-linux**.
7. Execute the **make** command. This will build the GNU binary utilities. This command will take considerable time to execute (longer than the **configure** command).
8. Execute the **make install** command. This will install the GNU binary utilities in the **/usr/local/bin** directory. All the executable files that are installed in this directory will have the **powerpc-ibm-eabi** file name prefix. The file name prefix is used to identify this version of the GNU binary utilities from other versions of the binary utilities that might be installed on the same machine.
9. Execute the **powerpc-ibm-eabi-as --help** command. This verifies that the tools were built and installed.

5 Downloading, Configuring, Compiling, and Installing The GNU GCC Compiler Package

The source for the following programs is included in the GNU GCC compiler package:

- **c++** (C++ compiler)
- **cpp** (C preprocessor)
- **g++** (same as c++)
- **g77** (Fortran 77 compiler)
- **gcc** (C/C++ compiler)
- **gccbug** (program that submits problem reports on GCC compiler)
- **gcj** (ahead of time compiler for Java language)
- **gcjh** (generate C++ header files from .class files)
- **gconv** (coverage testing tool can be used to optimize code)
- **jcf-dump** (display contents of a class file in readable form)
- **jv-scan** (print useful information read from Java source file)

Building a GNU Toolset For Use In Embedded PowerPC Cross Development

The GNU GCC compiler package can be downloaded from the <http://www.gnu.org> WEB site. On the main GNU WEB page choose **The Free Software Directory** link. On the following page choose **All GNU Packages** link. The next page will display a list of all available GNU packages. On this page choose the **GCC** link. The following page will have a link to a source code file. This file is sometimes referred to as a Source Tarball. Click on this link and save the file in a local directory, for example `c:\data\gcc`. The downloaded file contains the latest version of the GNU GCC compiler package.

There are a number of GNU mirror sites where the GNU software can be obtained. For example the <http://www.ibiblio.org/pub/gnu/> site offers number of GNU packages. The mirror sites often have several versions of the GNU packages.

This application note assumes that the latest version of the GNU GCC compiler package is 3.3.1. In this case the file containing the GNU binary utilities package will be called **gcc-3.3.1.tar.gz** (<http://www.ibiblio.org/pub/gnu/gcc/gcc-3.3.1.tar.gz>).

The **gz** file name extension indicates that the file was compressed with the GNU **gzip** program. The **tar** file name extension indicates that the file was created using a **tar** program. The following steps can be used to configure, compile and install the GNU GCC compiler package.

1. Complete the compilation and installation of the GNU binary utilities package described in the previous section. The GNU binary utilities package must be installed before the GNU GCC compiler package can be compiled. The target configuration for the GNU binary utilities package (**powerpc-ibm-eabi**) must be the same as the target configuration string for the GNU GCC compiler package.
2. Start the Cygwin **bash** shell.
3. Change to the directory where the GNU GCC compiler package was downloaded (for example `c:\data\gcc` by executing the `cd /cygdrive/c/data/gcc` command). The directory where the GNU GCC compiler package is built and the directory where the package is downloaded can not contain a space character.
4. Execute the `gunzip gcc-3.3.1.tar.gz` command. This command will uncompress the file and create a **gcc-3.3.1.tar** file. The compressed file will be removed by the `gunzip` command (use `bunzip2` command when the source file ends with the **bz2** suffix).
5. Execute the `tar -xvf gcc-3.3.1.tar` command. This command will extract all the source files from a single file. After this process is complete a **gcc-3.3.1** directory should have been created. This directory will contain all the files required to build the GNU GCC compiler.
6. Make a new directory called **gcc-build** (at the same level as **gcc-3.3.1** directory) by executing the `mkdir gcc-build` command and change to the newly created directory using the `cd gcc-build` command. It is recommended that the build process be performed in a directory that is different from the directory where all the source files are located.
7. Execute `../gcc-3.3.1/configure --target=powerpc-ibm-eabi --with-newlib --disable-nls --disable-multilib i686-ibm-cygwin` command. This command will take considerable time to execute and will create many files in the **gcc-build** directory. The newly created files will be used in the build process. The first parameter passed to the configure command specifies the target architecture for the GNU GCC compiler. In this case PowerPC architecture is specified. The third parameter identifies the host system used to build the GNU GCC compiler package and must accurately describe such system. For example when building under the Solaris operating system; the third parameter can be set to **sparc*-solaris**. When building under the AIX the third parameter can be set to **powerpc-ibm-aix** or **rs6000-ibm-aix** (the AIX operating system version should be specified after the **aix** keyword, for example **aix4.3**). When building under Linux the third parameter can be set to **i686-ibm-linux**. Ignore warning about certain libraries not being supported in this configuration. When compiling for a processor that supports the

Building a GNU Toolset For Use In Embedded PowerPC Cross Development

Altivec™ instruction set use the **~~---enable-altivec~~** option. When building the GNU GCC compiler for a 64-bit PowerPC architecture the target should be set to **powerpc64-ibm-linux**, and in addition to the options used for 32-bit PowerPC architecture both the **~~---disable-threads~~** and the **~~---disable-shared~~** options should also be used. For more information on configuring GCC compiler see <http://gcc.gnu.org/install/configure.html>.

8. Edit the **makefile** in the **gcc-build** directory. In the **Makefile** remove all the directories from the **TARGET_CONFIGDIRS** statement. The original statement might look something like this (the actual **TARGET_CONFIGDIRS** statement might be different from this example): **TARGET_CONFIGDIRS = libiberty libgloss newlib librx libstdc++-v3 libf2c libobjc examples**. The modified statement should look like this:

TARGET_CONFIGDIRS =

Note: When compiling GNU GCC compiler version 3.4.X or later, additional edits must be performed to the top-level **Makefile**. The **all-target**, **configure-target**, and **install-target** statements should be edited to include only the **maybe-all-target-newlib**, **maybe-configure-target-newlib**, and **maybe-install-target-newlib** items respectively. The modified statements should look like this:

all-target: maybe-all-target-newlib

configure-target: maybe-configure-target-newlib

install-target: maybe-install-target-newlib

9. When compiling GNU GCC compiler version 3.4.X or later for the 64-bit PowerPC architecture some target specific include file must be placed in the **/usr/local/powerpc64-ibm-linux/include** directory in order to correctly build the compiler. The files provided in the EPOS (Embedded PowerPC Operating System) **kernel/include** directory satisfy this requirement.
10. Execute the **make** command. This will build the GNU GCC compiler. This command will take considerable time to execute (longer than the **configure** command). Some warnings are expected during the GNU GCC compiler build process.
11. Execute the **make install** command. This will install the GNU GCC compiler in the **/usr/local/bin** directory. All the executable files that are installed in this directory will have the **powerpc-ibm-eabi** file name prefix. The file name prefix is used to identify this version of the GNU GCC compiler from other versions of the compiler that might be installed on the same machine.
12. Execute the **powerpc-ibm-eabi-gcc --help** command. This verifies that the tools were built and installed.

Note: When using the **---prefix=dirname** option the same prefix option must be used for configuring the GNU binary utilities package and the GNU GCC compiler package.

After compiling and installing the GNU GCC compiler package, the GCC compiler runtime libraries are placed in the **/local/lib/gcc-lib/3.3.1** directory. The compiler runtime library is contained in **libgcc.a** file.

The above procedure will not build the standard C++ or runtime libraries. In order to build these libraries the **TARGET_CONFIGDIRS** statement (and the **all-target**, **configure-target**, and **install-target** statements—must when compiling GNU GCC version 3.4.X or later) be edited in step 8 so that it contains the **libstdc++-v3** name and a number of include files must be placed in the **/usr/local/powerpc-ibm-eabi/include** directory. At a minimum, the following include files are necessary: **assert.h**, **ctype.h**, **errno.h**, **float.h**, **locale.h**, **math.h**, **signal.h**, **stdarg.h**, **stdio.h**, **stdlib.h**, **string.h**, **sys/types.h**, **time.h**, **types.h**. These files must be customized to work with the particular operating system that will run on the target PowerPC processor and should, therefore, be obtained from the operating system provider. Compiling and installing the GCC package with this configuration will place the C++ runtime library, **libsupc++a**, and the standard C++ library, **libstdc++a**, in the **/local/powerpc-ibm-eabi/lib** directory. If the standard C++ library will be used on the target system, the operating system must also implement all of the functions defined in include files.

6 Creating The HTML Documentation Files

The following instructions assume that the base directory name is **c:/data/gcc**. If different base directory name is used the directory names used in following commands must be changed.

The following commands can be used to create the GNU Compiler User's Manual in HTML format (**gcc.html**) from the documentation files:

1. **makeinfo -l/cygdrive/c/data/gcc/gcc-3.3.1/gcc/doc -l/cygdrive/c/data/gcc/gcc-3.3.1/gcc/doc/include --html --force /cygdrive/c/data/gcc/gcc-3.3.1/gcc/doc/gcc.texi --no-split**
2. **sed -e '9,38d' gcc.html > gcc.html2**
3. **mv gcc.html2 gcc.html**

The Following command can be used to create the GNU Preprocessor User's Manual in HTML format (**cpp.html**) from the documentation files:

1. **makeinfo -l/cygdrive/c/data/gcc/gcc-3.3.1/gcc/doc -l/cygdrive/c/data/gcc/gcc-3.3.1/gcc/doc/include --html --force /cygdrive/c/data/gcc/gcc-3.3.1/gcc/doc/cpp.texi --no-split**

The following instructions can be used to create the GNU Assembler User's Manual in HTML format (**as.html**) from the documentation files:

1. **cp /cygdrive/c/data/gcc/binutils-2.14/gas/doc/as.texinfo as.texinfo**
2. Create a text file named **sfile** with the following contents:

```
25 i \
@set AS gas \
@set TARGET PowerPC
```
3. **sed -f sfile as.texinfo > as2.texinfo**
4. **makeinfo -l/cygdrive/c/data/gcc/binutils-2.14/gas/doc --html --force as2.texinfo --no-split**

The following command can be used to create the GNU Linker User's Manual in HTML format (**ld.html**) from the documentation files:

1. **makeinfo -l/cygdrive/c/data/gcc/binutils-2.14/ld --html --force /cygdrive/c/data/gcc/binutils-2.14/ld/ld.texinfo --no-split**

The HTML documentation describing the command's usage can also be created. For example in order to create HTML documentation describing the **addr2line** program the following commands can be executed:

1. **man2html /usr/local/man/man1/powerpc-ibm-eabi-addr2line.1 > ppc-addr2line.html**
2. **sed -e '1,2d' ppc-addr2line.html > ppc-addr2line.html.1**
3. **sed -e '1,\$s/^G//g' ppc-addr2line.html.1 > ppc-addr2line.html**
4. **sed -e '1,\$s/.*man2html.*/g' ppc-addr2line.html > ppc-addr2line.html.1**
5. **mv ppc-addr2line.html.1 ppc-addr2line.html**

In the above example **ppc-addr2line** can be replaced in each step with **ar**, **as**, **c++filt**, **gcc**, **ld**, **nm**, **objcopy**, **objdump**, **ranlib**, **size**, **strings**, **readelf**, **strip** in order to create HTML command usage documentation for the other programs.

The HTML documentation describing the **cpp** program command's usage can also be created by executing the following commands:

Building a GNU Toolset For Use In Embedded PowerPC Cross Development

1. `man2html /usr/local/man/man1/cpp.1 > ppc-cpp.html`
2. `sed -e '1,2d' ppc-cpp.html > ppc-cpp.html.1`
3. `sed -e '1,$s/^G//g' ppc-cpp.html.1 > ppc-cpp.html`
4. `sed -e '1,$s/.*man2html.*/g' ppc-cpp.html > ppc-cpp.html.1`
5. `mv ppc-cpp.html.1 ppc-cpp.html`

7 Moving Executable Files to Another Machine

In order to transfer the PowerPC cross development toolset to another machine, the files listed in *Table 7-1* need to be copied:

Table 7-1. GCC Cross Development Toolset Files

Source File Directory and Name	Destination File Directory and Name
c:\cygwin\bin\cygwin1.dll	c:\gcc-toolset\bin\cygwin1.dll
c:\cygwin\bin\cygintl_2.dll	c:\gcc-toolset\bin\cygintl_2.dll
c:\cygwin\bin\cygiconv_2.dll	c:\gcc-toolset\bin\cygiconv_2.dll
c:\cygwin\bin\make.exe	c:\gcc-toolset\bin\gnumake.exe
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-addr2line.exe	c:\gcc-toolset\bin\ppc-addr2line.exe
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-ar.exe	c:\gcc-toolset\bin\ppc-ar.exe
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-as.exe	c:\gcc-toolset\bin\ppc-as.exe
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-c++.exe	c:\gcc-toolset\bin\ppc-c++.exe
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-c++filt.exe	c:\gcc-toolset\bin\ppc-c++filt.exe
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-cpp.exe	c:\gcc-toolset\bin\ppc-cpp.exe
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-g++.exe	c:\gcc-toolset\bin\ppc-g++.exe
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-gcc.exe	c:\gcc-toolset\bin\ppc-gcc.exe
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-ld.exe	c:\gcc-toolset\bin\ppc-ld.exe
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-nm.exe	c:\gcc-toolset\bin\ppc-nm.exe
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-objcopy.exe	c:\gcc-toolset\bin\ppc-objcopy.exe
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-objdump.exe	c:\gcc-toolset\bin\ppc-objdump.exe
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-ranlib.exe	c:\gcc-toolset\bin\ppc-ranlib.exe
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-readelf.exe	c:\gcc-toolset\bin\ppc-readelf.exe
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-size.exe	c:\gcc-toolset\bin\ppc-size.exe
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-strings.exe	c:\gcc-toolset\bin\ppc-strings.exe
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-strip.exe	c:\gcc-toolset\bin\ppc-strip.exe
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\include\altivec.h	c:\gcc-toolset\include\altivec.h
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\include\float.h	c:\gcc-toolset\include\float.h
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\include\iso646.h	c:\gcc-toolset\include\iso646.h
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\include\limits.h	c:\gcc-toolset\include\limits.h
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\include\ppc-asm.h	c:\gcc-toolset\include\ppc-asm.h

Building a GNU Toolset For Use In Embedded PowerPC Cross Development

Table 7-1. GCC Cross Development Toolset Files

Source File Directory and Name	Destination File Directory and Name
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\include\stdarg.h	c:\gcc-toolset\include\atdarg.h
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\include\stdbool.h	c:\gcc-toolset\include\atdbool.h
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\include\stddef.h	c:\gcc-toolset\include\stddef.h
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\include\syslimits.h	c:\gcc-toolset\include\syslimits.h
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\include\varargs.h	c:\gcc-toolset\include\varargs.h
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\crtbegin.o	c:\gcc-toolset\lib\crtbegin.o
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\crtbeginS.o	c:\gcc-toolset\lib\crtbeginS.o
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\crtbeginT.o	c:\gcc-toolset\lib\crtbeginT.o
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\crtend.o	c:\gcc-toolset\lib\crtend.o
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\crtendS.o	c:\gcc-toolset\lib\crtendS.o
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\crtendT.o	c:\gcc-toolset\lib\crtendT.o
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\crti.o	c:\gcc-toolset\lib\crti.o
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\crttn.o	c:\gcc-toolset\lib\crttn.o
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\libgcc.a	c:\gcc-toolset\lib\libgcc.a
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\ncrti.o	c:\gcc-toolset\lib\ncrti.o
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\ncrttn.o	c:\gcc-toolset\lib\ncrttn.o
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-as.exe	c:\gcc-toolset\lib\gcc-lib\as.exe
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\cc1.exe	c:\gcc-toolset\lib\gcc-lib\cc1.exe
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\cc1obj.exe	c:\gcc-toolset\lib\gcc-lib\cc1obj.exe
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\cc1plus.exe	c:\gcc-toolset\lib\gcc-lib\cc1plus.exe
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\collect2.exe	c:\gcc-toolset\lib\gcc-lib\collect2.exe
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\cpp0.exe	c:\gcc-toolset\lib\gcc-lib\cpp0.exe
c:\cygwin\usr\local\bin\powerpc-ibm-eabi-ld.exe	c:\gcc-toolset\lib\gcc-lib\ld.exe
c:\cygwin\usr\local\lib\gcc-lib\powerpc-ibm-eabi\3.3.1\tracpp0.exe	c:\gcc-toolset\lib\gcc-lib\tracpp0.exe
c:\cygwin\usr\local\powerpc-ibm-eabi\lib\ldscripts*	c:\gcc-toolset\lib\ldscripts\same as source file name
directory where the documentation files were created	c:\gcc-toolset\doc\same as source file name

Note: The **tracpp0.exe** and **cpp0.exe** files might not exist in the source directory. If the **tracpp0.exe** and **cpp0.exe** files are not present in the source directory they are not part of GCC compiler and do not have to be copied.

Note: With GNU GCC version 3.4.0 or later the **as.exe**, **cc1.exe**, **cc1obj.exe**, **cc1plus.exe**, **collect2.exe** and **ld.exe** files need to be placed in the **c:\gcc-toolset\lib\gcc-lib** directory (instead of **c:\gcc-toolset\lib\gcc-lib** directory).

Note: With GNU GCC version 3.4.0 or later the object file (for example **crtbegin.o**) and libraries (for example **libgcc.a**) need to be placed in the **c:\gcc-toolset\lib\gcc** directory (instead of **c:\gcc-toolset\lib** directory).

Building a GNU Toolset For Use In Embedded PowerPC Cross Development

The **c:\gcc-toolset** directory prefix on the target machine can be replaced with any other name except for the root directory (**c:**). The file name prefix of the executable files (**powerpc-ibm-eabi**) have been changed to (**ppc-**) in order to shorten the name of the executable files. After the files are transferred to the new machine, the path to the executable files (**c:\gcc-toolset\bin**) has to be added to the **PATH** environment variable.

If the GCC toolset is compiled on Windows NT/2000/XP the executable files will not be compatible with Windows 95/98/Me.

8 Using GNU Tools

The GNU GCC compiler can be invoked by executing the **powerpc-ibm-eabi-gcc** command. The generic compiler help information can be displayed by executing the **powerpc-ibm-eabi-gcc --help** command. The PowerPC specific help information can be displayed using the **powerpc-ibm-eabi-gcc --target-help** command. In general all the GNU tools will display help information if the tool is invoked with the **--help** option.

9 Summary

The GNU GCC compiler is a full featured ANSI C compiler with support for K&R C, as well as C++, Java, and Fortran. GCC support an extensive number of options for code generation, debugging information, and error checking. The GCC compiler and associated tools provide complete solution for embedded software development. The GCC compiler and the associated tools can be built as a cross compiler that runs in the Windows operating system and creates code for the PowerPC processor.

**Building a GNU Toolset For Use In Embedded PowerPC Cross Development**

Revision Log

Revision Date	Contents of Modification
09/19/2003	Initial version distributed for review.
09/28/2003	Review comments added. Added Moving Executable File to Another Machine section
10/06/2003	Change as a result of further review. Added section on 64-bit PowerPC
10/14/2003	Changes requested by legal
10/20/2003	Added note about the space character in the path where the files are built
12/15/2003	Added note about the prefix option
08/31/2004	Various minor and major corrections
09/08/2004	Grammar updates some instruction clarifications
09/08/2004	Added notes dealing with the GNU GCC compiler version 3.4.X
09/27/2004	Added notes about GNU GCC compiler version 3.4.X installation issues



Building a GNU Toolset For Use In Embedded PowerPC Cross Development

© Copyright International Business Machines Corporation 2003

All Rights Reserved

Printed in the United States of America September 27, 2004

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both.

IBM IBM Logo
PowerPC AIX

Cygwin is a trademark of Red Hat Incorporated in the United States, and/or other countries.

*AltiVec is a trademark of Motorola, Inc. used under license.

Solaris, Java, and all Java-based trademarks and logos are trademarks of Sun Microsystems, Incorporated in the United States and/or other countries.

Microsoft, Windows, Windows NT and the Windows logo are trademarks of Microsoft Corporation in the United States, and/or other countries.

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/OPEN Company Limited.

Other company, product and service names may be trademarks or service marks of others.

All information contained in this document is subject to change without notice. The products described in this document are NOT intended for use in applications such as implantation, life support, or other hazardous uses where malfunction could result in death, bodily injury, or catastrophic property damage. The information contained in this document does not affect or change IBM product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of IBM or third parties. All information contained in this document was obtained in specific environments, and is presented as an illustration. The results obtained in other operating environments may vary.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED ON AN "AS IS" BASIS. In no event will IBM be liable for damages arising directly or indirectly from any use of the information contained in this document.

IBM Microelectronics Division
2070 Route 52, Bldg 330
Hopewell Junction, NY 12533-6351

The IBM home page can be found at <http://www.ibm.com>

The IBM Microelectronics Division home page can be found at <http://www.ibm.com/chips>