

# **GnuCOBOL 1.1 [20JAN2014 version]**

## **Build Guide for MinGW - Minimalist GNU for Windows**

GnuCOBOL is copyrighted by:  
Keisuke Nishida  
Roger While

This document was prepared by: Arnold J. Trembley (arnold.trembley@att.net)  
and last updated Tuesday, 11 November 2014.

Please refer to the excellent manual written by Gary Cutler (CutlerGL@gmail.com):

### **OpenCOBOL-1.1-06FEB2009-Build-Guide-For-MinGW.pdf**

<http://www.mediafire.com/file/csc7gufle77tgsg/OpenCOBOL-1.1-06FEB2009-Build-Guide-For-MinGW.pdf>

This manual is also included in the GnuCOBOL 1.1 MinGW installer and can be found in the "documentation" folder.

I used Gary Cutler's document to build GnuCOBOL 1.1 using MinGW. This document will be much less detailed, and will emphasize what was different about building GnuCOBOL 1.1 as opposed to building OpenCOBOL 1.1.

This document will also cover building a Windows installer program for MinGW GnuCOBOL 1.1.

A separate document will be included in the GnuCOBOL documentation folder for testing the compiler with the GnuCOBOL test suite and the NIST COBOL85 test suite.

Simon Sobisch of the GnuCOBOL project on Sourceforge.net was extremely helpful to me whenever I encountered difficulties, especially in running the compiler test suites. Brian Tiffin also provided encouragement and assistance. The GnuCOBOL/OpenCOBOL project can be found at:  
<https://sourceforge.net/projects/open-cobol/>

### **Required Components:**

Based on Gary Cutler's build guide, you will need to find and install the following components in order to build the GnuCOBOL 1.1 compiler:

1. MinGW - Minimalist GNU for Windows
2. MSYS 1.0.11 - BASH shell for MinGW

3. GNU Multiple-Precision Arithmetic package (gmp)lib)
4. PDCurses 3.4 - used for screen read and write.
5. Berkeley Database (BDB) package from Oracle
6. GnuCOBOL 1.1 source code from the OpenCOBOL project on Sourceforge.net

You may want to download all these packages first and make your own backups before starting the GnuCOBOL build process. Follow Gary Cutler's detailed instructions for installing and building each component, except for specific instruction changes in this document for GnuCOBOL 1.1.

### Licensing:

This document will not go into the licenses in detail. Please refer to Gary Cutler's document. The GnuCOBOL compiler is licensed under the GNU General Public License (GPL) version 2, and the runtime libraries are licensed under the GNU Lesser General Public License (LGPL) version 2.1. The Berkeley Database (BDB) package from Oracle has some license restrictions related to distribution of compiled COBOL programs using BDB that could require distributing your COBOL source code.

### Download the packages:

The **MinGW** software package can be downloaded from:

<https://sourceforge.net/projects/mingw/files/>

Download the file named "mingw-get-setup.exe". This should be the 32-bit version of MinGW. MinGW is a Unix-like environment for Windows needed to run the GCC (Gnu Compiler Collection) to build the GnuCOBOL compiler. The GnuCOBOL compiler will translate COBOL source code into C source code, and the MinGW GCC C compiler will translate the intermediate code into executable code.

Gary Cutler's instructions included the following step immediately after installing C:\MinGW:  
Copy "C:\MinGW\libexec\gcc\mingw32\3.4.5\cc1.exe" to "C:\MinGW\bin\cc1.exe".

This needed to be changed for the newer version number of the MinGW GCC compiler collection, as follows: Copy "C:\MinGW\libexec\gcc\mingw32\4.8.1\cc1.exe" to "C:\MinGW\bin\cc1.exe".

The **MSYS 1.0.11** package can be difficult to find. You want the older version that is packaged as a Windows install file. The file should be named "MSYS-1.0.11.exe" and it can found here:

<https://sourceforge.net/projects/mingw/files/MSYS/Base/msys-core/msys-1.0.11/>

The MSYS 1.0.11 Windows installer will create a Unix-like bash shell needed for building GnuCOBOL and its related packages.

The **GMP** GNU Multiple-Precision Arithmetic Library can be downloaded from:

<https://gmplib.org/#DOWNLOAD>

The most current GMP version is in a file named "gmp-6.0.0a.tar.lz", but MSYS could not unpack a "tar.lz" file. I used free "lzip.exe" for windows to convert "gmp-6.0.0a.tar.lz" into "gmp-6.0.0a.tar". Then I had to alter the commands to build gmplib after copying it to MSYS:

```
cd /mingw/MSYS
tar xf gmp*.tar /* for non-compressed tar */
rm gmp-*.tar

cd gmp*
./configure --prefix=/mingw --enable-shared --disable-static
make
make install
make check
```

Building GMP took quite a long time and produced thousands of lines of informational messages.

The **PDCurses 3.4** package is in a file named "PDCurses-3.4.tar.gz" which can be downloaded from: <https://sourceforge.net/projects/pdcurses/files/pdcurses/3.4/>

The PDCurses package is used for COBOL SCREEN-SECTION and extended console input-output support (DISPLAY/ACCEPT AT/WITH). This package did not need any changes from Gary Cutler's original instructions.

The **Berkeley Database (DB)**, file is named "db-6.1.19.NC.tar.gz", the most current version as of October, 2014, and can be downloaded from: <http://www.oracle.com/technetwork/database/database-technologies/berkeleydb/downloads/index.html>

The Berkeley Database (DB) package provides indexed file access for the GnuCOBOL compiler.

**GnuCOBOL 1.1** source code, version dated 20JAN2014, is found in a file named "gnu-cobol-1.1.tar.gz" which can be downloaded from this Sourceforge webpage: <https://sourceforge.net/projects/open-cobol/files/gnu-cobol/1.1/>

Gary Cutler's manual included the following MSYS commands for unpacking the OpenCOBOL source code after downloading and copying into the tarball into C:\MinGW\MSYS:

```
cd /mingw/MSYS
tar xzf open*.tar.gz
rm open*.tar.gz
```

The tar command would need to be changed to "tar xzf gnu\*.tar.gz" because of the different name for the GnuCOBOL tarball.

```
cd /mingw/MSYS
```

```
tar xzf gnu*.tar.gz
rm gnu*.tar.gz
```

## Applying Patches

Gary Cutler's manual included extensive notes on modifying four source files for the OpenCOBOL compiler to apply patches. I reviewed the affected source code modules and only the patch to `termio.c` did not appear to be already applied. It is a fairly easy source code patch, but I recommend making a backup of `termio.c` in case of any problems. The patch removes the trailing LF character from an ACCEPT into a buffer that is longer than the ACCEPT'ed data. Applying this patch corrected several failed test cases in the NIST COBOL85 test suite.

The next step is to build the GnuCOBOL compiler from source code, changing the name from OpenCOBOL to GnuCOBOL, using the following commands in MSYS:

```
cd /mingw/MSYS/gnu*
./configure --prefix=/mingw
make
make install
```

## Packaging the C:\GnuCOBOL folder

Gary Cutler's manual included instructions for building the C:\OpenCOBOL folder, where a couple of problems were encountered. First, due to an error on my part, I did not get subfolders copied in the directory structure. Second, it appears that due to the newer version of MinGW there was an additional folder, named "libexec", to be copied into the C:\GnuCOBOL folder that did not exist for OpenCOBOL 1.1.

Assuming all the previous steps completed successfully, here are the logical steps for building the C:\GnuCOBOL folder:

```
Create the C:\GnuCOBOL folder      (mkdir C:\GnuCOBOL)

copy C:\MinGW\bin                  to C:\GnuCOBOL\bin
copy C:\MinGW\share\open-cobol\config to C:\GnuCOBOL\config
copy C:\MinGW\share\open-cobol\copy  to C:\GnuCOBOL\Copy
copy C:\MinGW\lib                  to C:\GnuCOBOL\lib
copy C:\MinGW\libexec              to C:\GnuCOBOL\libexec
copy C:\MinGW\include              to C:\GnuCOBOL\include
```

Note that these are logical instructions, not explicit copy command syntax. You can also use Windows Explorer to copy these folders. The important thing is to be sure that all subfolders are copied.

The Windows CMD.EXE commands would look like this:

```
mkdir C:\GnuCOBOL
xcopy C:\MinGW\bin\*. * c:\GnuCOBOL\bin\ /s /e
xcopy C:\MinGW\share\gnu-cobol\config\*. * C:\GnuCOBOL\config\ /s /e
xcopy C:\MinGW\share\gnu-cobol\copy\*. * C:\GnuCOBOL\copy\ /s /e
xcopy C:\MinGW\lib\*. * C:\GnuCOBOL\lib\
xcopy C:\MinGW\lib\gcc\*. * C:\GnuCOBOL\lib\gcc\ /s /e
xcopy C:\MinGW\libexec\gcc\*. * C:\GnuCOBOL\libexec\gcc\ /s /e
xcopy C:\MinGW\include C:\GnuCOBOL\include\ /s /e
```

The resulting C:\GnuCOBOL folder is sufficient for compiling COBOL programs, but some additional files should also be added.

The following files were added to the "C:\GnuCOBOL\copy" folder (copied from Gary Cutler's "C:\OpenCOBOL\copy" folder in his zip file distribution):

```
FileStat-Msgs.cpy (used by Ocic.cbl)
sqlca.cpy
sqlda.cpy
STREAMIOcb.cpy
STREAMIOError.cpy
```

The entire "C:\GnuCOBOL\documentation" folder and its subfolders were copied from Gary Cutler's "C:\OpenCOBOL\documentation" folder, found his OpenCOBOL 1.1 zip file, and some minor changes were made.

This included four excellent manuals in PDF format written by Gary Cutler:

```
OpenCOBOL-1.1-06FEB2009-Build-Guide-For-MinGW.pdf
OpenCOBOL-1.1-06FEB2009-OCic.pdf
OpenCOBOL-1.1-06FEB2009-Programmers-Guide.pdf
OpenCOBOL-1.1-06FEB2009-Textpad-Integration.pdf
```

I deleted the C:\GnuCOBOL\documentation\OpenCOBOL 1.1" folder and its place created the "C:\GnuCOBOL\documentation\GnuCOBOL 1.1" folder. The "gnu-cobol.info.txt" file, found in the "C:\MinGW\MSYS\gnu-cobol-1.1\texi" folder was copied into the "documentation\GnuCOBOL 1.1" folder.

The "C:\OpenCOBOL\samples" folder was copied from Gary Cutler's OpenCOBOL distribution into the "C:\GnuCOBOL\samples" folder.

I modified the Ocic.cbl program from the "samples" folder to change the report title from "OpenCOBOL 1.1" to "GnuCOBOL 1.1". Ocic.cbl was recompiled using the new GnuCOBOL 1.1 compiler and Ocic.exe was stored in the "C:\GnuCOBOL\bin" folder.

The "C:\GnuCOBOL\tests" folder was given to me by Simon Sobisch of the GnuCOBOL/OpenCOBOL project on Sourceforge.net. There is a separate guide for running the GnuCOBOL compiler test suite and the NIST (National Institute for Standards and Technology) COBOL85 test suite.

## **Additional files**

Several additional files were added to the "C:\GnuCOBOL" folder:

```
config.log
GnuCOBOL-1.1-20JAN2014-MinGW-Setup.iss
GNU-viking.ico
GPL-v2.txt
LGPL-v2.txt
License-Agreement.txt
OpenCOBOL-1.1-06FEB2009-MinGW-Distribution-README.pdf
Post-Install.txt
Pre-Install.txt
setgcglobal.cmd
setgcsession.cmd
setperl.cmd
testfunc.cob
TestGC.cmd
```

The "config.log" file was copied from the "C:\MinGW\MSYS\gnu-cobol-1.1" folder. It is used along with the test suites in the "tests" folder.

The OpenCOBOL 1.1 README PDF file was copied from Gary Cutler's OpenCOBOL distribution, and was very helpful.

The following files were written by me and intended for getting started with GnuCOBOL 1.1:

```
setgcglobal.cmd
setgcsession.cmd
setperl.cmd
testfunc.cob
TestGC.cmd
```

Simon Sobisch recommended several changes to these files to improve them. They contain comments which should make their purpose clear. The "setperl.cmd" file is only required if you plan to run either the GnuCOBOL or NIST COBOL85 test suites.

The following files were included as part of the GnuCOBOL Windows Installer generated by INNO Setup.

```
GnuCOBOL-1.1-20JAN2014-MinGW-Setup.iss
GNU-viking.ico
GPL-v2.txt
LGPL-v2.txt
License-Agreement.txt
Post-Install.txt
Pre-Install.txt
```

Simon Sobisch provided extensive guidance in preparing the License agreement, pre-install, and post-install text files used in the Installer.

## **Building a Windows Installer using INNO Setup**

INNO Setup is available for free download from:

<http://www.jrsoftware.org/isinfo.php>

Extensive help is available at the website and with the INNO Setup software. I also used an add-on package named INNO Script Studio as a GUI interface for editing and compiling the installer program. It is available for free download from:

<https://www.kymoto.org/products/inno-script-studio/>

"GnuCOBOL-1.1-20JAN2014-MinGW-Setup.iss" is the script file, or source code, for building the GnuCOBOL 1.1 Windows installer file.

The Installer script assumes the source files are in a folder named "C:\GnuCOBOL".

The script command "OutputBaseFilename=GnuCOBOL-1.1-20JAN2014-MinGW-Setup" specifies the name of the generated setup.exe file.

The script command "OutputDir=c:\work" specifies where the generated setup.exe file will be stored.

Even though the compiler was built from a folder named "C:\GnuCOBOL", the generated Windows setup program allows the compiler to be installed to any available windows drive (not just C:), and the path and folder can be changed at install time. The title of the Windows program group for GnuCOBOL can also be changed at install time. Be careful not to use a path name with embedded spaces (like C:\Program Files") because it may cause problems for GnuCOBOL and MinGW programs.

If you choose to build your own installer, you should disconnect from the internet and temporarily disable your anti-virus or security software while the installer program is being compiled. Otherwise you could see the Inno compile fail with a write error and no setup.exe file generated.

The generated GnuCOBOL installer program is packaged in a zip file and available at:

<http://www.arnoldtrembley.com/GnuCOBOL-MinGw-Installer.zip>

You would need to download the zip file, extract the "GnuCOBOL-1.1-20JAN2014-MinGW-Setup.exe" program and then run it to install MinGw GnuCOBOL 1.1 (after first checking if for viruses).