

GnuCOBOL 3.x or 4.x Build Guide for MSYS2 64-bit - Minimalist GNU for Windows GCC version (MinGW64) "12.2.0"

cobc (GnuCOBOL) 3.2 Preview (commit r4918)

Copyright (C) 2020 Free Software Foundation, Inc.

Written by Keisuke Nishida, Roger While, Ron Norman, Simon Sobisch, Edward Hart.

This document was prepared by: Arnold J. Trembley (arnold.trembley@att.net),
based on extensive work by Chuck Haatvedt (chuck.haatvedt@gmail.com),
and last updated Thursday, 12 January 2023.

Special thanks go to **Chuck Haatvedt**, who wrote the shell scripts “build_toolchains.sh”, “build-x32.sh” and “build-x64.sh” in their entirety. He spent weeks testing and refining the scripts. **Simon Sobisch**, the GnuCOBOL project leader, also provided Chuck with assistance and made changes to the GnuCOBOL source packages to facilitate this MSYS2 build process. I am deeply indebted to both Chuck and Simon for their work.

These scripts build GnuCOBOL with the following components at the time these scripts were created:

Sequential file handler	: built-in
Indexed file handler	: BDB, version 18.1.40
Mathematical library	: GMP, version 6.2.1
XML library	: libxml2, version 2.10.3
JSON library	: cJSON, version 1.7.15
Extended screen I/O	: pdcurses, version 4.3.5 (CHTYPE=64, WIDE=0, UTF8=0)

Note that if you would like to build with other configurations, please contact either Chuck or me and we will attempt to assist you. This process can also be adapted to build GnuCOBOL 4.x.

STEP01 - Download MSYS2/MinGW64

Go to <https://www.msys2.org/> and download "msys2-x86_64-*.exe"
Look for version "msys2-x86_64-20221216.exe" or newer.

STEP02 - Disable Anti-Virus (optional)

Close all web browsers and disable real-time Anti-Virus scanning. This may make the build process go more smoothly, but you cannot disconnect from the internet because the scripts download components from multiple trusted sources.

STEP03 - Install MSYS2

Install "msys2-x86_64-*.exe" into the default "**C:\msys64**" folder.
Accept the default start menu shortcut folder "MSYS2 64bit". Follow the installation instructions from the web page through step 4.

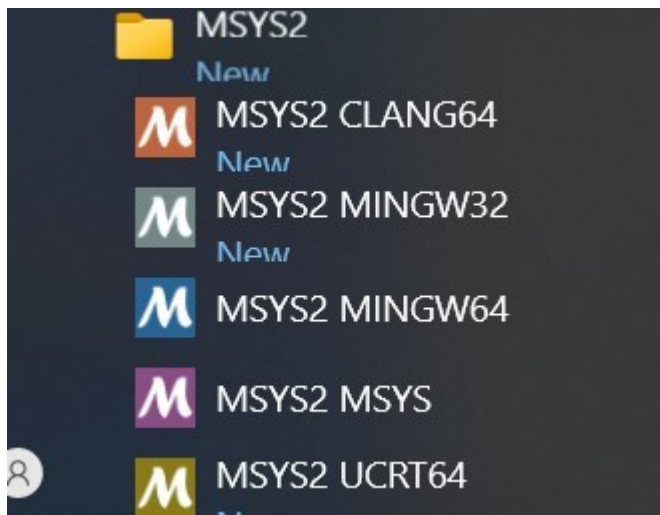
When you see the MSYS2 UCRT64 environment, type "exit".

You can actually install MSYS2 to ANY folder on ANY drive letter, but I recommend using the suggested defaults.

STEP04 – Update MSYS2 tool chain

Copy Chuck Haatvedt's "**build_toolchains.sh**" script into the "[C:\msys64](#)" folder. Note the underscore versus hyphen. Then open MSYS2 MSYS per Chuck's instructions below.

In the Windows Start menu navigate to the **MSYS2 MSYS** folder and double click on the MSYS2 MSYS icon. This will open a terminal with the MSYS2 MSYS shell.



Then start the script in that shell using the following command:

```
source /build_toolchains.sh | tee build_toolchains.txt
```

After it finishes successfully, you can close that shell window by typing "exit".

You can review the "build_toolchains.txt" log found in "C:\msys64\home\{username}" folder, which can be especially helpful if you encounter errors.

STEP05 – Prepare X32 GnuCOBOL build

Copy Chuck Haatvedt's "**build-x32.sh**" script (note hyphen versus underscore) into the "**C:\msys64\mingw32\bin**" folder. It must go into the correct \bin sub-folder.

Modify the script as needed for number of processors (cpu_count). If you plan to run x32 and x64 builds concurrently, set the cpu_count to half the total number of processor threads available. Even if you run them separately, you should not set the cpu_count higher than the maximum number of threads available.

You can modify the script for the version you want of PDCursesMod, GnuCOBOL, or GCSORT, or you can modify the script to use a local copy of one of those source components.

I used cpu_count=4 (because my PC has 4 cores with 2 threads each, for 8 threads total).

I took the default for PDCursesMod, which is currently "release 4.3.5".

I took the default for GnuCOBOL, which was SVN current commit, (In this case r4918, dated 2023-01-06).

I took the default for GCSORT which is current trunk (r1026)

STEP06 – Prepare X64 GnuCOBOL build

Copy Chuck Haatvedt's "**build-x64.sh**" script (note hyphen versus underscore) into the "**C:\msys64\mingw64\bin**" folder. It must go into the correct \bin sub-folder.

Modify the script as needed for number of processors (cpu_count). If you plan to run x32 and x64 builds concurrently, set the cpu_count to half the total number of processor threads available. Even if you run them separately, you should not set the cpu_count higher than the maximum number of threads available.

I found that sometimes the x64 build would hang at the end of the GnuCOBOL "make check" step, so therefore I reduced my "cpu_count" from 4 to 2.

You can modify the script for the version you want of PDCursesMod, GnuCOBOL, or GCSORT, or you can modify the script to use a local copy of one of those source components.

I took the default for PDCursesMod, which is currently "release 4.3.5".

I took the default for GnuCOBOL, which was SVN current commit, (In this case r4896, dated 2022-12-26).

I took the default for GCSORT which is current trunk (r1026).

NOTE: You do not have to build both 32-bit and 64-bit versions of GnuCOBOL concurrently. You can build them separately. Also, you can choose to build only a 32-bit or 64-bit version based on your needs.

STEP07 – Run the X32 GnuCOBOL build

Then open / select the **MSYS2 MINGW32** icon from STEP4 and run the script using the following command:

source build-x32.sh | tee build-x32.txt (note: no initial “/” needed)

You can review the “build-x32.txt” log file in your C:\msys64\home\user\ folder, , which can be especially helpful if you encounter errors.

NOTE: if you encounter any errors, you can simply run the script again as it first deletes the folder used to build each of the components.

STEP08 – Run the X64 GnuCOBOL build

Open / select the **MSYS2 MINGW64** icon from STEP4 and run the script using the following command:

source build-x64.sh | tee build-x64.txt (note: no initial “/” needed)

You can review the “build-x64.txt” log file in your C:\msys64\home\user\ folder, , which can be especially helpful if you encounter errors.

NOTE: if you encounter any errors, you can simply run the script again as it first deletes the folder used to build each of the components.

NOTE: You can run the X32 and X64 builds concurrently, since they each have their own shell window.

STEP09 – Prepare Binary Distribution Folders

At this point you should have working GnuCOBOL compilers, which can be found in the following folders:

C:\msys64\home\(\username)\x32\gnucobol\GnuCOBOL_mingw	691.7 mb
C:\msys64\home\(\username)\x32\gnucobol\GnuCOBOL_mingw_dbg	710.8 mb
C:\msys64\home\(\username)\x64\gnucobol\GnuCOBOL_mingw	711.2 mb
C:\msys64\home\(\username)\x64\gnucobol\GnuCOBOL_mingw_dbg	734.3 mb

Be aware that the “x32” folders are 32-bit compilers and the “x64” folders are 64-bit compilers.

In spite of the “_dbg” suffix, ALL four versions include FULL support for the Gnu Debugger and the code coverage testing tool, plus PERL and Python. The “_dbg” folders contain additional files that could help diagnose problems building GnuCOBOL with MSYS2, but which are not needed for COBOL programmers.

Here is a typical directory listing example using the 32-bit compiler folder:

Directory of C:\msys64\home\(\username)\x32\gnucobol\GnuCOBOL_mingw

```

12/27/2022 09:55 PM <DIR> .
12/27/2022 09:55 PM <DIR> ..
12/27/2022 08:18 PM <DIR> bin
12/27/2022 07:58 PM <DIR> config
12/27/2022 07:58 PM <DIR> copy
12/27/2022 07:59 PM <DIR> extras
12/27/2022 08:14 PM <DIR> include
12/27/2022 08:18 PM <DIR> lib
12/22/2022 09:53 PM <DIR> libexec
12/22/2022 09:53 PM <DIR> locale
12/27/2022 08:15 PM <DIR> share
12/27/2022 08:15 PM          94 BUGS.txt
12/27/2022 08:15 PM      48,247 ChangeLog.txt
12/27/2022 08:15 PM       9,020 ChangeLog_bin.txt
12/27/2022 08:15 PM     350,613 ChangeLog_cobc.txt
12/27/2022 08:15 PM     201,349 ChangeLog_libcob.txt
12/27/2022 08:15 PM     153,013 config.log
12/27/2022 08:15 PM     23,415 COPYING.DOC.txt
12/27/2022 08:15 PM      7,817 COPYING.LESSER.txt
12/27/2022 08:15 PM     35,823 COPYING.txt
12/27/2022 08:18 PM     355,578 gnucobol.pdf
12/27/2022 08:15 PM     57,063 NEWS.txt
12/27/2022 08:15 PM      1,224 NIST_summary.log
12/27/2022 08:15 PM      2,364 README.txt
12/27/2022 08:15 PM      2,872 set_env.cmd
12/27/2022 08:15 PM     288,746 testsuite.log
12/27/2022 08:15 PM      3,372 THANKS.txt

```

You can shrink these compiler folders by about 30 megabytes if you delete the “**man**” folders found on the following paths:

```

C:\msys64\home\(\username)\x32\gnucobol\GnuCOBOL_mingw\share\man
C:\msys64\home\(\username)\x32\gnucobol\GnuCOBOL_mingw_dbg\share\man
C:\msys64\home\(\username)\x64\gnucobol\GnuCOBOL_mingw\share\man
C:\msys64\home\(\username)\x64\gnucobol\GnuCOBOL_mingw_dbg\share\man

```

The “BUGS.txt” file can also be deleted.

In order to build a complete binary package, you should include the following files from the “extra-files” folder of the Build Kit zip package:

AUTHORS.txt
cobc-help.txt (can be built locally, using “cobc -vvh > cobc-help.txt”
gcsort-help.txt
gcx.cmd (compile an .exe, can be customized locally)
STARTHERE.txt
testfunc.cob
TestGC.cmd
VScobc.cmd
Vscobcrun.cmd

You should also add a **\docs** sub-folder containing the following manuals from the Build Kit ZIP package:

Directory of C:\GC32-BDB\docs

GCSORT_Manual.pdf
GnuCOBOL 3.2 Programmer's Guide.pdf
GnuCOBOL 3.2 Programmer's Reference.pdf
GnuCOBOL 3.2 Quick Reference.pdf
GnuCOBOL 3.2 Sample Programs.pdf

At this point you have a complete GnuCOBOL compiler binary folder that can be zipped and shared.

=====

As of 12 January 2023, the MSYS2 64-bit and 32-bit GnuCOBOL 3..2.0 binaries can be downloaded from the following addresses:

<http://www.arnoldtrembley.com/GC32M-BDB-x64-r4918-rename-7z-to-exe.7z>

<http://www.arnoldtrembley.com/GC32M-BDB-x32-r4918-rename-7z-to-exe.7z>

Due to a security restriction from my web hosting service I cannot directly host “.exe” files. So the new files have been renamed with “.7z” as their file extension. After downloading they can be opened and extracted using 7-Zip, or the windows file extensions can be renamed from “.7z” to “.exe”, allowing the files to be used as self-extracting archives. The self-extracting file will prompt you to supply a folder name for the compiler. It can also be installed to a drive other than your C: drive.

7-Zip is open source software available from <http://www.7-zip.org/>

In the future, new binaries will be added to the following page:

<http://www.arnoldtrembley.com/GnuCOBOL.htm>

For additional assistance, please feel free to try the GnuCOBOL Forums:

<https://sourceforge.net/p/open-cobol/discussion/>

=====end=====