OpenCOBOL 1.1

[06FEB2009 Version]

MinGW Distribution README

OpenCOBOL is copyrighted by:
Keisuke Nishida
Roger While

Distribution and documentation prepared by:
Gary Cutler
CutlerGL@gmail.com

Contents

Introduction

OpenCOBOL is a COBOL compiler and runtime environment. The OpenCOBOL compiler generates C code which is automatically compiled and linked by the GNU C compiler (gcc) and the GNU linker/loader (ld). All required GNU language components needed by OpenCOBOL are provided with this OpenCOBOL distribution in ready-to-use form.

THIS DISTRIBUTION OF OPENCOBOL HAS BEEN CREATED SPECIFICALLY FOR USE IN A MICROSOFT WINDOWS ENVIRONMENT. It has been tested under both Windows XP and Windows Vista.

To get started with OpenCOBOL, it is HIGHLY recommended that you check out the following documents (included in this distribution), found in the "Documentation" folder in addition to the one you're reading now:

- **OpenCOBOL 1.1\open-cobol.info.txt**

  This is the standard documentation released with OpenCOBOL. It instructs the reader on how to run the compiler and execute applications. It is (unfortunately) oriented around UNIX users but there is still much valuable information here.

- **OpenCOBOL-Programmers-Guide-06FEB2009-Release.pdf**

  This is a combination Reference Manual and Users Guide for OpenCOBOL. It documents the specific version of OpenCOBOL 1.1 defined by the February 6, 2009 tarball (i.e. "build") available from the OpenCOBOL website (http://www.opencobol.org).

This distribution was prepared and documented by:

Gary Cutler

CutlerGL@gmail.com

I am also the author of the OpenCOBOL Programmers Guide and the OCic program (not OpenCOBOL itself but the screen-based interface to the OpenCOBOL compiler included with this distribution). The OCic utility is documented in a separate document.

What’s Included in this Distribution

Software

All software included in this distribution is completely free of charge and you have broad rights regarding its use, redistribution and even modification.[1]

Software products included in this distribution are:

- **OpenCOBOL V1.1 (Preliminary) – Feb 06 2009 Build**

  OpenCOBOL is an open-source COBOL compiler. OpenCOBOL implements a substantial part of the COBOL 85 and COBOL 2002 standards, as well as many extensions of the existent COBOL compilers.

  OpenCOBOL translates COBOL into C and compiles the translated code using the system’s native C compiler.

  This is a pre-release version of the OpenCOBOL product. I chose to include it rather than the current release (1.0) because of it’s inclusion of the 2002-standard SCREEN SECTION, which enables OpenCOBOL programs to easily incorporate Textual User Interfaces (TUIs).

  The compiler is licensed under GNU General Public License.[2]

  The run-time library is licensed under GNU Lesser General Public License.[3]

  Visit the OpenCOBOL website (http://www.opencobol.org) for more information.

- **Berkeley Database V4.5.20**

  Berkeley Database (BDB) has been included into the OpenCOBOL project for its file-management functionality. BDB provides INDEXED file support as well as SORT/MERGE functionality to OpenCOBOL.

  From the BDB website (http://www.oracle.com/database/berkeley-db/db/index.html):

  > “Oracle Berkeley DB[4] is the industry-leading open source, embeddable, key-value database that provides developers with fast, reliable, local persistence with zero administration. Oracle Berkeley DB is a library that links directly

[1] Basically, you are free to redistribute “as-is”, modify, redistribute modified versions and use all software for your own purposes including the development of your own software which you intend to distribute. You may not, however, use the software in this distribution to develop and market so-called “proprietary” software. This brief summary is NOT a legal statement of your rights – READ THE ACTUAL LICENSES!

[2] A copy of the GPL can be found in the file GPL.txt, located in the same folder as this README file

[3] A copy of the LGPL can be found in the file LGPL.txt, located in the same folder as this README file

[4] Copyright (c) 1990-2006 Oracle Corporation. All rights reserved.

> into your application. Your application makes simple function calls, rather than sending messages to a remote server, eliminating the performance penalty of client-server architectures. Oracle Berkeley DB eliminates the overhead of SQL query processing, enabling applications with predictable access patterns to run faster."

The phrase "key-value database" in the statement above tells the tale – BDB provides a library of access methods that enable file records to be accessed based upon symbolic key values imbedded in those records; although fundamental to the implementation of so-called relational database tables, this capability is also the foundation of COBOL INDEXED files.

Although copyrighted to Oracle Corporation, BDB is made available free-of-charge and is redistributable under the terms set forth in its license.[5] The following statement, along with the BDB copyright at the bottom of this page[6], satisfies the terms of that license:

Source code to BDB is available from the BDB website. Source code to OpenCOBOL V1.1 is available from the OpenCOBOL website (**http://www.opencobol.org**).

- **GNU Multiple Precision math package (GMP) V4.3.1**

  GMP is a free library for arbitrary precision arithmetic, operating on signed integers, rational numbers, and floating point numbers. There is no practical limit to the precision except the ones implied by the available memory in the machine GMP runs on. GMP has a rich set of functions, and the functions have a regular interface.

  It is GMP that enables OpenCOBOL to perform the USAGE DISPLAY and USAGE COMP-3 (packed decimal) arithmetic that we've all come to expect from any COBOL implementation.

  This package is licensed under the terms of the GNU Lesser General Public License.

  See the GMP website (http://gmplib.org) for more information and source code.

- **Public-domain "Curses" (PDCurses) V4.3**

  PDCurses is a public domain curses library[7] for DOS, OS/2, Win32, X11 and SDL, implementing most of the functions available in X/Open and System V R4 curses. It supports many compilers for these platforms.

  This package is what enables OpenCOBOL to support the functionality of a SCREEN SECTION and extended (cursor-addressable, visual highlighting) ACCEPT and DISPLAY statements.

  This too is licensed under the terms of the GNU Lesser General Public License.

  See the PDCurses website at **http://pdcurses.sourceforge.net** for additional information and source code.

- **Minimalist GNU For Windows (MinGW) V5.1.4**

[5] The BDB license is available in the file **Documentation\Berkeley DB\license**\license_db.html, included in this distribution

[6] Copyright (c) 1990-2006 Oracle Corporation. All rights reserved.

[7] "Curses" packages provide full-screen character-oriented input and output capabilities for the management of UNIX console windows.

MinGW is a port of the GNU Compiler Collection (GCC), and GNU binutils, for use in the development of native Microsoft Windows applications.

MinGW provides the C-compiler that will be needed to complete the compilation of OpenCOBOL source programs. It also provided the pseudo-UNIX command shell environment needed to build the various components of this distribution from their respective sources.

Since most GNU software was developed for the UNIX world, getting GNU software to run under Windows required special consideration in the form of the development of a special "MinGW" DLL named "mingwm10.dll" (included in this distribution) which must be present on any system which will run the GCC or any application program produced by the GCC (including the OpenCOBOL compiler).

From the MinGW website:

> “The MinGW base runtime package has been placed in the public domain, and is not governed by copyright. This basically means that you can do what you like with the code.”

The "mingwm10.dll" discussed above is part of that base runtime package.

All of the GNU development tools, such as GNU binutils, GCC, GDB and GNU Make, are governed by the terms of the GNU General Public License.

Visit the MinGW website at http://www.mingw.org for more information. Additionally, for more information on GNU and their open-source philosophy toward software distribution, visit http://gcc.gnu.org.

Documentation

In addition to the document you are reading now, you will find a “**Documentation**” folder in the distribution media. This folder will contain sub-folders, as follows:

- **Berkeley DB**
- **GMP**
- **MinGW and GNU**
- **OpenCOBOL 1.1**

These folders will contain documentation files included with each of the components that make up this software distribution.

In addition, the Documentation folder contains the following documents in Adobe PDF format:

- **OpenCOBOL-1.1-06FEB2009-Programmers-Guide.pdf**

  This is a combination Reference Manual and Users Guide for OpenCOBOL. It documents the features and syntax available in the 06FEB2009 tarball release of OpenCOBOL 1.1.

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

Want to build your own OpenCOBOL distribution from source? Curious how this distribution was created? If so, then this is the document for you! A complete step-by-step "how to" guide to building your own binaries for OpenCOBOL and the required co-requisite packages.

- **OpenCOBOL-1.1-06FEB2009-OCic.pdf**

  OCic is an OpenCOBOL program that provides a full-screen, interactive interface to the standard OpenCOBOL compiler (cobc). It also generates extremely useful cross-reference listings of your OpenCOBOL programs, showing not only where your data-names and procedure-names are defined and referenced but (in the case of data names) where they are GETTING MODIFIED in your code as well! This document describes the utility and its use.

- **OpenCOBOL-1.1-06FEB2009-Textpad-Integration.pdf**

  This documents how to integrate the 06FEB2009 tarball release of OpenCOBOL 1.1 into the "Textpad" text editor for Windows.

Samples

The "Samples" folder in the distribution media contains source code for a number of sample OpenCOBOL programs, including the following helpful utilities:

OCic.cbl Source code to the OCic utility (see OCIC.pdf).

SCREENIO.cbl Source code to a useful subroutine that automates many of the chores in dealing with bytestream files. Read the comments at the start of the code for details.

WINSYSTEM.cbl Source code to a useful subroutine that enables you to submit commands to a Windows shell if you are running on a Cygwin OpenCOBOL installation (the "SYSTEM" built-in subroutine submits commands to the native shell for the OpenCOBOL build environment you are using – for a Cygwin build, that'll be a Unix-like "bash" shell, not Windows). You won't need it for this distribution of OpenCOBOL, but if your programs are coded to CALL the WINSYSTEM routine rather than the built-in SYSTEM routine, they'll work with any Windows build - MinGW, native Windows or Cygwin!

Copybooks

The "Copy" folder in the distribution media contains COBOL copybooks. The ones named entirely in lowercase are the ones packaged with OpenCOBOL by the authors of OpenCOBOL, while the ones names in mixed case were created by me as useful utility copybooks or in support of code included in the "Samples" folder.

Installing OpenCOBOL on your PC

Follow these simple steps to install this OpenCOBOL distribution onto your computer:

1. Unzip the entire **OpenCOBOL.06FEB2009.mingw.zip** file contents to **x:\OpenCOBOL**, where "x" is the drive letter of the drive where you wish OpenCOBOL to reside. After the folder has been

unzipped, make sure that the “**x:\OpenCOBOL**” folder contains subfolders named “**bin**”, “**copy**” and “**config**”. If, instead, you see only a single folder (most-likely this folder will be named “**OpenCOBOL.06FEB2009.mingw**”), you’ll want to open that folder and move its entire contents up to “**x:\OpenCOBOL**”.

2. Add definitions for the following environment variables to the Windows registry[8]:
    - COB_CONFIG_DIR – set to **x:\OpenCOBOL\config**
    - COB_COPY_DIR – set to **x:\OpenCOBOL\copy**
    - COB_LIBRARY_PATH – set to **x:\OpenCOBOL\bin**
    - COB_SCREEN_ESC – set to **Y**
    - COB_SCREEN_EXCEPTIONS – set to **Y**
    - PATH (the userid-local version) – **x:\OpenCOBOL\bin** is added to the existing value (if any)

Believe it or not, that’s it! You’re done.

The environment variables listed above are documented in the **OpenCOBOL Programmers Guide** (look them up in the index).

Creating a Run-Time Open COBOL Environment

If you will be distributing OpenCOBOL applications you develop to other computers, those systems will need a minimal subset of OpenCOBOL installed on them. You can easily create such a subset distribution using an external hard drive or a flash drive, as follows:

1. mkdir -p z:\OpenCOBOL\bin – where “z” is the drive letter of the external hard drive or flash drive
2. mkdir –p z:\OpenCOBOL\Documentation\Berkeley DB\license[9]
3. copy x:\OpenCOBOL\bin\*.dll z:\OpenCOBOL\bin
4. copy x:\OpenCOBOL\bin\OC*all.exe z:\OpenCOBOL\bin
5. copy x:\OpenCOBOL\*.bat z:\OpenCOBOL
6. copy x:\OpenCOBOL\Documentation\Berkeley DB\license\*.html z:\OpenCOBOL\Documentation\Berkeley DB\license[9]
7. copy x:\OpenCOBOL\*.pdf z:\OpenCOBOL

Of course, you might wish to include documentation to your applications in your distribution!

Installing an OpenCOBOL Run-Time Environment

To install the run-time distribution you created in the previous section, follow the same instructions as were presented in the “Installing OpenCOBOL on your PC” discussion. When you update the Windows registry to define the environment variables[8], only the following will need to be created and/or updated:

- COB_LIBRARY_PATH – set to **x:\OpenCOBOL\bin**
- COB_SCREEN_ESC – set to **Y**
- COB_SCREEN_EXCEPTIONS – set to **Y**

[8] Use the process described in “Setting the OpenCOBOL Environment Variables”

[9] This is to satisfy the license requirements of the Berkeley DB module

- **PATH** (the userid-local version) – **x:\OpenCOBOL\bin** is added to the existing value (if any)

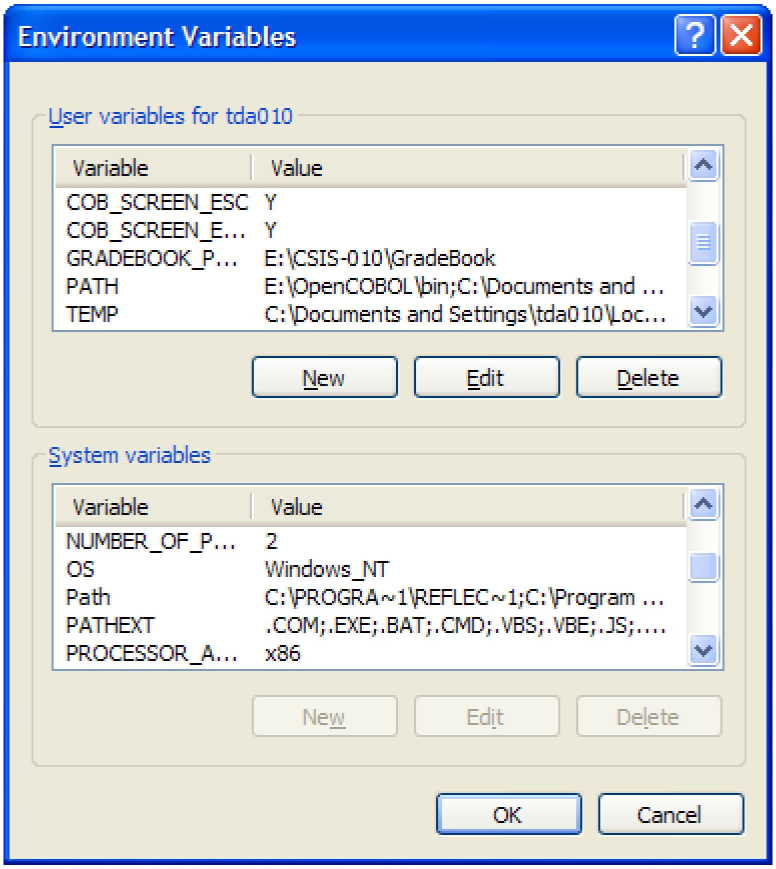
In lieu of permanently setting environment variables on run-time only systems, you could instead create Windows BAT files that execute your OpenCOBOL applications. These BAT files could then use Windows SET commands to establish the environment variables before executing the programs.

Setting the OpenCOBOL Environment Variables

If you REALLY know what you're doing, you can create the various environment variables OpenCOBOL needs using the Windows registry editor utility or a BAT file that utilizes the REG command to make the necessary changes (if you don't know what these are, you don't know enough to try it!).

A much safer approach is to use the "**Environment Variables**" configuration dialog.

This process is something you'll only have to do once per computer on which you're deploying OpenCOBOL.



Variable	Value
COB_SCREEN_ESC	Y
COB_SCREEN_E...	Y
GRADEBOOK_P...	E:\CSIS-010\GradeBook
PATH	E:\OpenCOBOL\bin;C:\Documents and ...
TEMP	C:\Documents and Settings\tda010\Loc...

Variable	Value
NUMBER_OF_P...	2
OS	Windows_NT
Path	C:\PROGRA~1\REFLEC~1;C:\Program ...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	x86

A sample of the "**Environment Variables**" configuration dialog is shown at the left. You'll need to create the various "COB_xxxx" environment variables in the "**User variables**" area shown at the top of the dialog.

This example shows how the Environment Variables dialog appears in Windows XP, but it's identical (except for basic windowing differences) on Windows Vista or Windows 7.

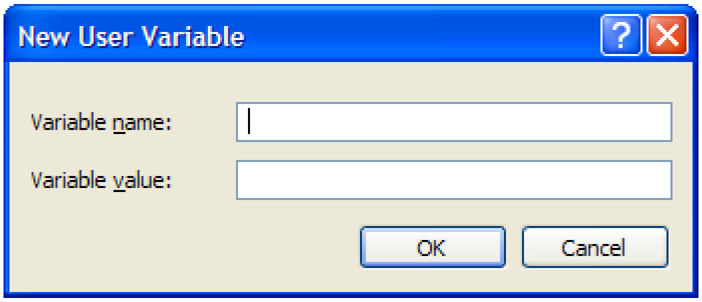
Note how the "Path" variable" appears in both the User and System areas[10]. Anything specified in both areas cause the actual environment variable value to be the union of both sets of values (User settings first, then System).

[10] Windows is generally case-insensitive. The "PATH" variable appearing in the "User variables" area is the same variable as the "Path" variable appearing in the "System variables" area.

The manner in which you access the “**Environment Variables**” dialog will differ somewhat, depending on the Windows version you’re using.

If You’re Using Windows XP	If You’re Using Windows Vista or Windows 7
1. Right click the “**My Computer**” icon and select “**Properties**”. 2. Click the “**Advanced**” tab 3. Press the “**Environment Variables**” button	1. Right-click the “**Computer**” icon and select “**Properties**”. 2. Click on “**Advanced System Settings**”. Note that this will produce a UAC dialog requiring you to temporarily elevate your session to Adminstrator status in order to continue. 3. Press the “**Environment Variables**” button.

Once you’ve brought up the Environment Variables dialog, use of the “**New**” button (the one in the “**User variables**” area!) will bring up the “**New User Variable**” dialog (shown to the right). Simply enter the variable name and variable and then press the “OK” button. Repeat as needed for each variable you need to create.



If you already have a PATH variable in the “**User variables**” area, simply select that row in the display and press the “**Edit**” button. Add the “**x:\OpenCOBOL\bin**” folder to those already present (use a semicolon to separate your entry from the others that were already there).

Once you have finished, simply use the “**OK**” buttons on the various dialogs to apply your changes. Your changes won’t actually take effect, however, until the next time you log on. *You don’t need to reboot the computer* – just logoff and log back in.

The process discussed above makes the OpenCOBOL environment variables available to just your userid on the computer. If there are others that will need to use OpenCOBOL on the same computer, the process will need to be repeated for each of them. Alternatively, you could make all your environment variable changes in the “**System variables**” area so that they’ll be available to ALL users of the computer. This will, however, require that account you use to make the changes from to have full administrator privileges.