zCOBOL

Overview
v1.5.00

Automated Software Tools Corporation.

zCOBOL Overview

zCOBOL is an open source portable mainframe COBOL compiler available as part of the z390 open source portable mainframe assembler for Windows or Linux starting with z390 v1.5.00. You can download z390 and zCOBOL in InstallShield format for Windows for file image format for Linux from www.z390.org. You will also need the J2SE Java runtime which you can download from Sun Developer Network. The first release of zCOBOL has been regression tested with J2SE 6 update 12.

Be sure to remove any old obsolete versions of J2SE such as 1.4 or 1.5 which may conflict with the current version. Once you have installed z390 with zCOBOL and J2SE runtime, then you can start the z390 GUI interface or command line interface and enter the following command to compile, link, and execute the COBOL hello world demo on Windows or Linux:

```
ZC390CLG zcobol\demo\HELLO
```

The zCOBOL compiler has been developed as a new flexible tool for testing and modernizing COBOL applications without requiring rewriting existing programs. With the recent addition of z390 structured conditional macro assembler extensions, the development of zCOBOL became feasible and has evolved rapidly. As the recent article in the z/System Journal titled, "Easy COBOL Modernization for SOA" by L. H. Couch and Charles F. Townsend, November 2008 indicates there is a growing demand for tools such as zCOBOL to help seamlessly bridge legacy and modern IT solutions.

The zCOBOL compiler translates COBOL source language programs into executable code using the following 3 major components:

- The java program zc390.class in z390.jar reads COBOL language source program with file extension CBL and generates a z390 HLASM compatible mainframe assembler source program with MLC extension. Each COBOL verb becomes a macro call opcode and all the words following up to the next verb or period become positional parameters for the macro call. Periods generate a PERIOD macro call to terminate all structures which may be missing the optional END-IF type words. All dashes in words are converted to underscores unless in quotes. The level numbers in data division statements are mapped to WS macro call with level as first positional operand.
- The macros in COBOL verb macro library z390\zcobol\*.mac parse the parameters for each verb, access global macro symbol table, and call code generation macros to generated executable code. For example the IF macro issues calls to GEN_COMP macro to generate executable source code to compare two fields, and issues call to GEN_BC to generate executable source code to branch on condition.
- There are currently 4 optional zCOBOL executable code generation macro libraries:
    - z390\zcobol\z390 - zCOBOL code generation macros for HLASM native z9/10 code
    - z390\zcobol\java - zCOBOL code generation macros for J2SE Java
    - z390\zcobol\vce - zCOBOL code generation macros for MS Visual Express C++
    - z390\zcobol\i586 - zCOBOL code generation macros for HLA and MASM native Intel code

  The z390\zcobol\z390 HLASM code generation library is the primary focus currently. However there is a COBOL demo program z390\zcobol\demo\HELLO.CBL which can be compiled and executed in all 4 different target language environments using the initial zCOBOL release.

Once the z390 HLASM code generation macros are complete and all the NIST COBOL 1985 standards tests have been completed successfully as a first milestone, then these macros can be copied to the other libraries and modified to replace HLASM source code model statements with the other target language statements.

If you are an assembler or COBOL developer who would like to contribute to the zCOBOL open source project, join the zCOBOL group and indicate your specific interests. All users are welcome and are encouraged to submit bug reports and requests for priority on future open source zCOBOL and z390 development.

zCOBOL Compiler Commands

ZC390	translate COBOL CBL source file to macro assembler MLC source file
ZC390C	compile CBL to HLASM BAL and assemble to relocatable object code
ZC390CL	compile CBL to HLASM BAL, assemble, and link to z390 load module
ZC390CLG	compile CBL to HLASM BAL, assemble, link, and execute z390 load module
ZCJAVCLG	compile CBL to J2SE java and execute class
ZCVCECLG	compile CBL to MS Visual C++, link, and execute exe
ZC586CLG	compile CBL to HLA/ASM, link, and execute exe
ZCRT390.BAT	run zCOBOL to HLASM demos
ZCRTJAV.BAT	run zCOBOL to Java demo
ZCRTVCE.BAT	run zCOBOL to C++ demo (requires MS Visual Express 2008 install)
ZCRT586.BAT	run zCOBOL to HLA/masm demo (requires HLA and MASM installs)
ZCRTTEST.BAT	run zCOBOL to HLASM regression tests
ZCRTSAVE.BAT	save regression test generated files after any changes have been verified

All zCOBOL commands start with ZC and are located in the zcobol\bat directory and the z390 root directory for ease of use. All commands require z390 v1.5.00+ and J2SE 6.0+.

zCOBOL Project

Have you been bored lately? If you know COBOL and assembler or Java, or C there is a job on the zCOBOL project waiting for you. The pay is poor ($0) but the self actualization rewards can be very satisfying. And there is always the possibility of future paying jobs helping companies use zCOBOL. Current jobs available include writing COBOL verb macros for currently unsupported verbs including SORT, MERGE. Optimizing the code generation macros to produce more efficient code and optional code based on zCOBOL options such as TRUNC, R64, etc. In addition major effort is still required to covert the HLASM code generation macros to generated java, C, or MASM. For COBOL programmers there is the constant need to extend the zCOBOL regression tests written in zCOBOL which verify that zCOBOL statements produce the expected results. And finally there is a need to develop documentation on the zCOBOL project as it evolves.

zCOBOL NIST ANSI 85 Test Suite Results

The following results were obtained using zcobol v1.5.00 and v1.5.00a. The improvement in v1.5.00a came from RPI 1001 for conditional 88 support, RPI 1002 SET and index support, and RPI 1012 miscellaneous syntax error corrections. Once some of the remaining critical support items such as COMPUTE are completed, these numbers should continue to significantly improve. The plan is to achieve 100% within the next few releases of zCOBOL and then provide optional regression test download for the NIST test suite for zCOBOL. These statistics were extracted from

Description	V1.5.00	v.15.00a	Notes
NIST programs with parsing errors	140	43	RPI 1012 corrections to zc390 parser
NIST Programs Compiled	319	416	RPI 1012 corrections to zc390 parser
Total minutes	26	44	33% increase in number of programs compiled
RC=0 No errors	11	12	most programs are still missing one or more items
RC=8 MNOTE support warning	19	151	Warning for unsupported items pending implementation
RC=16 At least 1 error message	249	160	Error messages from mz390 or az390 macro assembler

Trademarks

IBM, CICS, HLASM, MVS, OS/390, VSAM, z9, z10, and z/OS are registered trademarks of International Business Machines Corporation

Credits

Author : Don Higgins
Formatting : Walter Petras
Z390 version :