

z390 User Guide V1.5.01

z390 Portable Mainframe Assembler and Emulator User Guide for V1.5.01 Table of Contents

- 1 Introduction**
- 2 Installation**
- 3 Execution**
 - 3.1 Standalone graphical user interface**
 - 3.2 Command line interface**
 - 3.3 Batch command file interface**
 - 3.4 Sequential and Random file I/O Support**
 - 3.5 ASCII versus EBCDIC character set options**
 - 3.6 Debugging**
 - 3.7 SVC support options**
 - 3.8 Performance benchmarks**
 - 3.9 Utility programs**
 - 3.10 ASSIST extended instructions**
- 4 Reference Links**
- 5 Release Notes**
- 6 Appendix**
 - 6.1 Input demo.mlc macro assembler hello world demo source program file**
 - 6.2 Input demo.bat assemble, link, and execute command file**
 - 6.3 Output demo_2005_0812_091929.log z390 GUI time-stamped log file**
 - 6.4 Output demo.bal mz390 expanded basic assembler language file**
 - 6.5 Output demo.prn az390 assembler listing file**
 - 6.6 Output demo.obj az390 assembler object file**
 - 6.7 Output demo.lst lz390 linker listing**
 - 6.8 Output demo.390 lz390 load module file (superzap dump of binary file)**
 - 6.9 Output demo.log ez390 execution log file with WTO output and any errors**
 - 6.10 Output demo.tre ez390 execution trace file (specify option TRACE)**

z390 User Guide V1.5.01

1.0 Introduction

The z390 and zcobol portable macro assembler and COBOL compiler, linker, and emulator toolkit provides a way to develop, test, and deploy mainframe compatible assembler and COBOL programs using any computer that supports the Sun Microsystems J2SE 1.6.0+ runtime. The ASSIST instruction set is included for use by students at Northern Illinois University and other education institutions. The zcobol portable mainframe COBOL compiler is now included along with demos, regression test programs, and results from NIST COBOL 1985 test suite. The first z390 and zcobol session was presented at SHARE in Austin Texas, March 3, 2009. All the documentation for zcobol is on www.zcobol.org. Developers can now code and unit test assembler and COBOL programs using the latest z9/z10 mainframe instructions on Windows and Linux platforms. The zcobol compile supports static and dynamic linking with both zcobol and assembler programs and also supports EXEC CICS COBOL and assembler using z/CICS support developed and contributed by Melvyn Maltz. As of zcobol v1.5.01, 409 of the NIST COBOL test suite programs compile with no hard assembler errors, just level 8 MNOTE's for those features not yet supported by zcobol.

The z390 tool is distributed in both Java source and executable form under the open source GPL license. The www.z390.org website provides downloads in InstallShield exe format for Windows and zip file format for Linux and other systems plus online documentation. There is also a summary of source code statistics including lines by program provided on the website. Visit the www.z390.org website for more information on the installation and use of z390 on Windows and Linux including support for GUI and command line interface. The open source Eclipse IDE is recommended for those who wish to edit and debug the z390 Java source code. For Linux, there are perl scripts which provide support for execution of DOS BAT files and subset of DOS commands including those required to run all the z390 regression tests.

The Appendix contains source and generated file listings for the z390 demo “Hello World”. The source program demo\DEMO.MLC is a 4 line macro program which uses standard linkage macros SUBENTRY and SUBEXIT to enter and exit the program and uses WTO macro to issue the “Hello World” message on console and log file. To assemble, link, and execute the demo after installing z390 and J2SE, enter the command ASMLG demo\DEMO from the GUI command line or Windows command line with z390 install directory as current directory. All the generated files including expanded macro source DEMO.BAL, assembler listing DEMO.PRN, linker listing DEMO.LST, and execution log DEMO.LOG are also listed in the appendix.

The z390 Linux support includes detection of which operating system is running, taking appropriate defaults, and handling differences. Common code has been updated to avoid conflicts due to case sensitivity, file separator, and change directory differences. All z390 directory names are now lower case, and all test file

z390 User Guide V1.5.01

names are uppercase for consistency. Under Linux the default z390 GUI editor is gedit instead of notepad. The default web browser is Firefox and the default PDF reader is acroread. All of these are installed by default with Ubuntu Linux. The only additional required package which can be downloaded and installed using Firefox is the runtime JRE 6.0. Optionally the Eclipse 3.2 version can also be used for editing and debugging the z390 Java open source code. See the online web page on installation and use of Eclipse for debugging z390 java source code. There are detail instructions for the installation of Ubuntu Linux plus z390 and the above packages so you can have equivalent z390 development and test environments up and running on a dual boot Windows and Linux PC in less than a day.

Thanks to Melvyn Maltz, there is support for an open source CICS compatible transaction manager developed in z390 assembler supporting multiple CICS clients on a single CICS server over a TCP/IP network. The CICS support includes BMS basic mapping support for assembler language and COBOL. See BMS map MAP01.MLC and TESTBMS1.MLC sources in the cics directory. The transaction manager includes support for EXEC CICS compatible commands including SEND, RECEIVE, LINK, LOAD, XCTL, HANDLE, PUSH, POP, RETURN, and more. The latest version also supports browsing of ESDS, RRDS, and KSDS VSAM files. The z390 macro assembler automatically parses EXEC space delimited statements into comma delimited macro calls. The assembler also automatically converts DFHRESP(type) references into literal full word constant references to predefined response codes. See the cics directory for additional documentation, demos, and startup commands.

The above CICS compatible support uses z390 TCP/IP sockets I/O via the TCPIO macro and associated svc. There is also a z390 Service Oriented Architecture (SOA) client server application generator with COBOL and assembler demo in the soa\demo directory. COBOL SOA client code can be generated by the z390 SOA generator. The demo has been successfully run using Micro Focus COBOL on Window calling z390 services running on Windows. The new COBOL SOA support uses IBM standard EZASOKET interface from COBOL clients. The EZASOKET source code is available to Micro Focus customers from Micro Focus. The demo illustrates how previously statically linked service routines can be automatically regenerated as SOA services called from clients on a TCP/IP network.

The z390 runtime includes all the problem state instructions in the latest published Principles of Operations version 5 including 64 bit register instructions, and 128 bit HFP, BFP, and DFP floating point instructions. The runtime includes svc's which support the most common MVS compatible macro functions with corresponding macros in the mac directory including QSAM and VSAM macros.

This version of z390 has been regression tested on Windows Vista and XP using J2SE version 1.6.0_16 runtime. Editing and debugging of all z390 Java source code has been done using Eclipse 3.5. Note that the Windows Vista User Access Control (UAC) option has been turned off to allow use of the default install directory in

z390 User Guide V1.5.01

C:\Program Files to be updated without prompting. The install directory can be changed to allow leaving UAC on if desired.

The z390 tools can be accessed as a standalone graphical interactive application named z390 or the individual component tools can be executed via interactive command line or via batch application call command. The z390 main graphical user interface (GUI) interface provides easy to use menu commands to assemble, link, and execute macro assembler programs plus command files. In addition to the z390 primary GUI interface, each assembler program has option to open a separate individual Graphical User Access Method (GUAM) gui dialog interface supporting the following views (see separate z390 GUAM User Guide for more information):

- **MCS Console view for WTO and WTOR messages**
- **TN3270 view supporting both EXEC CICS and TPUT/TGET interfaces:**
 - **TGET/TPUT EDIT option text messages are displayed on scrolling 3270 screen with prompt before erasing screen.**
 - **TGET/TPUT FULLSCR and ASIS option support 3270 compatible data streams including the commands WCC, SF, SFE, SBA, SA, IC, and PT. The AID input codes for PF1-PF24, PA1-PA3, CLEAR, and ENTER are supported. For Windows the PF10 keystroke is consumed by the z390 keystroke handle to avoid erroneous pop-up. CTRL-PF6 is erase to end of field, and CTRL-F7 is erase all input fields. The font size used by the TN3270 screen mode is adjusted to maximum which will still allow the entire screen to appear in the current window size. The window can be maximized or dragged to readjust the size of the window and the corresponding font size.**
 - **See demo\DEMOGUI6.MLC for application program using TN3270 macro to assist in defining 3270 data stream codes. This program uses TPUT and TGET with full screen option to support updating of name, address, and numeric zip fields and supports PF1 help and PF3 exit. The program sounds alarm and updates status line with error messages. Run this demo with the command:
ASMLG demo\demogui6 guam trace**
 - **See cics\TESTGUI6.MLC for EXEC CICS compatible version of the same application as demo\demogui6.mlc to update multiple fields on screen. This CICS transaction can be run from local or remote CICS terminal session (cics\Z390KCPR.390) over TCP/IP network connected to CICS server (Z390CICS.390). For more information on the EXEC CICS compatible transaction management support see the documentation in the cics\doc directory and on www.z390.org. Use cics\Z390CICG.BAT command to start CICS server and 2 default local CICS client terminal sessions. Use cics\Z390KCPR.BAT command to start additional local or remote CICS client terminal sessions (requires * parm for local or IP server address for remote sessions). From any CICS client terminal session, enter the GUI6 transaction ID to run the TESTGUI6 application dialog via the CICS server.**

z390 User Guide V1.5.01

- **Graphic display view for graphics drawn by GUAM graphical macro commands. GUAM macros also provide support for customizing the GUAM CUI window including title, size, location, font, etc.**

•
All of the z390 graphical user interfaces attempt to conform to industry standards including:

- **Popup menus – file, edit, options, view, and help**
- **Scrollable view of entire output log showing results of each command**
- **Command entry field with keyboard scrolling of prior commands**
- **Command entry field right click pop-up help menu**
- **Optional status line to show batch I/O and status information**
- **Option to change window location, size, and font for accessibility**
- **Option to submit, monitor, and cancel batch commands from GUI**

The menus and basic commands all have mnemonic keystroke equivalents. All fields including the entire log view have edit cut, copy, and paste support to allow easy transfer of selected text to and from the clipboard and other Windows applications.

The entire z390 toolkit is packaged into single executable Java z390.jar which requires the free Java runtime 1.6.0+ from Sun Microsystems. J2SE 6.0 Update 12 is recommended as this is the primary version used for regression testing this release. The minimum platform requirements are Windows Vista, Windows XP, or Linux, and 512 MB memory. The default J2SE runtime user memory allocation for mz390 can be specified in MZ390.BAT. The system default is usually sufficient. See the z390 online options page for summary of all command line options plus environment variables and J2SE execution options. You can set the memory allocation parameter `-Xmx????????` up to physical limit available on machine for very large assembler applications. However, setting the allocation too high can cause system stall due to over commitment of memory. The largest assembler user case tested thus far on z390 expanded over 450 macros and issued 45,000 macro calls. This assembly used 88 MB memory and ran 24 seconds as shown by ERR file step statistics.

z390 User Guide V1.5.01

2.0 Installation

To download the source and executable files for the z390 toolkit, go to the following website and click on download link:

<http://www.z390.org>

After downloading the zip file z390.zip, unzip the file and execute the Install Shield z390.exe program to install the z390 toolkit on Windows. The default install directory for Windows is:

`“c:\program files\Automated Software Tools\z390\”`

The z390 install directory can be added to system path in order to be able to run any z390 tool from any Windows command line directory. In addition, the file types mlc, bal, obj, and 390 can be added to Windows file types to allow double clicking on any z390 file type to invoke z390.

The only pre-requisite to use z390 is the free version of Sun Microsystems Java 2 J2SE runtime 6.0+ which can be downloaded and installed from the following web site (go to website and click on “Download JRE”):

<http://java.sun.com/javase/downloads/index.jsp>

You do not need the Sun Microsystems Java SDK System Development Kit or the J2EE Enterprise Edition components, but if you have either of these already installed they include the required J2SE runtime. You may want to install the JDK which is 50+ MB if you want to compile and debug Java code and build your own JAR files. See the online documentation page on z390 Java source compiling and debugging. See the online page for more on compiling and debuggin java code: [http://z390.sourceforge.net/z390 Java Source Compile and Debug.htm](http://z390.sourceforge.net/z390%20Java%20Source%20Compile%20and%20Debug.htm)

Three optional z390 downloads are available:

1. **RT.ZIP** – regression tests can be downloaded and installed into the z390 install directory to create rt subdirectory with all the regression test source code used to debug and test z390 on Windows Vista, XP, and Ubuntu Linux.
2. **MVS.ZIP** – MVS 3.8 Public Domain macros can be downloaded and installed into the z390 install directory to create mvs subdirectory with all the public domain MVS 3.8 source macros which are useful examples for learning MVS macro code and the common macro interfaces. Note that although z390 can assemble MVS, z/OS, z/VM, and VSE macros, only the z390 macro library can be used to assemble, link, and then execute code under J2SE on Windows and Linux systems.
3. **WEBDOC.ZIP** – copy of the www.z390.org and www.zcobol.org online website documentation can be downloaded and installed in z390\webdoc. If the file z390\webdoc\index.html is found, then the help menu "Guide" will

z390 User Guide V1.5.01

open the local documentation index page otherwise it will open www.z390.org just like the help "Support" menu item always does.

There is an installation verification program which can be run by entering IVP.BAT from the GUI or Windows command line after setting current directory to the z390 install directory. This program will display the OS, J2SE, and z390 current versions and will issue warning messages if the required versions for successful z390 execution are not found. The z390 root directory plus the two optional rt and mvs directory installs all have new file named Z390.IVP which contains the current z390 version ID and is checked to verify installed version is compatible with the current z390.jar executable code. If the return code is not zero, please check the installation and rerun the IVP command again after resolving install problems. If the J2SE version is not at least 6.0+, z390 will not run successfully. Be sure to uninstall any older versions which may be causing conflict. If problems persist, first suggestion is to try removing all J2SE versions and all z390 versions including deleting the residual install directory files as some early versions left read only files which prevent correct install of current version. Then reboot and reinstall latest J2SE and z390 versions.

z390 User Guide V1.5.01

3.0 Execution

Once the installation is complete, the z390 toolkit for Windows can be executed in any one of the following modes:

- **3.1 Graphical User Interface (GUI) application mode**
 - Double click on the z390 toolkit icon on the desktop to startup in the standalone graphical user interface (z390.class within z390.jar).
 - From a command line, the graphical user interface can be started using the batch command file z390.bat located in the install directory. Use the command z390.
 - Once the GUI interface is started, you can issue commands by pointing and clicking the menu options or by typing commands on the command line entry field. The command line field support forward and backward scrolling of previous commands entered either way so you can repeat and or modify prior commands. The option menu allows you to click on desired default options, and the file menu commands enable you to select files via point and click dialog. The following commands are supported by the GUI interface:
 - **ABOUT** display summary information about z390 tool
 - **AMODE31 ON/OFF** set amode 24/31 for link cmd
 - **ASCII** toggle between ASCII and EBCDIC mode. The default is EBCDIC. This affects test commands to display memory and character values used in test commands.
 - **ASM MLC file** assemble MLC source to OBJ object code file
 - **ASML MLC file** assemble and link MLC source to 390 load module file
 - **ASMLG MLC file** assemble, link, and execute 390 load module file
 - **CD** change current directory
 - **CMD** command submit batch command or start/stop batch command mode
 - When in CMD mode, all commands are sent to batch processor
 - To cancel CMD processor, enter CTRL-BREAK once or toggle the CMD mode indicator on view menu on/off.
 - **Copy** copy selected text to clipboard (GUI right click)
 - **COMMANDS** alphabetical list of all commands
 - **CON ON/OFF** set console output for file cmds
 - **COPYLOG** copy the entire log text to clipboard (GUI only)
 - **Cut** cut selected text (GUI right click)
 - **Edit** any file edit source file in separate window

z390 User Guide V1.5.01

- **ERR nn** change error limit to nn from 100 default
- **EXIT** exit z390 after closing all files (also CTRL-BREAK)
- **EXEC 390 file** execute 390 load module
- **FONT size** change font size for accessibility
- **GUAM ON/OFF** set option for GUAM GUI interface dialog
- **GUIDE** view local optional documentation installed in z390\webdoc otherwise view online docs at www.z390.org
- **HELP** display help information summary
- **JOB bat file** execute batch job
- **LINK obj file** link obj file into 390 load module
- **LIST ON/OFF** set PRN, LST, and/or LOG output for file cmds
- **LISTCALL ON/OFF** set trace macro calls in BAL file
- **LOC x y pixels** change location of upper left corner of window
- **MAC mlc file** expand mlc macro file to bal source file with option NOASM to suppress assembly. This command can be used for text processing applications using AREAD and PUNCH such as LINKLIB\RTGENDIR, LINKLIB\RTGENCMP, and BMK\BMKTIME without the overhead of running the assembler.
- **NOTEPAD** start Notepad for use with clipboard data (GUI only)
- **PASTE** paste clipboard text (GUI right click)
- **PERM** display current Java security manager permissions
- **REL** display current release level for OS, Java, and z390
- **RMODE31 ON/OFF** set rmode 24/31 for link cmd
- **SIZE x y pixels** change width and height of window
- **STATS ON/OFF** set statistics for BAL, PRN, LST, and/or LOG file cmds
- **STATUS ON/OFF/sec** set status line on or off or set log interval sec
- **SUPPORT** open browser to online support web site
- **TEST ON/OFF** set prompt for interactive test cmds for EXEC cmd
- **TITLE 'text'** set GUI title
- **TIMEOUT seconds** set command timeout seconds or 0 for no limit (default)
- **TRACE ON/OFF** set trace for ez390 TRE file
- Any command not on the above list is assumed to be a batch command and is passed on to the batch command processor. Additional batch commands available in the default install directory include the following:

z390 User Guide V1.5.01

- **DEMO** – run demo.bat command to assemble, link and execute the demo.mlc program which displays “HELLO WORLD”.
- **MZ390** – run macro processor to expand MLC to BAL file (called by MAC, ASM, ASML, and ASMLG)
- **AZ390** – assemble to read BAL and generate OBJ file (called by ASM, ASML, and ASMLG)
- **LZ390** – link OBJ file and create 390 file
- **EZ390** – run z390 emulator (called by ASMLG and EXEC) supporting mainframe problem state instructions plus svcs for MVS compatible macro functions including QSAM and VSAM file I/O. Note linklib is default last SYS390 directory searched in order to support utilities without requiring path.
- **RT** – run the following batch regression tests from GUI or command line interface:
 - **RTDEMO** – run regression test on demo programs
 - **RTLINK** - run assembly and regression test on linklib utility programs and subroutines.
 - **RTMVS** – run MVS maclib test and demo programs in the optional download directory MVS (unzip z390_MVS.ZIP to z390 install directory from www.z390.org)
 - **RTTEST** – run regression tests on test programs in the optional download directory RT (unzip z390_RT.ZIP to z390 install directory from www.z390.org)
 - Visit website testing section for list of regression tests with brief descriptions
 - **RTVSAM** – run regression test and demos for ESDS, RRDS, and KSDS VSAM files including use of REPRO utility to load and unload VSAM files. See online z390 support page for more documentation and list of demos and regression tests including REPRO loads and unloads.
 - **RTVSE** – run regression tests for VSE support using macros which map VSE macro calls into MVS compatible z390 runtime macro or svc calls.
- **Run the following interactive regression tests from GUI or command line interface:**
 - **RTCICS** – start CICS server with 2 local clients and then start remote client (you must edit command to change the IP address of server processor. Use command IPCONFIG to find out IP address). Enter transaction id GUI6 to run demo CICS application with name and address fields plus PF1 help screen.
 - **RTSOA** – run demo of SOA application generator and then run client and server.

z390 User Guide V1.5.01

- **RTGUAM** – run interactive GUAM TN3270 demos and tests.
- You can also start the z390 GUI interface and specify a startup command by specifying parameter /sc followed by command. For example the command “z390 /sc maximize.ini” will startup the z390 gui interface and execute the commands in file maximize.ini such as:
 - **LOC 0 0**
 - **SIZE 800 600**
- The following additional parameters can be passed to z390 on any of the above startup commands:
 - **/NP** – suppress all permissions to test web version
 - **/NT** – suppress the unique time stamp included in the default log file name (By default the unique time stamp insures a log of all z390 commands executed are retained for each instance of z390 GUI window, command line, and batch file executions. The log files can be deleted periodically.)
 - **/RT** – regression test suppression of date and time stamps to simplify file comparisons
 - **/SC <file>** - startup bat command file to start Windows command after switching to interactive Windows command mode when running in GUI. This allows front ending any existing Windows interactive command line application with GUI front end supporting scrollable output window etc.
 - To change the GUI editor invoked by the file menu edit command, you can set environment variable named EDIT to path to any GUI editor you choose. The default is notepad.
- **3.2 Command line mode**
 - From the command line you can run any of the z390 toolkit components using the following batch command files (See online z390 Options page for documentation on all the options available):
 - **Z390.BAT** – start the GUI interface with optional startup command file input.
 - **MAC.BAT** – run mz390 macro processor to expand MLC macro source file to BAL assembler source file. See regression test TESTPCH2.MLC and RTTEST.BAT for example of how mz390 can be used to read and/or write any text file.
 - **ASM.BAT** – run mz390 and az390 assembler to expand MLC macro assembler source file and generate relocatable OBJ relocatable object file. Note the default object file format is binary 80 byte record mainframe compatible format. Use option OBJHEX for ASCII text format containing hex format for object code. The OBJHEX format supports single CSECT's over 16 MB and you can read the OBJ file for debugging purposes. ASM currently supports the following options:

z390 User Guide V1.5.01

- **ASML.BAT** – run **mz390**, **az390**, and **lz390** to expand MLC macro assembler source, assemble, and link to generate 390 load module.
- **ASMLG.BAT** – run **mz390**, **az390**, **lz390** and **ez390** to expand MLC macro source, assemble, link, and execute 390 load module.
- **LINK.BAT** – run **lz390** linker to read one or more relocatable OBJ files and create binary relocatable 390 load module file. If the linker option **AUTOLINK** is on, the linker will search **SYSLIB** OBJ file directory for external references to be statically linked. The linker includes options for **AMODE** and **RMODE** to control loading and execution modes. The linker also has optional input command file with suffix **LKD** which may contain explicit **INCLUDE**, **ENTRY**, **ALIAS**, and **NAME** commands.
- **EXEC.BAT** – run **ez390** emulator to execute 390 load module. The emulator supports 32 and 64 bit problem state IBM processor instructions including **HFP**, **BFP**, and **DFP** floating point instructions. The emulator supports the following interactive test commands when the **TEST** options is specified:
 - **addr=sdt** – set memory value (i.e. **1r?=x'80'** changes mem at (r1) 31 bit
 - **reg=sdt** - set register value (i.e. **15r=8** changes reg 15 to 8)
 - **A addr** – set or reset up to 100 instruction address stops with hex address or relative expression such as ***+4**
 - **B=addr** - set base for rel addr (ie **B=15r%** sets base to (r15) 24 bit
 - **D** – display DCB file information from **TIOT**
 - **E** – toggle between **EBCDIC** and **ASCII** mode
 - **F** – display floating point registers **F0-FF**
 - **G nn/addr/opcode** - exec nn instr. or until specified instruction address or opcode is found with no trace. One instruction is always executed before next opcode break even if it's the same instruction such as a **BCT 1,***. Addresses are distinguished from count by hex . or relative expression term such as *****, **+**, or **-**.
 - **H** - list help command summary
 - **J addr** - jump to new addr and trace instruction
 - **L** - list all regs and trace current instruction
 - **L reg** - list contents of register (ie **l 1r** dumps register 1
 - **L addr len** - list contents of memory area (ie **l 10. 4** dumps cvt addr
 - **M** – display total memory in MB and total allocate and free bytes

- P – display current loaded program information from CDE including name, entry and length
- Q - quit execution now
- R – display general purpose registers R0-RF
- S - clear register, address, and memory breaks
- S reg??sdt - set break on register change
- S addr??sdt - set break on memory change
- T nn/addr/opcode - trace n instr. or until specified instruction address or opcode is found. One instruction is always executed before next opcode break even if it's the same instruction such as a BCT 1,*. Addresses are distinguished from count by hex . or relative expression term such as *, +, or -. The symbol EPA may be used in place of address to refer to last program load point address.
- Z - exit test and run to end of program without trace
- * addr = hex.,+-hex, *+-hex, dec, nnr% (24 bit), nnr? (31 bit)
- * reg = nnr where nn = 0-15
- * sdt = self defining term (b'01',c'ab',f'1',h'2',x'ff')
- * ?? = break compare operator (=,!<,<=,>,>=)
- ZSTRMAC file1 file2 – translate ZSTRMAC structured conditional macro extensions into standard HLASM code. (See separate document on ZSTRMAC extensions).
- The emulator also supports basic OS svc services compatible at the macro interface such as:
 - ABEND - abort program with specified code
 - ACB - VSAM access control block
 - ACBD - VSAM ACB DSECT
 - BLDL - search for one or more 390 load modules in memory or in SYS390 load library directories.
 - CALL - call subroutine using standard linkage
 - CHECK - check completion of READ/WRITE
 - CLOSE - close DCB or ACB file
 - CPOOL - build and access quick cell memory
 - CMDPROC - control command processing tasks
 - CVTD - DSECT of limited CVT fields supported
 - CVTDCB field set to x'9B' indicating 31 bit OS
 - DCB - data control block for file (supports 31 bit DCBLRECLF field as extension to DCBLRECL. See SUPERZAP utility for example of how to read or write up to 2 GB file with 1 I/O operation.)
 - DCBD - DSECT of limited DCB fields
 - DCBE - define EODAD and SYNAD for DCB
 - DECBD - DSECT of limited DECB fields

z390 User Guide V1.5.01

- **DEFINE** - define VSAM Cluster Definition Table
- **DEQ** - release a resource lock set by ENQ
- **DELETE** - freemain 390 load module if use count 0
- **ENQ** - request lock on a resource
- **ERASE** - VSAM delete record
- **ESPIE** - set program interruption handler
- **ESTAE** - set program abend handler
- **FREEMAIN** - release memory area
- **GENCB** - generate VSAM ACB or RPL
- **GET** - read record from QSAM or VSAM file
- **GETENV** - get value of environment variable
- **GETMAIN** - allocate memory area
- **IHAEPIC** - DSECT for ESPIE exits
- **IHASDWA** - DSECT for SDWA ESTAE exits
- **LINK** - load and execute 390 load module
- **LOAD** - load 390 load module
- **MODCB** - modify ACB or RPL control block field
- **OPEN** - open DCB or ACB file
- **POINT** - position to relative record or RBA in file
- **POST** - post ECB as completed
- **PUT** - write record to QSAM or VSAM file
- **READ** - read block from file at current position
- **RETURN** - restore saved registers and return
- **RPL** - VSAM request parameter list
- **RPLD** - VSAM RPL DSECT
- **SAVE** - save specified registers
- **SETRP** - set register options for ESTAE exits
- **SHOWCB** - move ACB or RPL field to user area
- **SNAP** - dump selected area of memory and/or dump file information from TIOT/DCB's, general and floating point registers, program information from CDE, and memory allocation totals
- **STIMER** - wait specified interval of time
- **STORAGE** - obtain or release main storage
- **SYSSTATE** - set operating environment options
- **TESTCB** - test current value of ACB or RPL field
- **TGET** - read from GUI TN3270 interface
- **TIME** - get time and date in requested format
- **TTIMER** - test and/or cancel STIMER event
- **TPUT** - write to GUI TN3270 display
- **VCDTD** - VSAM Catalog Definition Table DSECT
- **WAIT** - wait for 1 or more ECB's to be posted
- **WRITE** - write block to file at current position
- **WTO** - write text message to operator console

- **WTOR** - write to operator with reply
- **XCTL** - transfer control and delete prior pgm
- **XLATE** - translate area to/from EBCDIC/ASCII
- **The following extensions are also supported:**
 - **Structured programming macros compatible with the IBM Toolkit:**
 - **ASMMREL** – macro to replace BC with BRC branches (beware that B *+offset code must be changed if this extension is used)
 - **ASMMSP.CPY** – copybook to include macros
 - **IF** – start alternate block selection
 - **ELSEIF** – start alternate block selection
 - **ELSE** – start alternate block selection
 - **ENDIF** – define end of selection blocks
 - **DO** – define WHILE= or UNTIL= iteration block of code
 - **ENDDO** – end iteration block
 - See DEMO\DEMOSTR1.MLC example
 - **Structured programming macros to support local CSECT performed procedure calls without using any registers plus option for use of re-entrant code stack:**
 - **PERFORM** or **PM** – call local procedure
 - **PENTRY** - define start of procedure
 - **PEXIT** return from procedure
 - See linklib\SUPERZAP.MLC for example usage.
 - **A'symbol** – extended expression operator to return 1 if symbol has been defined in lookahead but may not yet be defined as ordinary symbol. Note it may never be defined as ordinary symbol due to conditional branching between alternate source code paths.
 - **AREAD** and **P** extended parms to define up to 10 files open concurrently using **DDNAME=**, **DSNAME=**, **DSN=**, and **ID=**. Opening file for **PUNCH** output will close file if open for **AREAD** input. If file does not exist or file is at end of file, **AREAD** returns 0 length string. Blank lines return 1 space. **PUNCH** has option **FORMAT** to request **MLC/BAL** continuation formatting. The default is variable length text of any length. See RT\TEST\TESTPCH3.MLC for regression test example. Also see **LINKLIB\ZSTRMAC.ZSM**, **LINKLIB\RTGENDIR.MLC**, **LINKLIB\RTGENCMP.MLC**, **LINKLIB\RTGENDIF.MLC**, and **BMK\BMKTIME.MLC** for example of new **ZSTRMAC** structured macro code text processing utilities. **AREAD** and **PUNCH** extended file text records can be of any

length. When reading strings longer than 71 characters from inline, standard continuation format must be used.

- **CFD** – convert from 45 character display format which may include scientific notation to 128 bit integer or any HFP, BFP, or DFP floating point format. See macro for option to use z390 svc or mainframe compatible routines contributed by David Bond with www.tachyonsoft.com.
- **CTD** – convert from 128 bit integer or any HFP, BFP, or DFP floating point format to 45 character display format with scientific notation if required. See macro for option to use z390 svc or mainframe compatible routines contributed by David Bond.
- **CMDPROC** – support command processor with following command options. See TESTCMD1.MLC for demo execution of batch file.
 - **CMDPROC START** – start Windows command processor
 - **CMDPROC STOP** – stop Windows command processor
 - **CMDPROC WRITE,cmd** – send command to processor where cmd is RX label, (reg), or literal of EBCDIC field in double quotes or null terminated
 - **CMDPROC READ,record,length** – read output line from command processor where record is RX label or (reg), and length is absolute value or omitted in which case it uses L'record for length.
- **DCB** – supports following extensions:
 - **DSNAME** keyword defines file specification override for DDNAME. The DSNAME rx type field may include drive and path and must be defined as C'...' delimited by null byte or double quotes.
 - **RECORD** keyword defines default record area for use with GET, PUT, READ, WRITE if not specified.
- **EXEC** – supports EXEC CICS, SQL, and DLI commands that have been reformatted into standard EXEC macro call during mz390 loading. See SQL_INCLUDE for example of EXEC SQL support and see the CICS_SEND, CICS_RECEIVE, and other EXEC CICS command macros in the CICS directory which can be called by the EXEC macro.
- **GUI** – supports commands to control GUI window including:

- WINDOW commands to set title, location, size, and font
- GRAPH commands to draw graphics
- KEYBOARD commands
- SOUND commands
- LINK, LOAD, XCTL, and DELETE support
DSNAME= and DDNAME= extensions for multiple directory lists or specific 390 load module file or user data file of different type.
- SUBENTRY – generate standard entry for CSECT using default of register 13 as base pointing to save area 8 bytes into CSECT. This macro also supports multiple base register and RENT options.
- SOAGEN – generate Service Oriented Architecture (SOA) client server application from standard application. See SOA directory with macro library and demo. See SOA User Guide for more information.
- SUBEXIT RC=0 – generate standard exit linkage from CSECT setting R15 to 0 or value specified for RC=.
- &SYSCICS, &SYSCICS_EPILOG, &SYSCICS_PROLOG GBLB global variables added to allow macros to be CICS aware.
- TCPIO – TCP/IP sockets I/O for up to 10 client ports, 10 server ports, and up to 20 server connections
 - OPEN,PORT=[,HOST=] open client or host port
 - CLOSE,PORT= close port
 - SEND,PORT=,MSG=,LMSG=[,CONN=] send message
 - RECEIVE[,NOWAIT],PORT=,MSG=,LMSG=[,COND=] receive message from client port, any server port connection or a specific server connection returned from prior receive command.
- TIME – supports extended SVC linkage operands:
 - CLOCK,field,CLOCKTYPE=STCK – returns double word with bit 51 equal to microseconds since IBM epoch January 1, 1900.
 - CLOCK,field,CLOCKTYPE=STCKE – returns 2 double words with bit 59 equal to microseconds since IBM epoch January 1, 1900
 - CLOCK,field,CLOCKTYPE=JAVA – returns double word with bit 63 equal to milliseconds since JAVA epoch January 1, 1970.
 - SVC NS option for nanosecond interval time
 - SVC TS for JDBC time-stamp format including nanosecond precision

- **TN3270** – supports definition of TN3270 data streams using predefined control code names and row and column conversions for set buffer addresses. See DEMOGUI4.
- **VSAM** – support for VSAM ESDS, RRDS, and KSDS, type data sets are supported using DEFINE macro to assemble and link VCDT VSAM Cluster Definition Tables defining the type and location of VSAM base cluster data and index files. REPRO utility supports loading and unloading vsam files. Extensions include fixed and variable length records up to 2 GB with no fixed limit on number of alternate indexes. Visit VSAM support web page on www.z390.org for links to VSAM User Guide, reference links, and z390 examples. VSAM support in v1.3.09 includes browse support for ESDS, RRDS, and KSDS files and update support for ESDS and RRDS files. KSDS update with inserts is still in development. See directory vsam\demo for ESDS demo programs. Also see RTVSAM.BAT for regression tests. See z390 CICS support for browsing ESDS, RRDS< and KSDS files. Other VSAM file options will follow in future z390 releases.
- **ZCVTD** – define DSECT for z390 CVT located at x'2000' just beyond 8k PSA. The ZCVT contains fields unique to z390 which have been separated from the MVS compatible CVT which has pointer at X'10' in PSA.
- **ZSTRMAC** structured conditional macro coding extensions have been integrated into the mz390 macro processor during loading so no translation is required. See separate document on ZSTRMAC extensions, examples, and translator available for portability.
- **3.3 Batch execution mode**
 - The z390 GUI interface can be driven in batch mode for regression testing by specifying the parameters /SC file to start execution of the GUI interface reading commands from the specified file.
 - The macro processor, assembler, linker, and emulator can be driven in batch mode using the command described above (MAC, ASM, and LINK, and EXEC). For regression test examples see RT.BAT, RTDEMO.BAT and RTTEST.BAT which run programs and then verify the output files with expected output files previously verified and saved with different suffix (SV?).
 - The DEMO directory contains the following demo programs that are also included in regression testing:

z390 User Guide V1.5.01

- **DEMO.MLC** – 4 line macro program to display “HELLO WORLD” using SUBENTRY, WTO, SUBEXIT, and END statements.
- **DEMOWTO2.MLC** – use SETA variable to issue same WTO multiple times using open macro code loop and variable text.
- **DEMONUM1.MLC** – calculate and display primes using just conditional macro code instructions.
- **DEMONUM2.MLC** – calculate and display primes using 390 machine instructions.
- **DEMOM8Q1.MLC** – generate solutions to 8 queens on a chess board puzzle using just recursive conditional macro code with local and global variables.
- **DEMOBMK1.MLC** – execute the BCT instruction 2 million times to get performance statistics. On a 3 GHZ Dell Pentium running Windows XP, this program executes over 1 million BCT instructions per second.
- **DEMOBMK2.MLC** - execute an AXR and BCT 100,000 times to get performance statistics on 128 bit floating point. On a 3 GHZ Dell Pentium running Windows XP, this program executes 125,000 AXR instructions per second.
- **DEMOGUI1** – simple WTO and WTOR loop to demo GUI MCS console interface.
- **DEMOGUI2** – WTO and WTOR loop with instruction rate calculation demo.
- **DEMOGUI3** – TPUT and TGET edit mode TN3270 display demo.
- **DEMOGUI4** – TPUT and TGET native TN3270 data stream demo.
- **DEMOGUI5** – graphic display demo (runs to show all 3 display views but does little else in initial v1.0.08 version).
- The **TEST** directory contains a constantly growing number of regression tests including the following:
 - **TESTINS1.MLC** – verify assembler generation of over 700 IBM mainframe machine instructions.
 - **TESTINS2.MLC** – verify execution of problem state IBM machine instructions.
 - **TESTINS3.MLC** – verify execution of z10 extended instructions such as compare and branch.
 - **TESTFP1.MLC** – verify execution of HFP and BFP short, long, and extended floating point instructions.
 - **TESTTIM1.MLC** – verify time and date format options of TIME macro.
 - **TESTLNK1.MLC** – verify static and dynamic linking
 - **TESTMEM1.MLC** – verify getmain and freemain
 - **TESTTST1.MLC** – verify TEST interactive debug commands

- **TESTDCB1.MLC** – verify sequential GET/PUT of ascii text records using fixed length EBCDIC record DCB format FT
 - **TESTDCB2.MLC** – verify sequential GET/PUT of ascii text input file to EBCDIC FB output file.
 - **TESTDCB3.MLC** – verify sequential GET/PUT of EBCDIC FB input to FB output file.
 - **TESTDCB4.MLC** – verify sequential GET/PUT of ascii text records using variable length EBCDIC record DCB format VT
 - **TESTDCB5.MLC** – verify sequential GET/PUT of ascii text record input to EBCDIC V type output file.
 - **TESTDCB6.MLC** – verify sequential GET/PUT of EBCDIC V type input file to VB output file.
 - **TESTDCB7.MLC** – verify sequential GET/PUT of EBCDIC VB input to ascii text output file using VT format.
 - **TESTDCB8.MLC** – verify sequential READ/WRITE of EBCDIC F type EBCDIC files.
 - **TESTDCB9.MLC** – verify random access POINT/READ/WRITE of F type EBCDIC files.
 - **TESTDCBA.MLC** – verify use of DCBREC, DCBDSNAM override, and creation of new output file.
 - **TESTDCBB.MLC** – verify use of DCBREC with READ/WRITE.
 - **TESTDCBC.MLC** – test GET, PUT with ASCII mode including test of X'1A' end of file character processing.
 - **TESTERR4.MLC** – test S013 abend after error on bad DSNAM field.
 - **TESTERR5.MLC** – test SYNAD exit after error on bad DSNAM field.
- **3.4 Sequential and random file I/O support**
 - See test programs **TESTDCB1** – **TESTDCB9** for examples you can run as part of regression tests (**RT.BAT**). These programs use the z390 runtime macros **OPEN**, **CLOSE**, **GET**, **PUT**, **READ**, **WRITE**, **POINT**, and **CHECK** to perform sequential and random I/O. A data control block defined by **DCB** macro and mapped by **DCBD** macro provide common interface for defining options including which file to access. Each **DCB** defines **DDNAME** which is 8 character field which refers to environment variable name for Windows which in turn defines the path to the file to be accessed. The test programs are run using batch command **RT5.BAT** which defines 3 test **DDNAME** files **SYSUT1**, **SYSUT2**, and **SYSOUT** pointing to files with suffix **TF1**, **TF2**, and **TF3** respectively. These same **DDNAME** references are used in all the examples.
- **3.5 ASCII versus EBCDIC characters options**
 - The EBCDIC character set is the default mode for compatibility with mainframe programs many of which contain hex values that assume EBCDIC.

z390 User Guide V1.5.01

- The ASCII option can be specified to switch from EBCDIC to ASCII resulting in C'...' type strings generating EBCDIC values and macro string comparisons to be done in ASCII versus EBCDIC. Note that in ASCII digits (X'30'-X'39') compare less than characters where as in EBCDIC characters compare low to digits (X'F0'-X'F9').
- Extensions have been added to support generating C"... " strings as ASCII and C!..! in EBCDIC in both modes.
- Examples of ASCII and EBCDIC mode aware test programs include TESTDC1, TESTSDT1, TESTINS2, and TESTASC3 (copy or TESTINS2). Conditional macro code can be made character set aware by test such as:
AIF (C'A' EQ X'41').ASCII
- Execution time changes for ASCII mode elimination of EBCDIC dependencies include the following:
 - ED and EDMK require EBCDIC mask but generate ASCII output field
 - UNPK inserts X'3' as ASCII zone
 - All packed decimal instructions allow X'3' as alternate positive sign
 - See the TESTINS2.MLC and TESTASC3.MLC copy of TESTINS2 run in ASCII mode for example bi-modal code to handle this packed decimal character set dependency. Note TESTINS2 also handles different MVCL and CLCL fill character dependencies.
 - LINK, LOAD, and DELETE EPLOC fields plus DCB DDNAME field are all expected to be ASCII when running in ASCII mode.
 - When reading or writing records using DCB RECFM=FT or VT there is no translation performed as the records are assumed to already be ASCII versus EBCDIC.
 - Character type self defining terms such as C'...' used in test debug commands are generated as ASCII when in ASCII mode. Note C"... " and C!..!" can be used for mixed mode testing.
- 3.6 Debugging Tips
 - Review the ERR file for summary of steps executed and all errors issued prior to termination.
 - Turn on TRACEALL option and rerun failing process to get more information on where process is failing. TRACE information appears on TRM, TRA, TRL, and TRE trace files for mz390, az390, lz390, and ez390 respectively.
 - To interactively debug ez390 execution of assembler program, turn on the TEST option that will prompt for commands at the start of execution. You can set register, memory, or opcode breaks to execute up to point of interest in the program. Use T or Z commands to trace or zoom for n instructions or to next break point. At any break, you

- can display registers or memory and jump to different location if desired before proceeding to next break or end of program.
 - If the error appears to be in one of the z390 Java components, the fastest way to isolate the problem is to use a Java source code debugger such as Eclipse 3.2 which you can download and install free from www.eclipse.org. Once you have the source Java debugger up and running with the failing component, you can run with option notrap and notiming to allow process to run to point of failure. If the process is issuing an error message, you can search for the error message statement using “(nn” for search key and set break on the error statement to analyze the value of variables at the point of failure.
 - For information on compiling and debugging z390 java source code see the online web page.
 - If you make changes to the Java code, be sure to backup source code before each change and run all the regression tests after each change.
 - If you have found a problem and want a fix or enhancement to be provided in the next release of z390, please submit a problem report by visiting the support section on the www.z390.org web site.
 - If you are interested in participating in the development and support of z390, please join the z390 open source discussion group and get involved.
- 3.7 SVC Options
 - The current default svc support is for OS/390 and z/OS compatible macros as defined above under ez390 runtime services. There is also an optional set of VSE macros in the vse\mac directory along with VSE demo and test programs. The VSE macros map into z390 MVS compatible generated code and svc calls so the MVS macro library should be concatenated behind the VSE macro library for VSE programs macro assemblies as shown in RTVSE.BAT command for VSE regression tests. See new z390 VSE User Guide.
 - Additional svc's can be coding them in Java as part of ez390 or by writing an svc handler in assembler using the following options:
 - If the first byte of the new SVC PSW at low memory address x'60' is zero then all svc's will be executed using the native Java code in ez390. If the first byte is set to x'FF', then any svc will result in storing interrupt code and psw address of next instruction at old SVC PSW at low memory address x'20', and transferring control to the address set in new SVC PSW.
 - To enable a user written svc handler, set new SVC PSW address first and then move x'ff' to first byte of new SVC PSW.
 - While in the svc handler, move x'00' to first byte of new SVC PSW in order to issue native svcs, and then restore new SVC PSW prior to exiting the handler.

z390 User Guide V1.5.01

- See regression test TESTSVC1.MLC for example program which maps svc 201 to svc 35 WTO. Note it also handles mapping svc 3 to native svc 3 so user does not have to turn off handler prior to exiting main program via svc 3 at return address.
- **3.8 Performance Benchmark programs in BMK directory:**
 - The instruction timing benchmark program BMK\INSTTIME.MLC has been contributed by Melvyn Maltz. Run BMK\INSTTIME.BAT command file to generate the statistics report BMK\INSTTIME.OUT using the control file BMK\INSTTIME.TXT defining the instructions to be timed. The statistics report is also published on the website www.z390.org.
 - The program BMK\BMKTIME.MLC can be used to measure the elapsed time in milliseconds of any process run between the first and second call to BMKTIME. Run BMK\BMKTIME.BAT to see example timing of the DEMO\DEMOM8Q1 8 queen's recursive demo showing 40k versus 35k macro instructions per second on previous release. Note the speed the mz390 macro processor has increase by factor of 3 with the addition of macro pseudo code in v1.2. Large macro assembly processes now run 3 times faster. See separate PDF document on www.z390.org website describing the macro pseudo code option in more detail.

○

3.9 Utility Programs

- **z390** includes the following utilities in the linklib directory:
 - **SUPERZAP** – z390 utility program to verify, patch, and/or dump the contents of any file up to 2 GB. This utility also demonstrates how to use structured programming macros. SUPERZAP commands include:
 - **NAME** file – load and Windows or Linux file up to 2 GB in memory
 - **VER** offset value – verify file content at offset in hex or ‘text’
 - **REP** offset value – replace file content at offset in hex or ‘text’
 - **DUMP** offset length – dump file contents in hex and text
 - **ASCII** – switch text mode to ASCII
 - **EBCDIC** – switch text mode to EBCDIC (default)
 - **BASE** offset – set base for offsets (offset can be hex or *+hex or *-hex where * is current base set by FIND or NAME)
 - **FIND** value – search for hex or ‘text’ value from current base and set new base else set verify error
 - **LOOP** – repeat commands from last FIND until verify error or end of file is reached.
 - See linklib\ZAPDEMO1.BAT for example using command file.
 - Execute SUPERZAP.BAT with default command line interface and enter help for list of commands.
 - **S4BC** –generator contributed under GNU GPL V3 by Roger Williams with www.accessaba.com. See linklib\barcode for documentation and source used to build linklib\S4BC.390 utility.
 - **UNREF** – z390 utility contributed by Melvyn Maltz to scan PRN listing and identify all unreferenced symbols. See documentation and demo in util\unref. Run UNREF.BAT with PRN file name to generate UNF file with list of unreferenced symbols.
 - **REPRO** – VSAM load and unload utility
 - **INFILE** – ddname set to input QSAM file for load with optional suffix of [RECFM=FT/VT] indicating input file is ASCII text vs standard fixed or variable length QSAM file. Alternatively INFILE can be set to path and name of VCDT loadable VSAM catalog followed by . and base cluster name for unload.
 - **OUTFILE** – ddname set to output path and name of VCDT loadable VSAM catalog followed by . and base cluster name for load followed by optional [DISP=MOD] if records are to be added versus replaced which is the default. Alternatively OUTFILE can be set to path and name of QSAM file for unload with optional [RECFM=FT/VT] to indicate output QSAM file is to be ascii text versus the default fixed or variable length QSAM file.

z390 User Guide V1.5.01

- See `vsam\demo\RPO1.BAT` for demo load of KSDS file from text file with 6 name, address, city, state, zip records.
- You can execute `REPRO.BAT` with `INFILE` and `OUTFILE` specs as parms.
- **ZPAR** – z390 program analysis report utilities all written in structured conditional macro assembler are in the `zpar` directory and are documented on www.zpar.org:
 - **ZPARGEN** – generate batch execution commands for one or more programs
 - **ZPARERR** – summarize the ERR logs for one or more programs.
 - **ZPARMLC** – frequency distribution of opcodes for one or more MLC or BAL source files.
 - **ZPARSUM1** – summarize ERR log braking on first 2 characters of program name.
 - **ZPARSUM2** – summarize translation, assembly, and execution of NIST test programs by ID.
 - **ZPARTRS** – trace execution showing source assembler lines with data labels etc.
 - **FIXCRLF** – utility to add CR, LF to source (some editors don't require CR)

z390 User Guide V1.5.01

3.10 ASSIST Extended Instructions

The following instructions have been added to z390 to simplify input and output for use in student mainframe assembler classes:

- **z390 ASSIST instructions**
 - **XDECI RX X'53rxbdd' r1,s2 convert decimal to binary**
 - Start scan for next decimal number at s2 skipping leading blanks
 - Convert decimal number to binary in r1 until non decimal character found
 - Set register 1 to address of the last non decimal character found
 - Set condition code 0 if number converted successfully else set condition code 3
 - **XDECO RX X'52rxbdd' r1,s2 convert binary to decimal**
 - Convert binary 32 bit r1 value to right justified 12 character decimal field at s2
 - **XDUMP RXSS X'E06xbdddbdd' s1(x1),s2 - dump registers and/or storage**
 - If no operands are specified, dump registers and the default storage area
 - If no XLIMD instruction has reset default storage dump area, dump all storage
 - if s1(x1) address of storage area and s2 length of area are specified just dump that area
 - **XGET RXSS X'E0Axbdddbdd' s1(x1),s2 read record from ASCII file DDNAME=XGET into area s1(x1) with length of s2**
 - **XHEXI RX X'61' r1,s2 convert hex to binary (cc3 if no hex, update field addr)**
 - **XHEXO RX X'62' r1,s2 convert binary to hex (12 bytes, update field addr)**
 - **XLIMD RXSS X'E08xbdddbdd' s1(x1),s2 set default XDUMP storage area address and length**
 - **XPNCH RXSS X'E04xbdddbdd' s1(x1),s2 write to DDNAME=XPNCH with length s2**
 - **XPRNT RXSS X'E02xbdddbdd' s1(x1),s2 write to DDNAME=XPRNT with length s2**
 - **XPUT RXSS X'E0Cxbdddbdd' s1(x1),s2 write to DDNAME=XPUT for length s2**
 - **XREAD RXSS X'E00xbdddbdd' s1(x1),s2 read record from DDNAME=XREAD for length s2**
- **z390 ASSIST command can be used to assemble, link, and execute assist programs**
 - Use the command ASSIST filename where filename is the base name of any mainframe assembler source program with file type MLC to

z390 User Guide V1.5.01

assemble, link, and execute the program on Windows XP/Vista or Linux system using z390 GUI or command line interface. This command sets the ASSIST option on and defines the following environment variables for input and output ASCII text files:

- **XREAD=filename.XRD**
- **XPRNT=filename.XPR**
- **XPNCH=filename.XPH**
- **XGET=filename.XGT**
- **XPUT=filename.XPT**
- **All of the above files are optional and will be opened on first access and closed at end of execution or end of file. All files are ASCII with automatic conversion from or to EBCDIC. ASCII file records have trailing spaces removed and end with carriage return x'0d' and line feed x'0a'. On input fields are padded with EBCDIC spaces x'40'. Non ASCII output bytes are converted to periods.**
 - **If a file operation is successful the condition code is set to 0 for XREAD, XGET, and XPUT.**
 - **XPRNT and XPNCH do not set condition code.**
 - **At end of file for XREAD or XGET, the condition code 1 is set.**
 - **If a file error occurs on XREAD, XPRNT, XPNCH, the program aborts with abend code S013.**
 - **If a file open error occurs, program terminates with S013 abend with error message showing the file specification which failed.**
 - **If any other error occurs on XGET or XPUT such as missing length, condition code 2 is set.**
- **The following ASSIST debugging aids have been added to z390:**
 - **All registers are initialized to x'F4', memory is initialized to x'F5', and load module uninitialized areas are set to x'F6' to aid in debugging initialization problems. Use option NOINIT to revert to binary zero default.**
 - **List last 10 instructions at ABEND.**
 - **Display BC format PSW with ILCC and AMODE bits**
 - **Load programs in low memory versus default high memory.**
- **Example commands:**
 - **ASSIST assist\DEMOAST1 - assemble, link, and execute new ASSIST demo program which uses the ASSIST XDECO and XPRNT instructions to print prime numbers to 100. See the following files:**
 - **assist\DEMOAST1.MLC - source program**
 - **assist\DEMOAST1.PRN - generated assembly listing**
 - **assist\DEMOAST1.LOG - generated output log including all XPRNT output**
 - **assist\DEMOAST1.XPR - generated output file with all XPRNT lines.**

z390 User Guide V1.5.01

- **ASSIST assist\TESTAST1 - assemble, link, and execute new ASSIST regression test named TESTAST1.MLC in the assist sub-directory. After the command completes, the following files are available for viewing:**
 - assist\TESTAST1.MLC - source assembler program with ASSIST instructions
 - assist\TESTAST1.ERR - error log with start/stop messages plus any error messages
 - **assist\TESTAST1.PRN - assembly listing**
 - **assist\TESTAST1.LST - linker listing**
 - **assist\TESTAST1.LOG - execution log with XPRNT and XDUMP output**
 - assist\TESTAST1.XRD - XREAD test input file
 - assist\TESTAST1.XPR - XPRNT output file
 - **assist\TESTAST1.XPH - XPNCH output file**
 - **assist\TESTAST1.XGT - XGET test input file**
 - **assist\TEST\AST1.XPT - XPUT output file**
- **ASSIST assist\TESTAST1 TRACE - add the TRACE option to generate TESTAST1.TRE output file with XPRNT, XDUMP, plus instruction trace for use in debugging.**
- **ASSIST assist\TESTAST1 TEST - add the TEST option to start interactive test execution of the program with prompts for test commands which support tracing one or more instructions, dumping and/or changing registers and memory, etc. Type H for help list of test commands.**

z390 User Guide V1.5.01

4.0 Reference Links:

- 1 [Eclipse Universal Java Tool Platform](#)
- 2 [IBM Principles of Operations](#)
- 3 [IBM High Level Assembler \(HLASM\)](#)
- 4 [Microsoft Windows](#)
- 5 [Sun Micro Systems Java 2 J2SE Runtime](#)
- 6 [z390 Portable Mainframe Assembler Toolkit](#)
- 6.1 [z390 Documentation Links](#)
- 6.2 [z390 User Guide](#)
- 6.3 [z390 Options and File Types](#)
- 7 [z390 Portable Mainframe COBOL](#)

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

z390 User Guide V1.5.01

Release Notes:

Visit the www.z390.org website download section for the latest downloads of z390 for Windows and Linux plus detail information on release fixes and enhancements:

1. z390 Download Links:

<http://www.automatedsoftwaretools.com/z390/#Download%20Links>

2. z390 Download Archive with latest release fixes and enhancements:

http://www.automatedsoftwaretools.com/z390/z390_Download_Archive.htm

3. z390 Download History with prior release fixes and enhancements:

http://www.automatedsoftwaretools.com/z390/z390_Download_History.htm

4. Pending RPI fixes and enhancements log:

http://www.automatedsoftwaretools.com/z390/z390_Support_Request_Log.htm

z390 User Guide V1.5.01

Appendix: Input and output files from Hello World Demo program:

6.1 Input demo.mlc macro assembler source program file

```
TITLE 'Z390 HELLO WORLD DEMO'
* AUTHOR.  DON HIGGINS.
* DATE.    08/12/05.
* REMARKS. YOU CAN ASSEMBLE, LINK, AND EXECUTE THIS DEMO
*          USING Z390 GUI INTERFACE BY ENTERING DEMO IN COMMAND BOX
*          OR BY EXECUTING DEMO FROM WINDOWS COMMAND LINE
*          WITH CURRENT DIRECTORY SET TO Z390 INSTALL DIRECTORY WHICH
*          IS "C:\PROGRAM FILES\AUTOMATED SOFTWARE TOOLS\Z390"
*          THE Z390 TOOLKIT IS DISTRIBUTED IN SOURCE AND EXECUTABLE
*          FORMAT UNDER OPEN SOURCE GPL LICENSE.  VISIT WWW.Z390.ORG
*          FOR MORE INFORMATION.
DEMO      SUBENTRY
          WTO    'HELLO WORLD'
          SUBEXIT
          END
```

6.2 Input demo.bat command file

```
rem demo hello world demo
asmclg demo\demo.bat trace
```

z390 User Guide V1.5.01

6.3 Output demo_2005_0812_091929.log z390 GUI time-stamped log file

```
Z390I V1.2.00
Copyright 2006 Automated Software Tools Corporation
z390 is licensed under GNU General Public License
Log file = D:\WORK\Z390\z390.LOG_2006_1112_145100.LOG
Enter command or help
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
D:\Work\z390>CD \WORK\Z390\
*** 11/12/06 14:51:09 CMD task started
D:\Work\z390>demo
D:\Work\z390>rem demo hello world demo
D:\Work\z390>call D:\Work\z390\asmlg D:\Work\z390\demo\demo
D:\Work\z390>rem asmlg assemble, link, and go from mlc to 390 execution
C
D
ould Not Find D:\Work\z390\demo\demo.bal
:\Work\z390>erase D:\Work\z390\demo\demo.bal
D:\Work\z390>erase D:\Work\z390\demo\demo.prn
D:\Work\z390>erase D:\Work\z390\demo\demo.obj
D:\Work\z390>erase D:\Work\z390\demo\demo.lst
D:\Work\z390>erase D:\Work\z390\demo\demo.390
D:\Work\z390>erase D:\Work\z390\demo\demo.log
D:\Work\z390>erase D:\Work\z390\demo\demo.err
D:\Work\z390>call D:\Work\z390\mz390 D:\Work\z390\demo\demo
sysmac(D:\Work\z390\mac+.) syscpy(D:\Work\z390\mac+.)
D:\Work\z390>if exist z390.jar goto jar
D:\Work\z390>java -classpath z390.jar -Xrs mz390 D:\Work\z390\demo\demo
sysmac(D:\Work\z390\mac+.) syscpy(D:\Work\z390\mac+.)
AZ390I V1.2.00 Current Date 11/12/06 Time 14:51:09
Copyright 2006 Automated Software Tools Corporation
z390 is licensed under GNU General Public License
AZ390I program = D:\Work\z390\demo\DEMO.BAL
AZ390I options = sysmac(D:\Work\z390\mac+.) syscpy(D:\Work\z390\mac+.)
MZ390I V1.2.00 Current Date 11/12/06 Time 14:51:09
MZ390I Copyright 2006 Automated Software Tools Corporation
MZ390I z390 is licensed under GNU General Public License
MZ390I program = D:\Work\z390\demo\DEMO.MLC
MZ390I options = sysmac(D:\Work\z390\mac+.) syscpy(D:\Work\z390\mac+.)
MZ390I total mnote warnings = 0
MZ390I total mnote errors = 0 max level= 0
MZ390I total errors = 0
AZ390I total errors = 0
AZ390I return code(DEMO) = 0
D:\Work\z390>if errorlevel 5 goto mzerr
D:\Work\z390>call D:\Work\z390\lz390 D:\Work\z390\demo\demo
D:\Work\z390>if exist z390.jar goto jar
D:\Work\z390>java -classpath z390.jar -Xrs lz390 D:\Work\z390\demo\demo
LZ390I V1.2.00 Current Date 11/12/06 Time 14:51:10
LZ390I Copyright 2006 Automated Software Tools Corporation
LZ390I z390 is licensed under GNU General Public License
LZ390I program = D:\Work\z390\demo\DEMO.OBJ
LZ390I options =
LZ390I total errors = 0
LZ390I return code(DEMO) = 0
D:\Work\z390>if errorlevel 5 goto lzerr
```

z390 User Guide V1.5.01

```
D:\Work\z390>call D:\Work\z390\ez390 D:\Work\z390\demo\demo
D:\Work\z390>if exist z390.jar goto jar
D:\Work\z390>java -classpath z390.jar -Xrs ez390 D:\Work\z390\demo\demo
EZ390I V1.2.00 Current Date 11/12/06 Time 14:51:11
EZ390I Copyright 2006 Automated Software Tools Corporation
EZ390I z390 is licensed under GNU General Public License
EZ390I program = DEMO
EZ390I options =
HELLO WORLD
EZ390I total errors          = 0
EZ390I return code(DEMO    )= 0
D:\Work\z390>if errorlevel 5 goto ezerr
D:\Work\z390>goto end
D:\Work\z390>
D:\Work\z390>exit
*** 11/12/06 14:51:11 CMD task ended TOT SEC=2 TOT LOG IO=61
Z390I total errors = 0
```

z390 User Guide V1.5.01

6.4 Output demo.bal mz390 expanded basic assembler language file

```
*****
* Copyright 2005 Automated Software Tools Corporation *
* This source code is part of z390 assembler/emulator package *
* The z390 package is distributed under GNU general public license *
* Author - Don Higgins *
* Date - 09/30/05 *
*****
        TITLE 'Z390 HELLO WORLD DEMO'
* YOU CAN ASSEMBLE, LINK, AND EXECUTE THIS DEMO
* USING Z390 GUI INTERFACE BY ENTERING DEMO IN COMMAND BOX
* OR BY EXECUTING DEMO FROM WINDOWS COMMAND LINE
* WITH CURRENT DIRECTORY SET TO Z390 INSTALL DIRECTORY WHICH
* IS "C:\PROGRAM FILES\AUTOMATED SOFTWARE TOOLS\Z390"
* THE Z390 TOOLKIT IS DISTRIBUTED IN SOURCE AND EXECUTABLE
* FORMAT UNDER OPEN SOURCE GPL LICENSE. VISIT WWW.Z390.ORG
* FOR MORE INFORMATION.
*MCALL #= 1 LV= 1 LN= 17 DEMO SUBENTRY
DEMO    CSECT
        USING *,15
        STM 14,12,12(13)
        BAL 15,**+4+18*4
        DROP 15
        USING *,13
        DS 18F
        ST 15,8(13)
        ST 13,4(15)
        LR 13,15
*MEXIT #= 1 LV= 1 SUBENTRY
*MCALL #= 2 LV= 1 LN= 18 WTO 'HELLO WORLD'
        BRAS 1,**+(WTO#2_EOT-**+1)/2*2
        DC AL2(WTO#2_EOT-*,0),C'HELLO WORLD'
WTO#2_EOT EQU *
        SVC 35
*MEXIT #= 2 LV= 1 WTO
*MCALL #= 3 LV= 1 LN= 19 SUBEXIT
        LA 15,0
        L 13,4(13)
        LM 0,12,20(13)
        L 14,12(13)
        BR 14
*MEXIT #= 3 LV= 1 SUBEXIT
* Stats total MLC/MAC loaded = 90
* Stats total BAL output lines= 42
* Stats total BAL instructions= 18
* Stats total macros = 4
* Stats total macro loads = 4
* Stats total macro calls = 3
* Stats total global set names= 41
* Stats tot global seta cells = 3
* Stats tot global setb cells = 3
* Stats tot global setc cells = 35
* Stats max local pos parms = 2
* Stats max local key parms = 10
* Stats max local set names = 11
* Stats max local seta cells = 4
```

z390 User Guide V1.5.01

```
* Stats max local setb cells = 0
* Stats max local setc cells = 7
* Stats total array expansions= 0
* Stats total Keys           = 981
* Stats Key searches         = 1180
* Stats Key avg comps        = 0
* Stats Key max comps        = 3
* Stats total macro line exec = 57
* Stats total pcode line exec = 0
* Stats total pcode line gen. = 12
* Stats total pcode line reuse= 0
* Stats total pcode op   gen. = 44
* Stats total pcode op   exec = 0
* Stats total pcode gen  opt  = 16
* Stats total pcode exec opt  = 0
* Stats total milliseconds = 376
* Stats instructions/second = 151
* MZ390I FID= 1 ERR= 0 D:\Work\z390\demo\DEMO.MLC
* MZ390I FID= 2 ERR= 0 D:\WORK\Z390\MAC\SUBENTRY.MAC
* MZ390I FID= 3 ERR= 0 D:\WORK\Z390\MAC\WTO.MAC
* MZ390I FID= 4 ERR= 0 D:\WORK\Z390\MAC\SUBEXIT.MAC
      END
```

z390 User Guide V1.5.01

6.5 Output demo.prn az390 assembler listing file

```
AZ390I V1.2.00 Current Date 11/12/06 Time 14:55:46
Copyright 2006 Automated Software Tools Corporation
z390 is licensed under GNU General Public License
AZ390I program = D:\Work\z390\demo\DEMO.BAL
AZ390I options = sysmac(D:\Work\z390\mac+.) syscpy(D:\Work\z390\mac+.)
bal
External Symbol Definitions
  ESD=0001 LOC=00000000 LEN=00000088 TYPE=CST NAME=DEMO
Assembler Listing
000000          1
*****
000000          2 * Copyright 2005 Automated
Software Tools Corporation      *
000000          3 * This source code is part
of z390 assembler/emulator package *
000000          4 * The z390 package is
distributed under GNU general public license *
000000          5 * Author - Don Higgins
*
000000          6 * Date   - 09/30/05
*
000000          7
*****
000000          8          TITLE 'Z390 HELLO
WORLD DEMO'
000000          9 * YOU CAN ASSEMBLE, LINK,
AND EXECUTE THIS DEMO
000000         10 * USING Z390 GUI INTERFACE
BY ENTERING DEMO IN COMMAND BOX
000000         11 * OR BY EXECUTING DEMO
FROM WINDOWS COMMAND LINE
000000         12 * WITH CURRENT DIRECTORY
SET TO Z390 INSTALL DIRECTORY WHICH
000000         13 * IS "C:\PROGRAM
FILES\AUTOMATED SOFTWARE TOOLS\Z390"
000000         14 * THE Z390 TOOLKIT IS
DISTRIBUTED IN SOURCE AND EXECUTABLE
000000         15 * FORMAT UNDER OPEN SOURCE
GPL LICENSE. VISIT WWW.Z390.ORG
000000         16 * FOR MORE INFORMATION.
000000          17 DEMO          SUBENTRY
000000         18+DEMO          CSECT
000000         19+          USING *,15
LISTUSE DEMO      ESD=0001 LOC=00000000 LEN=01000 REG=F OFF=00000 LAB=
000000 90ECD00C   20+          STM
14,12,12(13)
000004 45F0F050   21+          BAL   15,*,+4+18*4
000008           22+          DROP   15
LISTUSE NONE
000008           23+          USING *,13
LISTUSE DEMO      ESD=0001 LOC=00000008 LEN=01000 REG=D OFF=00000 LAB=
000008           24+          DS    18F
000050 50F0D008   25+          ST    15,8(13)
000054 50D0F004   26+          ST    13,4(15)
000058 18DF       27+          LR    13,15
```

z390 User Guide V1.5.01

00005A	18	29	WTO	'HELLO
WORLD'				
00005A A715000A		30+	BRAS	
1,*(WTO#2_EOT-#+1)/2*2				
00005E 000F0000C8C5D3D3		31+	DC	
AL2(WTO#2_EOT-*,0),C'HELLO WORLD'				
00006D		32+WTO#2_EOT EQU *		
00006E 0A23		33+	SVC	35
000070	19	35	SUBEXIT	
000070 41F00000		36+	LA	15,0
000074 58D0D004		37+	L	13,4(13)
000078 980CD014		38+	LM	0,12,20(13)
00007C 58E0D00C		39+	L	14,12(13)
000080 07FE		40+	BR	14
000082		42 *	Stats total	MLC/MAC
loaded = 90				
000082		43 *	Stats total	BAL output
lines= 42				
000082		44 *	Stats total	BAL
instructions= 18				
000082		45 *	Stats total	macros
= 4				
000082		46 *	Stats total	macro loads
= 4				
000082		47 *	Stats total	macro calls
= 3				
000082		48 *	Stats total	global set
names= 41				
000082		49 *	Stats tot	global seta
cells = 3				
000082		50 *	Stats tot	global setb
cells = 3				
000082		51 *	Stats tot	global setc
cells = 35				
000082		52 *	Stats max	local pos
parms = 2				
000082		53 *	Stats max	local key
parms = 10				
000082		54 *	Stats max	local set
names = 11				
000082		55 *	Stats max	local seta
cells = 4				
000082		56 *	Stats max	local setb
cells = 0				
000082		57 *	Stats max	local setc
cells = 7				
000082		58 *	Stats total	array
expansions= 0				
000082		59 *	Stats total	Keys
= 981				
000082		60 *	Stats Key	searches
= 1180				
000082		61 *	Stats Key	avg comps
= 0				
000082		62 *	Stats Key	max comps
= 3				

z390 User Guide V1.5.01

```
000082          63 * Stats total macro line
exec = 57
000082          64 * Stats total pcode line
exec = 0
000082          65 * Stats total pcode line
gen. = 12
000082          66 * Stats total pcode line
reuse= 0
000082          67 * Stats total pcode op
gen. = 44
000082          68 * Stats total pcode op
exec = 0
000082          69 * Stats total pcode gen
opt = 16
000082          70 * Stats total pcode exec
opt = 0
000082          71 * Stats total milliseconds
= 376
000082          72 * Stats
instructions/second = 151
000082          73 * MZ390I FID= 1 ERR= 0
D:\Work\z390\demo\DEMO.MLC
000082          74 * MZ390I FID= 2 ERR= 0
D:\WORK\Z390\MAC\SUBENTRY.MAC
000082          75 * MZ390I FID= 3 ERR= 0
D:\WORK\Z390\MAC\WTO.MAC
000082          76 * MZ390I FID= 4 ERR= 0
D:\WORK\Z390\MAC\SUBEXIT.MAC
000082          77          END
```

Symbol Table Listing

```
SYM=DEMO      LOC=00000000 LEN=00000088 ESD=0001 TYPE=CST  XREF=18
SYM=WTO#2_EOT LOC=0000006D LEN=00000001 ESD=0001 TYPE=REL  XREF=32 30
31
```

Literal Table Listing

```
Stats BAL lines      = 77
Stats symbols        = 2
Stats Literals       = 0
Stats alloc passes   = 2
Stats Keys           = 867
Stats Key searches   = 1005
Stats Key avg comps  = 0
Stats Key max comps  = 3
Stats ESD symbols    = 1
Stats object bytes   = 57
Stats object rlds    = 0
Stats total seconds  = 0
MZ390I total mnote warnings = 0
MZ390I total mnote errors  = 0 max level= 0
MZ390I total errors       = 0
AZ390I total errors       = 0
AZ390I return code(DEMO   )= 0
```

z390 User Guide V1.5.01

6.6 Output demo.obj az390 assembler object file

```
.ESD ESD=0001 LOC=00000000 LEN=00000080 TYPE=CST NAME=DEMO
.TXT ESD=0001 LOC=00000000 LEN=08 90ECD00C45F0F050
.TXT ESD=0001 LOC=00000050 LEN=10 50FD000850DF000418DF4510D066000F
.TXT ESD=0001 LOC=00000060 LEN=0D 0000C8C5D3D3D640E6D6D9D3C4
.TXT ESD=0001 LOC=0000006E LEN=10 0A2358DD0004980CD01458ED000C07FE
.END
```

z390 User Guide V1.5.01

6.7 Output demo.lst lz390 linker listing

```
LZ390I V1.2.00 Current Date 11/12/06 Time 14:55:47
LZ390I Copyright 2006 Automated Software Tools Corporation
LZ390I z390 is licensed under GNU General Public License
LZ390I program = D:\Work\z390\demo\DEMO.OBJ
LZ390I options = bal
LZ390I ESD=DEMO      LOC=00000000 LEN=00000088
Stats total obj files      = 1
Stats total esds           = 1
Stats total csects        = 1
Stats total entries        = 0
Stats missing wxtrn's     = 0
Stats Keys                 = 1
Stats Key searches         = 2
Stats Key avg comps        = 0
Stats Key max comps        = 1
Stats total obj bytes      = 57
Stats total obj rlds       = 0
Stats total seconds        = 0
LZ390I total errors        = 0
LZ390I return code(DEMO    )= 0
```

z390 User Guide V1.5.01

6.8 Output demo.390 lz390 load module file (superzap dump of binary file)

```
SuperZap V1.0.09
Copyright 2005 Automated Software Tools Corporation
SuperZap is licensed under GNU General Public License
Current Date = 08/12/05 Time = 09:53:13
Log file = D:\Work\eclipse\WORKSP~1\superzap\superzap.log
Enter command or help
OPEN
open "D:\Work\eclipse\workspace\z390\DEMO.390"
Data file D:\Work\eclipse\workspace\z390\DEMO.390
Data file Date = 08/12/05 Time = 09:19:35
Data file len(hex) = 94 len(dec) = 148
dump
00000000 *31303032 54463F3F 00000080 00000000* *1002TF??.....*
00000010 *00000000 90ECD00C 45F0F050 00000000* *.....E..P....*
00000020 *00000000 00000000 00000000 00000000* *.....*
00000030 *00000000 00000000 00000000 00000000* *.....*
00000040 *00000000 00000000 00000000 00000000* *.....*
00000050 *00000000 00000000 00000000 00000000* *.....*
00000060 *00000000 50FD0008 50DF0004 18DF4510* *....P...P....E.*
00000070 *D066000F 0000C8C5 D3D3D640 E6D6D9D3* *.f.....@.....*
00000080 *C4000A23 58DD0004 980CD014 58ED000C* *...#X.....X...*
00000090 *07FE0000 * *..... *
SuperZap total errors = 0
```

z390 User Guide V1.5.01

6.9 Output demo.log ez390 execution trace file (use ez390 demo trace)

```
EZ390I V1.2.00 Current Date 11/12/06 Time 14:55:47
EZ390I Copyright 2006 Automated Software Tools Corporation
EZ390I z390 is licensed under GNU General Public License
EZ390I program = DEMO
EZ390I options = bal
HELLO WORLD
EZ390I Stats total instructions      = 13
EZ390I Stats current date 11/12/06 time 14:55:47
EZ390I Stats total seconds          = 0
EZ390I Stats instructions/sec       = 13000
EZ390I total errors                  = 0
EZ390I return code(DEMO             )= 0
```

z390 User Guide V1.5.01

6.10 Output demo.tre trace file showing instructions, WTO's, etc.

```
EZ390I EZ390I V1.2.00 Current Date 11/12/06 Time 15:03:31
EZ390I EZ390I Copyright 2006 Automated Software Tools Corporation
EZ390I EZ390I z390 is licensed under GNU General Public License
EZ390I EZ390I program = DEMO
EZ390I EZ390I options = bal trace

 800FFF78 0 90ECD00C      STM   RE=00002018 RC=00000000
S2(0000210C)=00000000
 800FFF7C 0 45F0F050      BAL   RF=800FFF78 S2(000FFFC8)=50F0D008 ST

 800FFFC8 0 50F0D008      ST    RF=800FFF80 S2(00002108)=00000000
 800FFFC0 0 50D0F004      ST    RD=00002100 S2(000FFF84)=00000000
 800FFFD0 0 18DF          LR    RD=00002100 RF=800FFF80
 800FFFD2 0 A715000A      BRAS  R1=00002300 S2(000FFFE6)=0A23 SVC

 800FFFE6 0 0A23          SVC   I1=23 WTO
R1=ADDR(AL2(LEN),AL2(FLAGS),C'MSG')
EZ390I HELLO WORLD
 800FFE8 0 41F00000      LA    RF=00000000 S2(00000000)
 800FFEC 0 58D0D004      L     RD=800FFF80 S2(000FFF84)=00002100
 800FFF0 0 980CD014      LM    R0=800FFF78 RC=00000000
S2(00002114)=800FFF78
 800FFF4 0 58E0D00C      L     RE=00002018 S2(0000210C)=00002018
 800FFF8 0 07FE          BR    RE(00002018)=0A03 SVC

 80002018 0 0A03          SVC   I1=03 EXIT
EZ390I EZ390I Stats total instructions      = 13
EZ390I Stats Keys                          = 1589
EZ390I Stats Key searches                   = 18
EZ390I Stats Key avg comps                  = 0
EZ390I Stats Key max comps                  = 1
EZ390I EZ390I Stats current date 11/12/06 time 15:03:31
EZ390I EZ390I Stats total seconds           = 0
EZ390I EZ390I Stats instructions/sec       = 764
EZ390I EZ390I total errors                  = 0
EZ390I EZ390I return code(DEMO)
```