z390 User Guide V1.0.09

z390 Portable Mainframe Assembler and Emulator User Guide for V1.0.09

Table of Contents

z390 User Guide V1.0.09

1.0 Introduction

The z390 macro assembler, linker, and emulator toolkit provides a way to develop, test, and deploy mainframe compatible assembler programs using any computer that supports the Sun Microsystems Java J2SE 1.5.0 runtime. The entire toolkit 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 other systems.

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. Each assembler program has option to open individual gui interface supporting the following views:

- **MCS Console view for WTO and WTOR messages**
- **TN3270 terminal view for TPUT and TGET edit or native 3270 data streams.**
- **Graphic display view for graphics drawn by GUI macro commands.**

The graphical user interfaces conforms 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. The only requirement to run z390.jar is installation of the free Java J2SE runtime 1.5.0 from Sun Microsystems.

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.

If you have not already installed the free version of Sun Microsystems Java 2 J2SE runtime 1.5.0 update 5+, it can be downloaded and installed from the following web site (go to website and click on "Download JRE):

http://java.sun.com/j2se/1.5.0/download.html

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 runtime. The runtime install download file for J2SE 1.5.0 runtime for Windows is about 16 MB and has the name:

jre-1_5_05-windows-i586.exe.

Note 1_5_05 is the last version which has been tested on XP with z390. The z390.jar executable has been built using 1_5_04 versus 1_5_05 to prevent update version compatibility issue on Windows/2000 running with 1_5_b05.

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**

- **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**
- **GUI ON/OFF set option for 390 GUI interface dialog**
- **GUIDE view PDF user guide in web browser**
- **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 calls for MAC file cmd**
- **LOC x y pixels change location of upper left corner of window**
- **MAC mlc file expand mlc macro file to bal source file**
- **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 LOG file**

o **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:**

- **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 emulator (called by ASMLG and EXEC)
- RT – run regression tests
- RTDEMO – run regression test on demo programs
- RTTEST – run regression tests on test programs
  - Visit website testing section for list of all regression tests with descriptions

- You can also start the 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 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 command line you can run any of the z390 toolkit components using the following batch command files:
    - 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. MAC currently supports the following options:
      - ASCII – switch from EBCDIC to ASCII mode which causes the following changes in character handling:
        - C'…' self defining terms have ASCII values.

  - **SETC string compares are done in ASCII versus EBCDIC which results in digits before letters.**
  - **Note mz390 extension supports C"…" self defining ASCII terms and C!...! self defining EBCDIC terms which are not affected by ASCII versus EBCDIC mode option**
- **CON – default copy log messages to console**
- **ERR(nnn) – change max errors from 100 to nnn**
- **LISTFILE – default list all macro and copybook file references**
- **LISTCALL – default generate BAL file comments showing each macro call with parms and each exit from macro**
- **MAXFILE(10) – default maximum output file limit in MB.**
- **MFC – mainframe compatibility mode prevents searching for macros with instruction names and supports type and length operator for ordinary symbol labels**
- **SYM – default support assembler symbols and type references**
- **STATS – default generate statistics comments**
- **SYSBAL(dir) – set directory for BAL file**
- **SYSCPY(dir) – set directory list for CPY files using "+" delimiter to avoid conflict with BAT parm parsing. The batch commands default to mac directory followed by current directory.**
- **SYSDAT(dir) – set directory for text files read via AREAD with optional DDNAME=ddname specified. If parm not specified AREAD data is read from MLC source file**
- **SYSMAC(dir) – set directory list for macro files using "+" as delimiter to avoid conflict with BAT parm parser. The batch commands default to mac directory followed by current directory. On this and other path lists, the search file type can be overridden by adding *.type to path such as sysmac(macros*.mcr) to look for macro s with type .mcr.**
- **SYSMLC(dir) – set directory for MLC files**
- **SYSPCH(dir) – set directory for PCH files**
- **TIME(seconds) - default 15 seconds elapsed execution time seconds (NOTIME turns off time limit checking)**
- **TIMING – default time stamping (NOTIMING turns of time stamping for regression test file compares)**

- **TRACEM – generate mz390 trace information comments**
- **TRACEALL – turns on detail tracing including regular expression parsing detail trace for debugging**
- **TRAP – catch any Java exception traps and log error (NOTRAP turns off trap for debugging)**

- **ASM.BAT – run mz390 and az390 assembler to expand MLC macro assembler source file and generate relocatable OBJ relocatable object code file. Note the OBJ file is in ASCII text format containing hex format for object code (i.e. you can read the OBJ file for debugging purposes). ASM currently supports the following options:**
  - **ASCII – switch from EBCDIC to ASCII mode which causes the following changes in character set handling:**
    - **C'…' generate ASCII character values**
    - **C'…' self defining terms generate ASCII values**
    - **Note: az390 extension supports C"…" ASCII strings and C!..! EBCDIC strings which are not affected by ASCII mode**
  - **CON – default copy log messages to console**
  - **ERR(nnn) – change max errors from 100 to nnn**
  - **LIST – generate PRN listing file with cross reference**
  - **MAXFILE(10) – default maximum output file limit in MB.**
  - **STATS – default generate statistics comments**
  - **SYSBAL(dir) – set directory for BAL source file**
  - **SYSOBJ(dir) – set directory for OBJ output file**
  - **SYSPRN(dir) – set directory for PRN listing file**
  - **TIME(seconds) - default 15 seconds elapsed execution time seconds (NOTIME turns off time limit checking)**
  - **TIMING – default time stamping (NOTIMING turns of time stamping for regression test file compares)**
  - **TRACEM – generate mz390 trace information comments**
  - **TRACEA – generate az390 trace information comments**
  - **TRACEALL – turns on detail tracing including regular expression parsing detail trace for debugging**
  - **TRAP – catch any Java exception traps and log error (NOTRAP turns off trap for debugging)**
- **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. 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 supports the following options:**
  - **AMODE24 – set execution mode to 24 bit addressing**
  - **AMODE31 – default execution 31 bit addressing**
  - **ASCII – switch from EBCDIC to ASCII mode**
  - **CON – default copy log messages to console**
  - **ERR(nnn) – change max errors from 100 to nnn**
  - **LIST – generate LST listing file**
  - **MAXFILE(10) – default maximum output file limit in MB.**
  - **RMODE24 – default load in 24 bit address memory**
  - **RMODE31 – load in 31 bit addressing memory**
  - **STATS – default generate statistics comments**
  - **SYSOBJ (dir) – set directory list for OBJ files delimited by "+" to avoid conflict with BAT parm parsing.**
  - **SYS390(dir) – set directory for 390 load module file**
  - **SYSLST(dir) – set directory for LST listing file**
  - **TIME(seconds) - default 15 seconds elapsed execution time seconds (NOTIME turns off time limit checking)**
  - **TIMING – default time stamping (NOTIMING turns of time stamping for regression test file compares)**
  - **TRACEL – generate lz390 trace information comments**
  - **TRACEALL – turns on detail tracing including regular expression parsing detail trace for debugging**
  - **TRAP – catch any Java exception traps and log error (NOTRAP turns off trap for debugging)**
  - **XREF – generate symbol and literal cross reference by line number.**
- **EXEC.BAT – run ez390 emulator to run 390 load module. The emulator supports 32 and 64 bit problem state IBM processor instructions including HFP and BFP floating point instructions. The emulator supports the following command options:**
  - **ASCII – switch from EBCDIC to ASCII mode which causes the following changes in character set handling during execution:**
    - **memory dumps show ASCII characters**
    - **test character constant C'…' ASCII values.**
    - **DDNAME, DSNAME, LINK, LOAD, DELETE names in storage are all ASCII**

  - **ED and EDMK still require EBCDIC input mask but translate output filed to ASCII**
  - **MAXFILE(10) – default maximum output file limit in MB.**
  - **PACK is unchanged and all packed decimal instructions allow X'3' ASCII zone as positive sign**
  - **UNPK inserts X'3' zone for digits and sign byte nibbles are reversed in both modes.**
  - **PKA and UNPKA are unchanged**
  - **DCB RECFM=FT and VT do no translation**
  - **To make a program ASCII or EBCDIC transparent, all characters including fill characters for CLCL and MVCL must be coded as C'..' type constants. See TESTINS2 and TESTASC3 regression tests.**
- **CON – default copy log messages to console**
- **DUMP – request full dump versus indicative dump**
- **ERR(nnn) – change max errors from 100 to nnn**
- **GUI – enable program to use GUI dialog in any or all of the following display views:**
  - **MCS console display for WTO and WTOR's**
  - **TN3270 display for TPUT and TGET in either EDIT or native TN3270 full screen mode**
  - **Graphic display for graphics drawn by GUI macro commands. See GUI User Guide.**
- **LIST – generate LOG file with WTOs and errors**
- **MEM(mb) – default 1 MB memory below line (set to required memory in MB up to physical limit. For example MEM(32) would allocate 16 MB below line and 16 MB above the line if physical memory available. See regression test TESTMEM1.MLC and RTTEST.BAT for example of memory access above and below line)**
- **REGS – turns on register trace when TRACE is on**
- **STATS – default generate statistics comments**
- **SYS390(dir) – set directory list for 390 load modules delimited by "+" to avoid conflict with BAT parm parsing.**
- **SYSLOG(dir) – set directory for log file**
- **TEST – invoke interactive debug command processor from GUI or command line to support the following sub commands similar to TSO TEST:**
  - **addr=sdt – set memory value (ie 1r?=x'80' changes mem at (r1) 31 bit**
  - **reg=sdt - set register value (i.e. 15r=8 changes reg 15 to 8)**

  - **B=addr - set base for rel addr (ie B=15r% sets base to (r15) 24 bit**
  - **D – display DCB file information from TIOT**
  - **F – display floating point registers F0-FF**
  - **G nn/opcode - go exec n instr. or until specified opcode is found or reg/mem break occurs**
  - **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 and memory breaks**
  - **S reg??sdt - set break on register change**
  - **S addr??sdt - set break on memory change**
  - **T nn/opcode - trace n instr. or until specified opcode is found or reg/mem break occurs. Note one instruction is always executed before next opcode break even if it's the same instruction such as a BCT 1,*.**
  - **Z nn/opcode - zoom n instr. or until specified opcode is found or next reg/mem break occurs**
  - *** 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 (=,!=,<,<=,>,>=)**
- **TEST(DDNAME) – invokes batch test mode reading command from file defined by DDNAME.**
- **TIME(seconds) - default 15 seconds elapsed execution time seconds (NOTIME turns off time limit checking)**
- **TIMING – default time stamping (NOTIMING turns of time stamping for regression test file compares)**
- **TRACE – generate ez390 trace information on log file**
- **TRACEALL – turns on detail tracing including regular expression parsing detail trace for debugging**
- **TRACEMEM – trace all memory FQE updates to help debug any memory corruption (you can use TEST**

break on memory change to find where corruption is occurring once you locate address of corrupted memory).

- **TRAP – catch any Java exception traps and log error (NOTRAP turns off trap for debugging)**

- **The emulator also supports basic OS svc services compatible at the macro interface such as:**
  - **ABEND - abort program with specified code**
  - **CALL - call subroutine using standard linkage**
  - **CHECK - check completion of READ/WRITE**
  - **CLOSE - close sequential or random access DCB**
  - **CMDPROC - control command processing tasks**
  - **CVTD - DSECT of limited CVT fields supported**
  - **DCB = data control block for file**
  - **DCBD - DSECT of limited DCB fields**
  - **DECBD - DSECT of limited DECB fields**
  - **DELETE - freemain 390 load module if use count 0**
  - **ESPIE - set program interruption handler**
  - **ESTAE - set program abend handler**
  - **FREEMAIN - release memory area**
  - **GET - read sequential file record**
  - **GETMAIN - allocate memory area**
  - **LINK - load and execute 390 load module**
  - **LOAD - load 390 load module**
  - **OPEN - open DCB sequential or random file**
  - **POINT - position to relative record or RBA in file**
  - **PUT - write sequential record to file**
  - **READ - read block from file at current position**
  - **RETURN - restore saved registers and return**
  - **SAVE - save specified registers**
  - **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**
  - **TGET - read from GUI TN3270 interface**
  - **TIME - get time and date in requested format**
  - **TPUT - write to GUI TN3270 display**
  - **WAIT - wait for ECB to be posted by other task**
  - **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 macro extensions are also supported:**

- **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.**
- **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 R13 as base pointing to save area 8 bytes into CSECT.**
- **SUBEXIT RC=0 – generate standard exit linkage from CSECT setting R15 to 0 or value specified for RC=.**
- **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.**
    - **TN3270 – supports definition of TN3270 data streams using predefined control code names and row and column conversions for set buffer addresses. See DEMOGUI4.**
- **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:**
        - **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 (runs in edit mode for v1.0.08 initial release so you can see control codes but they are not being decoded yet).**
- **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.**
  - **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 DSNAME field.**

        - **TESTERR5.MLC – test SYNAD exit after error on bad DSNAME 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.**
    - **The ASCII option can be specified to switch from EBCDIC to ASCII resulting in C'…' type strings generating ASCII versus EBCDIC values and macro string comparisons to be done in ASCII versus EBCDIC which causes digits and letter sequence to be reversed.**
    - **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
  - Turn on TRACE option and rerun failing process to get more information on where process is failing. TRACE information appears on BAL file, PRN file, LST file, and LOG file.
  - 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 3.1 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.
  - 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.

z390 User Guide V1.0.09

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

IBM is a trademark for International Business Machines

Notes:

- **12/23/05 – v1.0.09 maintenance and enhancement release updates:**
  - **RPI 57 - Add symbol cross reference option to az390. New XREF option for az390 is on by default. Specify NOXREF to turn off the cross reference for symbols and literals.**
  - **RPI 103 - Add TEST support for multiple + or - operands in memory addresses. For example you can now entry L 12R?+11R?+13F. to add base R12+ index R11 + offset.**
  - **RPI 127 - Allow user specified file type overrides on filename parm and directory lists using path*.type. See new regression test TESTMCR1.ASM which uses macros with MCR suffix.**
  - **RPI 130 - Add the following mz390 system variables:**
    - **&SYSASM = z390**
    - **&SYSCLOCK = YYYY-MM-DD HH:MM:SS.mmmmmm**
    - **&SYSDATC = YYYYMMDD year, month, day**
    - **&SYSDATE = MM/DD/YY month, day, year within cc**
    - **&SYSECT = current section**
    - **&SYSLOC = current LOCTR or last section**
    - **&SYSNEST = macro call nesting level**
    - **&SYSPARM = mz390 user option sysparm(xxxxx)**
    - **&SYSTEM_ID = operating system and version**
    - **&SYSTEM_JAVA - Java vendor and version**
    - **&SYSTIME = HH.MM time of day**
    - **&SYSVER.= version of z390 macro assembler**
    - **See TESTSYS1 regression test for all of above. Note when option NOTIMING is specified the same time and date is always returned - 01/02/05 33:44:55.567000.**
  - **RPI 133 - Add MAXFILE option with default of 10 MB maximum size for any output file from mz390, az390, lz390, or ez390. You can override with MAXFILE(nn) in MB.**
  - **RPI 138 - Change batch commands to place sysmac and syscpy parms before user overrides to they can be changed by user.**
  - **RPI 139 - Allow comments on conditional macro instructions such as LCLA, SETA, AIF, etc..**
  - **RPI 140 - Correct handling of * and .* comments in macros to avoid erroneous opcode processsing.**
  - **RPI 141 - Correct handling of lower case macro opcodes.**
  - **RPI 142 - Add mz390 support for T' operator types:**
    - **ABCDEFHLPSVXYZ and explicit length types:**
    - **RBCKKGGKPRVXRZ plus O, J, I, and M types for omitted, control section, insruction, and macro labels. See TESTSYM1 for additional regression tests added.**
  - **RPI 143 - Correct N'&SYSLIST to return actual positional parm count rather than minimum of positional parms defined on prototype statement.**
  - **RPI 144 - Correct precedence in mz390 expression parser to make + or - higher than logical compare**
  - **RPI 145 - Correct precedence in mz390 expression parser to prevent error when K' operator appears as only token in**

substring starting offset expression ot when two K' operators appear in succession.
    - RPI 146 - Add new mainframe compatibility option for mz390 called MFC as default which prevents instruction opcodes from being treated as macros unless previously defined via inline macro. This option for mz390 replaces option SYM to request ordinary symbol label type and length support during macro processing. To turn off both opcode checking and symbol support, specify NOMFC. Also new statistic added showing unique instruction count.
    - RPI 147 - For mainframe compatibility correct NOT for SETB type variables to return 0 or 1 rather than complement (ie -1 for NOT 1).
- 12/16/05 – V1.0.08 maintenance and enhancement release updates:
    - RPI 47 - Add GUI option to ez390 which enables GUI dialog window with 3 different view options:
        - 1) MCS console view for WTO and WTOR messages. Run DEMOGUI1 and DEMOGUI2 with GUI option for demo.
        - 2) TN3270 view for TPUT and TGET with EDIT and FULLSCR options. Run DEMOGUI3 and DEMOGUI4 with GUI option for demo (FULLSCR support is not done yet and uses EDIT mode for now - see RPI 136).
        - 3) GRAPH view support for graphics output generated using GUI macro graphics options. Run DEMOGUI5 with GUI option for demo (Graphics support not done yet so demo just shows title and text output on graphic display for now - see RPI 137). Note this demo also outputs MCS and TN3270 messages so you can view all 3 displays via menu. See new z390 GUI Graphical User Interface Guide for supported macro interfaces.
    - RPI 105 - Research issue with maximum memory that can be allocated via MEM(mb) option on a system. The J2RE virtual machine option -Xmx can be used to set maximum memory available. See FAQ for more information and example.
    - RPI 108 - Improve performance of additional compute intensive instructions:
        - 1) BC no branch reduced from 519 to 343 NS by skipping RX target address fetch if no branch to be taken.
        - 2) AP and all PD speedup by using Long versus BigInteger when values within range.
        - 3) MVCL and MVCLE speedup by using block copy and fill.
        - 4) MVZ, MVN, OC, NC, XC all speeded up by replacing byte loop with Long, Int, Short, byte loops.
        - 5)All instructions speedup by replacing mem.get(...) buffered fetch with mem_byte[...] direct byte array fetch, also mem.put() and reg.get() and reg.put().
        - See Benchmark Testing page for more results.
    - RPI 109 - Fix packed decimal instructions to issue 0C7 versus 0C5 under certain error conditions. Note 0C5 is the default abend for any error not recognized or specifically handled by ez390.
    - RPI 110 - Fix SNAP macro bug setting wrong bit for STORAGE request. Update SNAP svc to correctly format TIOT entries for closed files showing DCBOFLG open bit off. Add check

for register conflict. See TESTDCBD for regression test including SNAP of open and closed DCB's.

- RPI 111 - Fix LOAD support to strip leading and trailing blanks in DSNAME field. This fix causes a LOAD of a file name and suffix matching any previously loaded file or 390 program to be treated as identical resulting in CDE use count being incremended and same address being returned in register 0. Register 1 is set to length in double words or 390 or bytes for any other file. See updated TESTLOD2 regression test.
- RPI 112 - Fix I/O macros:
  - 1) POINT - add missing AGO
  - 2) CLOSE - check for invalid DCB RX form (reg) vs ((reg))
  - 3) Correct EZ390E error 90 for I/O error on READ/WRITE
- RPI 113 - Fix all z390 file handling to correctly handle files with drive letter and colon path but no file separator in the path.
- RPI 114 - Fix WTO and WTOR to use more unique labels to avoid duplicating user labels.
- RPI 115 - Fix mz390 and az390 continuation processing to correctly handle comma and space appearing within literal which is continued (previous truncating line after comma). See TESTDCBA for example.
- RPI 116 - Issue error if no END statement found in az390.
- RPI 117 - Fix GETMAIN and FREEMAIN macros to allow 31 bit length for LV=, support nnnK = nnn * 1<<10 and nnnM = nnn * (1<<20) short form extensions, issue error for missing LV=, and correct spelling errors.
- RPI 118 - Fix LINK, LOAD, XCTL, and DELETE macros to use SR verus LA, check for register conflicts. Correct OI for VL bit setting in LINK, XCTL, and CALL. Remove BALR, USING and add optional register restore in XCTL.
- RPI 119 - Add floating point and hex options to EQUREGS macro. See TESTSET1.MLC for new tests of multiple value sets required for EQUREGS update. See TESTEQU1 regression test for use of REGS=GPR/FPR and TYPE=DEC/HEX options.
- RPI 120 - Fix assembly of SRP instruction when using explicit d2(b2) operand (was dropping base). See updated TESTINS2 regression test.
- RPI 121 - Change TEST subcommand Q to abort rather than continue execution. Fix G xxx where xxx is invalid to just issue error message and return to prompt instead of running.
- RPI 122 - Add support for following az390 statements:
  - 1) AMODE, RMODE - ignored (use linker options)
  - 2) AEJECT, ASPACE, CSPACE, EJECT, PRINT, SPACE
- RPI 123 - Correct multiple directory search logic to correctly handle multiple directories in mz390, lz390, and ez390.
- RPI 124 - Truncate trailing blanks and sequence # beyond 72 during loading of mz390 and az390 source code.
- RPI 125 - Add support for setting multiple SET values in array by adding multiple expressions separated by commas, See updated TESTSET1 regression test.

  - **RPI 126 - Ignore duplicate macro variable array declarations and use the first declaration to set fixed maximum array size. This is a current restriction compared to mainframe which allows arrays to grow dynamically.**
  - **RPI 129 - Fix mz390 expression parser to handle '....' setc strings where sdt of the form c'...' or b'...' or x'...' is returned as regular expression for last character and single quote of literal.**
  - **RPI 131 - Ignore label field in TITLE statements to prevent duplicate label errors.**
  - **RPI 132 - Correct ESPIE macro to correctly handle range of interrupt codes within list of interrupts.**
  - **RPI 134 - Fix all macros to generate label if specified for &N.**
  - **RPI 135 - Move shared tables and file methods to tz390 to reduce redundant code, and improve consistency. Fix shared find file method used by mz390, lz390, and ez390 to support both ; and + path separators.**
- **11/20/05 – V1.0.07 maintenance release with following updates:**
  - **RPI 82 Document instruction timing results on website and look for ways to speed-up frequently used instructions. Performance RPI's will be accepted including requests to add new svcs and/or opcodes to "microcode" frequently used instructions loops that need more speed. See instruction timming benchmark test program BMKINS1.MLC with results posted on Benchmark page of www.z390.org. The first performance improvement completed was replacement of MVC byte loop with single function move or fill function resulting in reduction of 256 byte move from 41 microseconds to 880 nanoseconds.**
  - **RPI 92 Correct instruction trace format errors in test versus normal instruction trace due to type index off by 1 causing those instructions such as CVDG at beginning of new type to be formatted as previous type. See BMKINS1.MLC for usage of CVDG.**
  - **RPI 93 Correct SNAP svc x'51' which was incorrectly using ending storage address as length instead of ending address. See updated TESTDMP4.MLC.**
  - **RPI 94 Add new extended TIME NS macro option to return high resolution nano-second timer value useful for interval measurements. See BMKINS1 for example usage.**
  - **RPI 95 Update LOAD macro with DDNAME= and DSNAME= support identical to LINK macro. Add new XCTL macro using svc 7. Add DDNAME= and DSNAME= conflict check on LINK, LOAD, XCTL, and DCB. See TESTLNK1 for LINK tests. See TESTLOD1 for LOAD tests, and see TESTXCL1-3 for XCTL test. See TESTERR3 for conflict error tests.**
  - **RPI 96 Add validity checking for DCB LRECL, BLKSIZE and RECFM. Accept both OPEN file and OPEN (reg) as extensions for input. Correct missing AGO in POINT macro. See TESTERR3.**
  - **RPI 98 Add new DUMP and ASCII mode to the GUI option menu. Also correct failure observed on first batch command after changing menu options due to not correctly flushing queues between end of previous batch initiator process and restart. Manual test of batch commands with DUMP and ASCII options tested.**

  - RPI 99 Correct mz390 AIF expression parser bug when substring expression appears after logical operator in complex expression. See TESTERR3 tests using complex AIF testsin DCB macro.
  - RPI 100 When running in TEST mode, issue ABEND dump if requested before returning to test prompt. Also reformat ABEND dump to include system/user type ABEND codes with PSW and current instruction. If dump requested by caller or system dump option then also display GPR and FPR registers, TIOT DCB's if any, CDE program entries, and dump of each program. See TESTDMP1 to TESTDMP4 for examples of ABEND and SNAP output.
  - RPI 101 Correct erroneous read past end of file error due to not resetting TIOT DCB rba pointer on reopen of existing entry. See new TESTDCBD and TESTDCBE (same test run in ASCII mode) regression test program provided by MM. Note ASCII mode does not translate RT or VT input so all the hex values including EBCDIC characters are retained as is in input records.
  - RPI 102 Add additional LOAD and DELETE macro support:
    - Return 390 file length as double word count in R1 for OS compatibility.
    - If DDNAME or DSNAME point to file spec with path, file name, and suffix other than 390 then set R0 to load point for file content and set R1 to total file length in bytes.
    - Allow DDNAME or DSNAME to replace EP or EPLOC.
    - See TESTLOD2 regression test loading 2 text files.
  - RPI 106 Correct following instruction bugs:
    - SPM and IPM not putting mask in correct position (MM)
    - NR, OR, and XR condition code not set (MM)
    - PR restoring R15 incorrectly (IS)
    - See TESTINS2, TESTFP1, AND ESPIE macro updated regression tests.

- 11/14/05 – V1.0.06 maintenance release with following updates:
  - RPI73 - Update ED and EDMK to support ASCII option by translating output bytes to ASCII. Input mask must be EBCDIC. Add new DC C!...! option to define EBCDIC bytes for mask which is not affected by ASCII mode. Add PKA and UNPKA instructions. See TESTINS1 and copy TESTASC3 run in ASCII mode for example.
  - RPI74 - Flag any relative instruction branch to odd address rather than generating incorrect branch off by 1. See TESTERR2 updates.
  - RPI75 - Add dump option which defaults to indicative dump of just PSW and general registers on any abend when not in test mode. Specifying DUMP option will generate dump of program storage on any abend except when in test mode. Specifying DUMP parm on ABEND macro will override the DUMP option. See TESTDMP1 - TESTDMP4 for default indicative dump, dump option, user abend dump, and new SNAP dump macro with options to dump general and floating point registers, memory totals, loaded programs in CDE, and open files in TIOT.
  - RPI76 - Cancel all command processes running when ez390 terminates.

  - **RPI77 - Set 15 = 0 if CMDPROC record read successfully. Set R15 = 4 if there is no output line ready after specified number of milliseconds WAIT=500 default. Set R15 = 8 if no record and task has ended. Set R15 = 12 if process error occurred. See TESTCMD1 and TESTCMD2.**
  - **RPI79 - Add support for up to 10 concurrent CMDPROC tasks using keyword task ID= which defaults to first task ID=0. See TESTCMD2 with 2 processes.**
  - **RPI80 - Add check for missing macro labels and duplicate macro labels in mz390 and correct undefined .ERR1 label in WTO macro. See TESTERR1.**
  - **RPI81 - Fix FONT support to change menu item and took tip font size along with the scrolling log text, command line, and status line font size. Manual test ok on Windows with font sizes from 10 - 30.**
  - **RPI84 - Correct mz390 copy expansion parser to ignore comments after file name. See TESTCPY1 updated regression test.**
  - **RPI85 - Issue errors for DC F and H type signed field values that exceed range for length specified. See TESTERR2.**
  - **RPI86 - Add following additional TEST sub-commands:**
    - **D - display DCB file information**
    - **F - display floating point registers**
    - **M - display memory totals**
    - **P - display program information**
    - **R - display general registers**
    - **See TESTTST1 updated regression test.**
  - **RPI87 - Correct erroneous error 102 on ORG statement occurring whenever last expression calc prior to ORG resulted in absolute value rather than RX type symbol in same csect. See TESTASM1.**
  - **RPI88 - Correct error in OPEN and CLOSE macro for DCB register parm. Add DCB validity checking at execution time. Note this RPI requires reassembly of all DCB's with new validation field named DCBID in DCB and DCBD macros. See new TESTERR6 regression test.**
  - **RPI89 - Add option for LOAD and LINK to use extended keyword DDNAME= or DSNAME= defining program file path list or specific file via environment variable or program constant with file specification. If directory path list is specified, then EP or EPLOC is required to define the program name. If a file specification for 390 program is specified, then it overrides any EP or EPLOC name. DSNAME overrides DDNAME. See TESTLNK1 for example regression test.**
  - **RPI90 - Correct v1.0.05 regression error causing trap when single character labels B, C, or X are used due to erroneous test of 2nd character for self defining term quote. See TESTASM1 updated regression test.**
  - **RPI91 - Add support for XLATE macro via svc x'67' to translate between ASCII and EBCDIC. See regression test TESTXLT1**
- **11/04/05 – V1.0.05 maintenance release with following updates:**
  - **RPI58 - Correct STCK and STCKE to be mainframe OS time compatible. STCK stores double word with bit 51**

corresponding to microseconds from January 1, 1900. STCKE stores 16 bytes with bit 59 corresponding to microseconds from January 1, 1900. See TESTTIM1.MLC update.

- RPI61 - Remove duplicate CR,LF from RECFM=FT and VT output format in ASCII mode. No duplicates found, see RPI66 and regression test TESTDCBC.
- RPI62 - Remove extra space from command recall in TEST mode. Manual GUI test.
- RPI63 - Recognize x'1A' as end of file marker on DCB RECFM=FT or VT read and branch to EODAD. Note you can use superzap (www.superzap.net) to view and verify hex codes in any Windows file such as the input and output test file in this regression test. I also used superzap to add the hex x'1a' code to test file with the command sequence OPEN, DUMP, SFL (set file length), and REP command. See TESTEOF1.MLC.
- RPI64 - Correct error handling for invalid DCBDSNAM field with or without TEST mode. The DCB synad handling routine now issues error message followed by S013 abend for consistent handling with or without TEST active. See TESTERR4.MLC.
- RPI65 - Correct BALR, BASR, and BASSM to branch when R1 = R2. See TESTINS2.MLC update.
- RPI66 - Correct ASCII mode padding of RECFM=FT records to use ASCII X'20' versus EBCDIC X'40'. This should also fix RPI61. See TESTDCBC.MLC run in ASCII mode.
- RPI67 - Change TEST mode command G, T, or Z opcode to always execute at least 1 instruction before breaking on next matching opcode. Breaking on BCT to itself now executes once per G, T, or Z BCT. Manual test.
- RPI68 - Add 3 SVC type extended options to the current IBM compatible TIME macro:
  - 1. CLOCKTYPE=STCK store double word with bit 51 = microseconds from Jan. 1, 1900 IBM Epoch.
  - 2. CLOCKTYPE=STCKE store 2 double words with bit 59 = microseconds from Jan 1, 1900 IBM Epoch.
  - 3. CLOCKTYPE=JAVA store double word with bit 63 = microseconds from Jan 1. 1970 Java Epoch.
  - See TESTTIM1.MLC update.
- RPI69 - Change ED and EDMK to map X'40' to ASCII space X'20' when running in ASCII mode. Note the following ASCII characters cannot appear as characters in ASCII mode due to conflicts with predefined special codes:
  - 1. ASCII @ (X'40')
  - 2. ASCII space (X'20')
  - 3. ASCII ! (X'21')
  - 4. ASCII " (X'22')
  - The above codes can be moved into edited fields after ED/EDMK. See TESTASC3.MLC regression test which is run in ASCII mode.
- RPI70 - After install of z390 v1.0.04 on Win/2000 with J2RE 1.5.0_5, system failure with error - java.lang.UnsupportedClassVersionError: mz390 (Unsupported major.minor version 49.0). Planned fix is to build 1.0.05 on XP update 4 instead of XP update 5 since that fixed

win/2000 problem seen back in v1.0.00. Manual test on Windows/2000.
    - RPI71 - ez390 loader incorrectly relocating AL3(*) type address constant. See TESTINS2.MLC update.
- 10/28/05 – V1.0.04 maintenance release with following updates
    - RPI42 – fix RECFM=FT and VT output processing to translate all EBCDIC characters to ASCII characters instead of just printable characters and do not remove leading blanks. See updated TESTDCB4 regression test.
    - RPI43 – correct ED and EDMK instructions to delay significance by 1 position. See TESTINS2.
    - RPI44 – correct TM instruction to set CC3 when AND of mask and test byte equals mask versus test byte. See TESTINS2.
    - RPI45 – correct labels on dump register display.
    - RPI46 – add SVC 34 to support command processing with input and output controls. See TESTCMD1 example which invokes batch command file and reads output.
    - RPI48 – correct GET, PUT, READ, and WRITE macros to detect register 1 conflicts and issue MNOTES. See TESTERR3.
    - RPI49 – correct D instruction to use low half or register pair like DR. See updated TESTINS2.
    - RPI50 – issue 0C1 abend for undefined svc requests versus abort error message so it can be trapped by TEST mode.
    - RPI51 – exit TEST mode on Q command without error message.
    - RPI52 – standardize all error messages using "?z390E error" where ? is G for GUI, M – macro processor, A – assembler, L = linker, and E – execution.
    - RPI53 – set R15 to error number in DCB synad exit.
    - RPI54 – document how to cancel batch tasks in user guide and FAQ on website.
    - RPI55 – update /SC startup option to execute GUI commands from specified file.
    - RPI56 – update ABEND macro and SVC 13 processing to dump registers and program memory to log when DUMP is requested.
    - RPI59 – add DROP 15 to SUBENTRY to avoid multiple USING's.
    - RPI60 – correct mz390 expression parser to correctly handle multiple logical conditions. See GET, PUT, READ, WRITE macro example AIF test for R1 conflicts.
- 10/21/05 – V1.0.03 maintenance release with following updates
    - RPI23 - add string instructions CLST, CUSE, and SRST. See updated TEST\TESTINS2.MLC regression tests.
    - RPI24 - change batch task commands to use %~dp0 path wherever needed instead of changing directory to allow user to use relative paths in alternate directories.
    - RPI25 – change the trace options for mz390, az390, and lz390 to TRACEM, TRACEA, and TRACEL respectively. The TRACEALL option will generate all detail traces.
    - RPI26 – correct DCB missing label .DDDEF
    - RPI27 – issue error message for all macro undefined keyword parameters
    - RPI28 – change default for DCB DSNAME field to EBCDIC. Use ASCII option for all fields to be generated in ASCII vs EBCDIC.
    - RPI29 – change error messages to have consistent error prefix in upper case with I or E for information or error.
    - RPI30 – add SAVE and RETURN standard linkage macros

  - **RPI31 – set r15 to zero on normal svc exit.**
  - **RPI32 – support multiple directory list for SYSMAC, SYSCPY, SYSOBJ, and SYS390 delimited by "+" to avoid conflict with BAT parm parsing**
  - **RPI33 – change default PSW program mask to suppress fixed point overflow. To trap this or other exceptions use ESPIE or ESTAE macros. See regression test TESTINS2 for examples.**
  - **RPI34 – support EBCDIC and ASCII full character set translation instead of only printable characters.**
  - **RPI35 – correct loop in batch task TEST mode execution when the GUI interface is cancelled before batch task is cancelled.**
  - **RPI36 – trap any abend in TEST mode and issue error message and return to the TEST prompt.**
  - **RPI37 – update user guide on use of G, T, Z to break on opcode, and other documentation updates.**
  - **RPI38 – issue error if user supplied register parameters conflict with use of R1 for DCB or DECB in GET, PUT, READ, or WRITE.**
  - **RPI39 – set batch return code to R15 on normal exit else 16.**
  - **RPI40 – correct 2 expression parser bugs in mz390 for '?' string and '&MF (2)'(1,1) type operands where ? is a valid prefix operator such as T' or K' and &MF(2) is any parameter sublist in string.**
  - **RPI41 – add the following additional parameter support in macros:**
    - **CALL – VL positional parm**
    - **LINK – PARAM= and VL=1 keyword parms**
    - **WTO - MF=L keyword parm**
- **10/14/05 – V1.0.02 maintenance release with following updates:**
  - **RPI14 - correct desktop z390 icon to point to install directory when location is changed by user during install**
  - **RPI15 - remove CMD prefix from GUI command line generated entries using file menu to run batch command tasks. And turn off batch task when command is done.**
  - **RPI16 - remove dependency between TEST and TRACE options on the GUI options menu.**
  - **RPI17 - allow help commands such as HELP and ABOUT to be entered via command line by turning off CMD mode at end of each batch task unless user requested has requested it stay on via the view menu or CMD command.**
  - **RPI18 - add support for RECORD= parm on DCB to define default record area for GET, PUT, READ, WRITE if not specified on those macros referencing the DCB.**
  - **RPI19 - add support for DSNAME= on DCB defining ASCII file specification including path and delimited by null or double quotes for long spacey names up to 265 characters.**
  - **RPI20 - correct erroneous 62 error on output file which does not exist yet.**
  - **RPI21 - issue 0C5 addressing exception abend when PSW set to address beyond allocated memory.**
  - **RPI22 - turn time limit option off when test option specified.**
- **10/06/05 - V1.0.01 maintenance release with following updates:**

  - **RPI1 - Trap messages on misspelled dir**
  - **RPI2 - DC duplication field az390 error 51**
  - **RPI3 - Omitted base reg syntax az390 error 35**
  - **RPI4 - Allow changing install directory**
  - **RPI5 - Option ASCII to generate ASCII vs EBCDIC character fields in DC C'...' etc. Also support C"..." ASCII character generation.**
  - **RPI6 - Option to limit errors ERR(nn)**
  - **RPI7 - Document I/O support describing DCB and DDNAME usage.**
  - **RPI8 - Correct "compatible java" on about menu**
  - **RPI9 - L' attribute symbol not found error 98**
  - **RPI10 - ORG without parameter error 35**
  - **RPI11 - DC S(1) error 38 no base reg found**
  - **RPI12 - remove erroneous error 35 after error 45**
  - **RPI13 - correct up and down arrow scrolling of previous commands entered via GUI command line.**
- **09/30/05 - V1.0.00 initial release**
  - **10/03/05 - PTF1 fix for hyper RPI's 1,2,3,9,10,11**

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
```

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

```
z390 V1.0.01
Copyright 2005 Automated Software Tools Corporation
z390 is licensed under GNU General Public License
Current Date = 08/12/05 Time = 09:19:30
Log file = D:\Work\eclipse\WORKSP~1\z390\z390_2005_0812_091929.log
Enter command or help
Starting Windows command process
D:\Work\eclipse\WORKSP~1\z390>rem demo hello world demo
D:\Work\eclipse\WORKSP~1\z390>asmclg demo
D:\Work\eclipse\WORKSP~1\z390>rem assemble link and execute mainframe
assembler program
D:\Work\eclipse\WORKSP~1\z390>call mz390 demo
D:\Work\eclipse\WORKSP~1\z390>if exist lib\z390.jar goto jar
D:\Work\eclipse\WORKSP~1\z390>java -classpath .\lib -Xrs mz390 demo
mz390 V1.0.01 Current Date 08/12/05 Time 08/12/05
Copyright 2005 Automated Software Tools Corporation
z390 is licensed under GNU General Public License
MZ390 program = D:\WORK\ECLIPSE\WORKSP~1\Z390\DEMO.MLC
MZ390 options =
Stats total MLC source      = 42
Stats total BAL output      = 36
Stats total opcodes         = 20
Stats total macro loads     = 4
Stats total macro calls     = 3
Stats total global set var  = 0
Stats total local pos parms = 4
Stats total local key parms = 1
Stats total local set var   = 4
Stats total Keys            = 57
Stats Key searches          = 142
Stats Key avg comps         = 0
Stats Key max comps         = 2
Stats total macro instr.    = 38
Stats total milliseconds    = 188
Stats instructions/second   = 202
MZ390 file=1 path=D:\WORK\ECLIPSE\WORKSP~1\Z390\DEMO.MLC
MZ390 file=2 path=D:\Work\eclipse\WORKSP~1\z390\SUBENTRY.MAC
MZ390 file=3 path=D:\Work\eclipse\WORKSP~1\z390\WTO.MAC
MZ390 file=4 path=D:\Work\eclipse\WORKSP~1\z390\SUBEXIT.MAC
MZ390 total errors          = 0
MZ390 return code           = 0
D:\Work\eclipse\WORKSP~1\z390>goto end
D:\Work\eclipse\WORKSP~1\z390>if errorlevel 1 pause error
D:\Work\eclipse\WORKSP~1\z390>if errorlevel 1 goto :end
D:\Work\eclipse\WORKSP~1\z390>call az390 demo
D:\Work\eclipse\WORKSP~1\z390>if exist lib\z390.jar goto jar
D:\Work\eclipse\WORKSP~1\z390>java -classpath .\lib -Xrs az390 demo
az390 V1.0.01 Current Date 08/12/05 Time 08/12/05
Copyright 2005 Automated Software Tools Corporation
z390 is licensed under GNU General Public License
AZ390 program = D:\WORK\ECLIPSE\WORKSP~1\Z390\DEMO.BAL
AZ390 options =
Stats BAL lines       = 35
Stats symbols         = 3
Stats Literals        = 0
```

```
Stats Keys            = 785
Stats Key searches    = 860
Stats Key avg comps   = 0
Stats Key max comps   = 3
Stats ESD symbols     = 1
Stats object bytes    = 53
Stats object rlds     = 0
Stats total seconds         = 0
AZ390 total errors          = 0
AZ390 return code           = 0
D:\Work\eclipse\WORKSP~1\z390>goto end
D:\Work\eclipse\WORKSP~1\z390>if errorlevel 1 pause error
D:\Work\eclipse\WORKSP~1\z390>if errorlevel 1 goto :end
D:\Work\eclipse\WORKSP~1\z390>call lz390 demo
D:\Work\eclipse\WORKSP~1\z390>if exist lib\z390.jar goto jar
D:\Work\eclipse\WORKSP~1\z390>java -classpath .\lib -Xrs lz390 demo
lz390 V1.0.01 Current Date 08/12/05 Time 08/12/05
Copyright 2005 Automated Software Tools Corporation
z390 is licensed under GNU General Public License
LZ390 program = D:\WORK\ECLIPSE\WORKSP~1\Z390\DEMO.OBJ
LZ390 options =
Stats total obj files = 1
Stats total esds      = 1
Stats Keys            = 1
Stats Key searches    = 1
Stats Key avg comps   = 0
Stats Key max comps   = 0
Stats total obj bytes = 53
Stats total obj rlds  = 0
Stats total seconds         = 0
LZ390 total errors          = 0
LZ390 return code           = 0
D:\Work\eclipse\WORKSP~1\z390>goto end
D:\Work\eclipse\WORKSP~1\z390>if errorlevel 1 pause error
D:\Work\eclipse\WORKSP~1\z390>if errorlevel 1 goto :end
D:\Work\eclipse\WORKSP~1\z390>call ez390 demo
D:\Work\eclipse\WORKSP~1\z390>if exist lib\z390.jar goto jar
D:\Work\eclipse\WORKSP~1\z390>java -classpath .\lib  -Xrs ez390 demo
ez390 V1.0.01 Current Date 08/12/05 Time 08/12/05
Copyright 2005 Automated Software Tools Corporation
z390 is licensed under GNU General Public License
EZ390 program = DEMO
EZ390 options =
WTO MSG = HELLO WORLD
Stats total instructions    = 12
Stats total seconds         = 0
Stats instructions/sec      = 12000
EZ390 total errors          = 0
EZ390 return code           = 0
D:\Work\eclipse\WORKSP~1\z390>goto end
D:\Work\eclipse\WORKSP~1\z390>if errorlevel 1 pause error
z390 total errors = 0
```

6.4 Output demo.bal mz390 expanded basic assembler language 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.
* CALL    1 SUBENTRY - DEMO     SUBENTRY
DEMO     CSECT
         USING *,15
         STM   14,12,12(13)
         BAL   15,*+4+18*4
         USING *,13
         DS    18F
         ST    15,8(13)
         ST    13,4(15)
         LR    13,15
* EXIT    1 SUBENTRY
* CALL    2 WTO      -          WTO 'HELLO WORLD'
         BAL   1,WTO2
         DC    AL2(WTOEND2-*,0),C'HELLO WORLD'
WTOEND2  EQU   *
WTO2     SVC   35
* EXIT    2 WTO
* CALL    3 SUBEXIT  -          SUBEXIT
         L     13,4(13)
         LM    0,12,20(13)
         L     14,12(13)
         BR    14
* EXIT    3 SUBEXIT
         END
* Stats total MLC source      = 42
* Stats total BAL output      = 36
* Stats total opcodes         = 20
* Stats total macro loads     = 4
* Stats total macro calls     = 3
* Stats total global set var  = 0
* Stats total local pos parms = 4
* Stats total local key parms = 1
* Stats total local set var   = 4
* Stats total Keys            = 57
* Stats Key searches          = 142
* Stats Key avg comps         = 0
* Stats Key max comps         = 2
* Stats total macro instr.    = 38
* Stats total milliseconds    = 188
* Stats instructions/second   = 202
* MZ390 file=1 path=D:\WORK\ECLIPSE\WORKSP~1\Z390\DEMO.MLC
* MZ390 file=2 path=D:\Work\eclipse\WORKSP~1\z390\SUBENTRY.MAC
* MZ390 file=3 path=D:\Work\eclipse\WORKSP~1\z390\WTO.MAC
* MZ390 file=4 path=D:\Work\eclipse\WORKSP~1\z390\SUBEXIT.MAC
```

```
* MZ390 total errors          = 0
* MZ390 return code           = 0
```

6.5 Output demo.prn az390 assembler listing file

```
az390 V1.0.01 Current Date 08/12/05 Time 08/12/05
Copyright 2005 Automated Software Tools Corporation
z390 is licensed under GNU General Public License
AZ390 program = D:\WORK\ECLIPSE\WORKSP~1\Z390\DEMO.BAL
AZ390 options =
External Symbol Definitions
 ESD=0001 LOC=00000000 LEN=00000080 TYPE=CST NAME=DEMO
Assembler Listing
000000                                          TITLE 'Z390 HELLO WORLD
DEMO'
000000                                 * AUTHOR.  DON HIGGINS.
000000                                 * DATE.    08/12/05.
000000                                 * REMARKS. YOU CAN ASSEMBLE,
LINK, AND EXECUTE THIS DEMO
000000                                 *          USING Z390 GUI
INTERFACE BY ENTERING DEMO IN COMMAND BOX
000000                                 *          OR BY EXECUTING DEMO
FROM WINDOWS COMMAND LINE
000000                                 *          WITH CURRENT DIRECTORY
SET TO Z390 INSTALL DIRECTORY WHICH
000000                                 *          IS "C:\PROGRAM
FILES\AUTOMATED SOFTWARE TOOLS\Z390"
000000                                 *          THE Z390 TOOLKIT IS
DISTRIBUTED IN SOURCE AND EXECUTABLE
000000                                 *          FORMAT UNDER OPEN
SOURCE GPL LICENSE.  VISIT WWW.Z390.ORG
000000                                 *          FOR MORE INFORMATION.
000000                                 * CALL    1 SUBENTRY - DEMO
SUBENTRY
000000                                 DEMO     CSECT
000000                                          USING *,15
000000 90ECD00C                                 STM   14,12,12(13)
000004 45F0F050               000050            BAL   15,*+4+18*4
000008                                          USING *,13
000008                                          DS    18F
000050 50FD0008                                 ST    15,8(13)
000054 50DF0004                                 ST    13,4(15)
000058 18DF                                     LR    13,15
00005A                                 * EXIT    1 SUBENTRY
00005A                                 * CALL    2 WTO      -
WTO 'HELLO WORLD'
00005A 4510D066               00006E            BAL   1,WTO2
00005E 000F0000C8C5D3D3                         DC    AL2(WTOEND2-
*,0),C'HELLO WORLD'
00006D                 00006D           WTOEND2  EQU   *
00006E 0A23                            WTO2     SVC   35
000070                                 * EXIT    2 WTO
000070                                 * CALL    3 SUBEXIT  -
SUBEXIT
000070 58DD0004                                 L     13,4(13)
000074 980CD014                                 LM    0,12,20(13)
000078 58ED000C                                 L     14,12(13)
00007C 07FE                                     BR    14
00007E                                 * EXIT    3 SUBEXIT
00007E                                          END
```

```
Symbol Table Listing
 SYM=DEMO     TYPE=CST ESD=0001 LOC=00000000 LEN=00000080
 SYM=WTOEND2  TYPE=REL ESD=0001 LOC=0000006D LEN=00000001
 SYM=WTO2     TYPE=REL ESD=0001 LOC=0000006E LEN=00000002
Stats BAL lines       = 35
Stats symbols         = 3
Stats Literals        = 0
Stats Keys            = 785
Stats Key searches    = 860
Stats Key avg comps   = 0
Stats Key max comps   = 3
Stats ESD symbols     = 1
Stats object bytes    = 53
Stats object rlds     = 0
Stats total seconds         = 0
AZ390 total errors          = 0
AZ390 return code           = 0
```

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
```

6.7 Output demo.lst lz390 linker listing

```
lz390 V1.0.01 Current Date 08/12/05 Time 08/12/05
Copyright 2005 Automated Software Tools Corporation
z390 is licensed under GNU General Public License
LZ390 program = D:\WORK\ECLIPSE\WORKSP~1\Z390\DEMO.OBJ
LZ390 options =
Stats total obj files = 1
Stats total esds      = 1
Stats Keys            = 1
Stats Key searches    = 1
Stats Key avg comps   = 0
Stats Key max comps   = 0
Stats total obj bytes = 53
Stats total obj rlds  = 0
Stats total seconds         = 0
LZ390 total errors          = 0
LZ390 return code           = 0
```

z390 User Guide V1.0.09

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
```

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

```
ez390 V1.0.01 Current Date 08/12/05 Time 08/12/05
Copyright 2005 Automated Software Tools Corporation
z390 is licensed under GNU General Public License
EZ390 program = DEMO
EZ390 options = trace
 80FFFF80 0 90ECD00C     STM   RE=000002A6 RC=00000000
S2(00000222)=00000000
 80FFFF84 0 45F0F050     BAL   RF=80FFFF80 S2(00FFFFD0)=50FD0008
 80FFFFD0 0 50FD0008     ST    RF=80FFFF88 S2(0000021E)=00000000
 80FFFFD4 0 50DF0004     ST    RD=00000216 S2(00FFFF8C)=00000000
 80FFFFD8 0 18DF         LR    RD=00000216 RF=80FFFF88
 80FFFFDA 0 4510D066     BAL   R1=00000080 S2(00FFFFEE)=0A23
 80FFFFEE 0 0A23         SVC   I1=23
WTO MSG = HELLO WORLD
 80FFFFF0 0 58DD0004     L     RD=80FFFF88 S2(00FFFF8C)=00000216
 80FFFFF4 0 980CD014     LM    R0=00FFFF80 RC=00000000
S2(0000022A)=00FFFF80
 80FFFFF8 0 58ED000C     L     RE=000002A6 S2(00000222)=000002A6
 80FFFFFC 0 07FE         BR    R2(000002A6)=0A00
 800002A6 0 0A00         SVC   I1=00
Stats total instructions    = 12
Stats Keys                  = 663
Stats Key searches          = 12
Stats Key avg comps         = 1
Stats Key max comps         = 1
Stats total seconds         = 0
Stats instructions/sec      = 705
EZ390 total errors          = 0
EZ390 return code           = 0
```