

Z390 Portable Mainframe Assembler and ZCOBOL Options and File Types

Z390 Options

Option	Scope	Default	Description
@filename	MALE	Options in Z390.OPT file if found.	Retrieve additional options from free form text file with default suffix OPT. Options can be specified delimited by spaces on as many lines as requires. All characters on a line following * are ignored as comments. The @file option can be nested. The default path is the program path. If a file named Z390.OPT exists in the z390 install directory, these options will be applied first as default options.
ALIGN	A	YES	Align DS/DC data fields based on type unless explicit length is specified. If duplication factor is 0, then NOALIGN is ignored.
ALLOW	M	NO	<p>Allow extensions to HLASM syntax including:</p> <ol style="list-style-type: none"> 1) No quotes required for SETC variables. 2) Duplication factor does not require (..). 3) Array declarations with variables in expression accepted 4) Substring starting beyond end returns null string 5) Quoted strings allowed in SETA and SETB expressions 6) Allow &var as null string if not found during substitution 7) Allow duplicate local and global variable declarations 8) Allow AREAD and PUNCH file records greater than 80 characters. <p>The default of NOALLOW insures HLASM compatibility.</p>
AMODE24	LE	NO	Set 390 load module options to start in 24 bit address mode
AMODE31	LE	YES	Set 390 load module options to start in 31 bit address mode

ASCII	AE	NO	Generate ASCII versus EBCDIC DC character constants, compare character strings in ASCII versus EBCDIC in macro processor, generate ASCII versus EBCDIC output for UNPK, ED, and EDMK. Note ASCII mode is not mainframe compatible and requires careful review of program to insure no EBCDIC immediate compare constants such as X'40', 64, X'F0', 240 etc. are used and that there are no assumptions about EBCDIC versus ASCII collating sequences which have numbers and letters reversed, and EBCDIC letters are non-contiguous. Typically HLASM compatible programs are run in default EBCDIC mode and DCB RECFM=FT?VT or some other translation option is used to convert between EBCDIC and ASCII where necessary. Note there are a few instructions affected by ASCII mode:
			1) ED/EDMK generate ASCII versus EBCDIC but the mask must always be in EBCDIC (masks are usually coded in X'...' format)
			2) UNPK generates X'3' versus X'F' in zone field (high order nibble)
ASM	MA	YES	Run az390 assembler as subtask of mz390 passing BAL. Note NOASM is for use in pure text processing programs which only use conditional macro code and AREAD/PUNCH with extensions to process ascii text files. NOASM does not support ordinary symbol attribute tests, OPSYN, or lookahead mode, and requires CHKMAC(0) and CHKSRC(0-2).
ASSIST	MAE	NO	Enable assembly and execution of 11 new ASSIST extended instructions to simplify I/O for student use. This option also sets NOLOADHIGH. See ASSIST Support.
AUTOLINK	L	YES	Search for unresolved external references in SYSOBJ directory list which defaults to linklib.
BAL	M	NO	Generate BAL expanded assembler source file
BS2000	MA	NO	Support Siemens BS2000 assembler global variables
CHKMAC(0-2)	M	0	Check macros during loading as follows:
			0 - no checking
			1- check for missing AGO and AIF labels and issue warning
			2 - also check missing labels and also check for non comment text after MEND

CHKSRC(0-3)	MA	1	Check input source files as follows:
			0 - no checking
			1 - check MLC or BAL input source files for any non-ASCII characters and issue error
			2 - check MLC, MAC, CPY, and BAL input source files for any non-ASCII characters and issue error.
			3 - also check for out of sequence characters in 73-80 or any non-blank characters beyond 80 and issue error.
			Note this should detect any EBCDIC literal character strings containing non-ASCII characters such as binary 0 byte which must be changed to hex X'00' type strings for portability.
CICS	MA	NO	Support EXEC CICS pre-processor expansion and constants. If this option is not on during macro expansion, PROLOG and EPILOG option settings will be ignored.
CODEPAGE(ascii+ebcdic+LIST)	MALE	YES	The default is CODEPAGE(ISO-8859-1+IBM1047). If +LIST is added the mapping of the 2 codepages along with printable character and Unicode values are displayed on ERR file. A hex dump of the tables and a list of the valid ASCII and EBCDIC Unicode Charset codepages are listed. There are 162 Charset codepages available on J2SE 1.6.0_16 including 64 valid ASCII and 15 valid EBCDIC codepages. If the EBCDIC codepage you need is not on the host system, you can replace the EBCDIC codepage name with a file specification such as IBM1047.HCP (example included) which is in hex dump format. The above defaults match z/OS nnd has been successfully tested on Windows Vista and XP.
CON	MALE	YES	Console output for all start/stop, error, trace, MNOTE's with level > 4, and WTO messages. When the TRACE??? option is specified, the CON option is turned off so only start/stop and abort error messages appear on console log. You can specify CON after last TRACE option to turn it back on if you want all display, trace, and error messages displayed on console log. All the trace messages appear on corresponding TR? file for each z390 program executed.
DUMP	E	NO	Generate full memory dump on LOG or TRE if abort
EDF	M	YES	CICS Execution Diagnostic Facility

EPILOG	M	YES	Generate epilog macro call DFHEIEND for CICS program at END statement if CICS option and EPILOG option are on.
ERR(100)	MALE	100	Terminate process if total errors for program exceeds limit. Use ERR(0) to eliminate any error limit and always generate PRN file.
ERRSUM	M	NO	Generate critical error summary on console and ERR file listing missing macros and copybooks. This option requires ASM option and is turned on automatically if missing macros or copybooks are found in the executed macro code path. When ERRSUM is on, ERR(0) is set to prevent abort prior to finding all macro and copybook references. All errors are listed on ERR file along with summary report. Note several iterations may be required to identify and resolve all missing macros and copybooks.
GUAM	E	NO	Support one or more GUI Graphical User Access Method dialogs for MCS, TN3270, or graphics user interface.
INIT	E	YES	Initialize all registers to hex x'F4', all memory to hex x'F5', and all uninitialized load module areas to x'F6' for easier identification of access to uninitialized registers, memory. or program fields. This default option was adopted when ASSIST option was added after fixing a number of regression test initialization bugs. Use NOINIT to revert to all zeros as work-around for bugs. The x'F6' init was added in v1403d PTF.
INSTALL(dir)	MALE	NO	Define alternate z390 install directory to run batch command. The default is set from J2SE Property "user.dir".
IPL(pgm)	E	none	Execute 390 program at startup.
LIST	ALE	YES	Generate PRN, LST, and LOG files for assembler, linker, and execution respectively.
LISTCALL	MA	YES	Generate level macro call and exit comments in BAL file which are used by assembler to format first level macro calls preceding assembler lines with "+" for macro generated source.
LISTUSE	A	YES	List active USING definitions each time USING or DROP changes status in the PRN assembly listing file.

LOADHIGH	E	YES	Load programs and allocate memory for GETMAIN/STORAGE from high end of first FQE large enough to satisfy request. The ASSIST option changes this option to NOLOADHIGH causing initial program to load starting at X'10000' which simplifies relative address calculations.
LOG(file)	MALE	pgm	Set file name for ERR, TR?, and LOG files. The default is the program name. This option is required when running multiple copies of the same program via CMD startup in order to create separate unique ERR, TR?, and LOG files. Otherwise duplicate programs running in parallel will mod the same ERR, TR?, and LOG file.
MAXCALL(50)	M	50	Set limit for nested macro calls.
MAXDISPLAY(80)	M	80	Use to increase zcobol DISPLAY line size up to 256 characters.
MAXESD(1000)	AL	1000	Set limit for ESD sections or entries in one program.
MAXFILE(1000)	M	1000	Maximum macro and copybook files
MAXGBL(1000000)	M	100000	Maximum global macro variables
MAXHEIGHT(600)	E	600	Maximum pixel height for GUI dialog windows
MAXLCL(100000)	M	100000	Maximum local macro variables
MAXLINE(200000)	MA	200000	Maximum MLC, MAC, CPY, and BAL source lines that can be loaded into memory during an assembly.
MAXLOG(1000000)	MALE	1000000	Maximum GUI log file output before truncation begins by removing 50% to limit memory consumption. The only limit on size of log file is MAXSIZE.
MAXPARM(100000)	M	10000	Maximum length of any string during macro processing. This is an extension as HLASM limit is currently 1024.
MAXPASS(2)	A	2	Maximum passes of the BAL source by assembler to resolve forward nested symbol references prior to final pass to generate object code. The number of LOCTR statements is added to this limit in order to resolve the final address of each LOCTR section.
MAXPC(50000)	M	50000	Maximum pseudo code instructions held in cache before LRU replacement begins.

MAXQUE(1000)	E	1000	Maximum output queue length from any CMD started task before queue messages are automatically copied to GUI output log to conserve memory and prevent stall.
MAXRLD(10000)	L	10000	Maximum RLD relocation records allowed in a program.
MAXSIZE(50)	MALE	50	Maximum size of any output file in MB
MAXSYM(50000)	MA	50000	Maximum symbols in one macro assembly
MAXWARN(4)	MA	4	Maximum MNOTE warning level without generating error.
MAXWIDTH(800)	E	800	Maximum pixel width for GUI dialog windows
MCALL	A	NO	List each macro call and exit on the PRN assembly listing in addition to first level calls if LISTCALL option is also on.
MEM(1)	E	1	Memory allocated for 390 program execution in MB. Note for RMODE31 loads and GETMAIN's memory above the 16 MB line must be allocated. For example MEM(32) would provide 16 MB below the line and 16 MB above.
MINHEIGHT(150)	E	150	Minimum pixel height for GUI dialog windows
MINWIDTH(150)	E	150	Minimum pixel width for GUI dialog windows
MNOTE(0)	MA	0	Control MNOTE error/warning messages during macro expansion and assembly: 0 - default generates MNOTE error/warning during mz390 and az390 1 - suppress mz390 error/warning and only pass MNOTE to az390 2 - generates MNOTE error/warning during mz390 and suppresses passing them on to az390
MOD	L	NO	Generate raw code file from lz390 with .MOD suffix and no header or trailer and no RLD's.
OBJ	A	YES	Generate relocatable object code file
OBJHEX	A	NO	Generate ASCII readable form of OBJ file with support for 31 bit long sections. The default is to generate binary OBJ file which is compatible with mainframe linkers but is limited to 24 bit long sections.

PARM('text')	E	none	Define character string parm to passed to executable program via address in R1 at startup pointing to half word length followed by EBCDIC characters. Default length is 0.
PC	M	YES	Generate macro pseudo code in cache memory for speeding up macro code that is executed more than once during macro expansion.
PCOPT	M	YES	Optimize macro pseudo code by looking for push, add/sub, store sequences and replacing them with inc/dec pseudo code opcodes. These pseudo codes are maintained in a cache in memory during macro execution only and have no affect on generated object code.
PRINTALL	A	NO	Suppress PRINT OFF and PRINT NOGEN commands to force all source lines on PRN listing.
PDSMEM8	M	NO	If PDSMEM8 option is specified, error messages will be issued for any macro or copybook file names with length greater than 8. This options is available to provide compatibility with mainframe PDS libraries that are limited to 8 character names. Note zCOBOL and zCICS use names longer than 8 characters such as zcobel\IDENTIFICATION.MAC and cics\CICS_INQUIRE.MAC.
PROFILE(file)	M	NO	Insert COPY file in front of MLC source file
PROLOG	M	YES	Generate CICS prolog macro calls DFHEIGBL, DFHEISTG, and DFHEIENT if CICS option and PROLOG options are on.
PROTECT	E	YES	Prevent modification of PSA in low memory 8k block
REFORMAT	M	NO	Reformat expanded BAL code to align all opcodes at column 10 and operands at column 16 if possible.
REGS	E	NO	Generate GPR register trace before and after each instruction if TRACE option on.
RMODE24	LE	YES	Set 390 load module options to load module below the 24 bit address line.
RMODE31	LE	NO	Set 390 load module options to load module above the 31 bit address line. Note this options requires option MEM be set to include enough memory above the 31 bit address line.

STATS or STATS(file)	MALE	NO	Generate statistics file STA with final options listing, macro and copybook file listing, plus all program statistics. If file is omitted the program path and file name are used with STA suffix.
SYS390(dir)	LE	pgm dir	Define directory for storing and retrieving 390 load modules
SYSBAL(dir)	MA	pgm dir	Define directory for storing and retrieving generated BAL files. This option may also override file name and/or suffix (*.sfx overrides just the suffix in the pgm directory).
SYSCPY(dir)	MA	pgm dir	Define one or more directories for retrieving CPY source input files. If the option starts with + the directories listed will be concatenated with current list. Multiple directories are always separated by +. This option may also override suffix by adding *.sfx.
SYSDAT(dir)	M	pgm dir	Define directory for retrieving DAT source input files for AREAD. Set to SYSDAT(.) for current directory when no path is specified on DSNAME parm for AREAD.
SYSERR(dir)	MALE	pgm dir	Define directory for output ERR files. This option may also override file name and/or suffix (*.sfx overrides just the suffix in the pgm directory).
SYSLKD(file)	L	pgm dir	Define input source file for linker commands such as INCLUDE, ALIAS, ENTRY, and NAME.
SYSLOG(dir)	E	pgm dir	Define directory for output LOG file. This option may also override file name and/or suffix (*.sfx overrides just the suffix in the pgm directory).
SYSLST(dir)	AL	pgm dir	Define directory for output LST files. This option may also override file name and/or suffix (*.sfx overrides just the suffix in the pgm directory).
SYSMAC(dir)	M	mac dir	Define one or more directories for source MAC files. If the option starts with + the directories listed will be concatenated with current list. Multiple directories are always separated by +. This option may also override suffix by adding *.sfx.
SYSMLC(dir)	MA	pgm dir	Define directory for source input MLC files.
SYSOBJ(dir)	AL	pgm dir	Define directory for OBJ relocatable object files
SYSOPT(dir)	MALE	pgm dir	Define directory for @file option files. This option may override suffix using *.sfx.
SYSPARM('text')	M	none	Define text string which can be accessed by mz390 global macro variable &SYSPARM.

SYSPCH(dir)	M	pgm dir	Define directory for PCH output files from PUNCH. This option may also override file name and/or suffix (*.sfx overrides just the suffix in the pgm directory). Note PUNCH output is in ASCII source format and is not directed to the OBJ file. These files may be used as linker input commands via SYSLKD(file) option. Set to SYSPCH(.) for current directory when no path is specified on DSNAME parm for PUNCH.
SYSPRN(dir)	A	pgm dir	Define directory for assembler listing PRN output files. This option may also override file name and suffix. This option may also override file name and/or suffix (*.sfx overrides just the suffix in the pgm directory).
SYSTEM(file)	MALE	pgm.ERR	Define alternate file for all start/end messages plus any error messages, plus related source file statistics for each file containing errors. The default is the program name with ERR suffix.
SYSTRC(dir)	MALE	pgm dir	Define directory for output TRACE TR? files
TEST	E	NO	Start execution with interactive TEST mode active which prompts for commands such as T opcode or G opcode to trace or execution to the next occurrence of opcode name.
TESTDD(file)	E	NO	Define input command file to run TEST commands. See regression test rt\test\TESTTST1.MLC for example.
THREAD	A	NO	Assign continuing CSECT addresses for multiple CSECT's assembled in the same module to help identify location of instruction and data labels.
TIME or TIME(seconds)	MALE	YES	Limit execution time of each program to the number of seconds specified. The default is 15 seconds. This option is turned off if NOTIMING specified. Use NOTIME to allow tasks such as SOA servers like z390CICS to run continuously while still supporting application use of timing functions.

TIMING	MALE	YES	Calculate elapsed time and instructions per second statistics for display on START/END messages and STA statistics. tTIMING also displays current z390 and J2SE version on start message and memory usage on ended message. Note NOTIMING is used in regression tests to force generated files to be identical by using fixed data/time stamp using GregorianCalendar(2005,0,2,22,33,44). NOTIMING suppresses versions on start message and memory usage on ended message. NOTIMING also sets NOTIME which prevents application from using time functions..
TRACE or TRACE(AEGILMPQTV)	E	NO	TRACE or TRACE(E) will turn on ez390 execution trace generation on TRE file and turn off CON. TRACE will display every instruction executed along with address and value of each operand. Any combination of the following trace options can be set using the TRACE(...) option. TRACE(*) will also set TRACEALL. Note the &SYSTRACE global SETC variable can be used to turn any trace options on or off during execution.
TRACEA or TRACE(A)	A	NO	TRACEA or TRACE(A) will turn on az390 assembly trace generation on TRA file and turn off CON. TRACEA will display each BAL statement during each pass of the assembler.
TRACEALL	MALE	NO	Turn on all trace options generating TRM, TRA, TRL, and TRE files and turn off CON.
TRACEC or TRACE(C)	M	NO	Trace copy file code for TRACEM and TRACEP. The default is NOTRACEC to eliminated redundant code that usually just contains global variable declarations.
TRACEG or TRACE(G)	E	NO	Turn on ez390 emulator trace generation on TRE file with trace of GETMAIN/FREEMAIN FQE memory management control block changes and turn off CON.
TRACEI or TRACE(I)	M	NO	TRACEI will trace each AINSERT showing source on the TRM trace file.
TRACEL or TRACE(L)	L	NO	TRACEL or TRACE(L) will turn on lz390 linker trace generation on TRL file and turn off CON. TRACEL shows each CSECT, ENTRY, EXTRN and OBJ file being loaded.

TRACEM or TRACE(M)	M	NO	TRACEM or TRACE(M) will turn on mz390 macro processor trace generation on TRM file and turn off CON. TRACEM displays each conditional macro statement executed plus stored values and AIF compare values.
TRACEP or TRACE(P)	M	NO	TRACEP or TRACE(P) will turn on mz390 macro processor pseudo code generation and execution trace on TRM file and turn off CON. TRACEP shows each conditional macro pseudo operation performed for each conditional macro statement and the values of all variables.
TRACEQ or TRACE(Q)	E	NO	TRACEQ or TRACE(Q) will turn on ez390 emulator QSAM/BSAM DCB I/O trace generation on TRE file and turn off CON.
TRACES or TRACE(S)	M	NO	Displace MLC source lines on console regardless of CON setting to help locate any loops, waits, or hangs in large macro process. If MCALL is also on, then *MCALL and *MEXIT comments will also be displayed to show location with nested macros.
TRACET or TRACE(T)	E	NO	TRACET or TRACE(T) will turn on ez390 emulator TGET/TPUT and TCP/IO trace generation on TRE file and turn off CON.
TRACEV or TRACE(V)	E	NO	TRACEV or TRACE(V) will turn on ez390 emulator VSAM ACB/RPL I/O trace generation on TRE file and turn off CON.
TRAP	E	YES	Trap any unexpected J2SE program exceptions and generate emulator OC5 exception which can be handled by SPIE/STAE exits if defined. NOTRAP can be used with Eclipse source debugger to stop at J2SE statement causing exception.
TS	MALE	NO	Generate JDBC compatible time-stamp on all TRACE and ERR file records for use in debugging time dependent issue.
VCB	E	YES	Generate VSAM cache buffer for improving VSAM performance by storing most recently accessed records and indexes.
XREF	A	YES	Cross reference symbols to source lines in PRN assembly listing.

ZSTRMAC	M	YES	Support expansion of ZSTRMAC structured conditional macro code instructions during loading of MLC, MAC, and CPY files. Note z390 macros in z390\mac directory now are using zstrmac for structured coding. NOZSTRMAC can only be used with MVS, VSE, or other non-structured macro libraries.
ZVSAM	E	0	ZVSAM 0=no vsam, 1=zvsam1 support, 2=zvsam2 support