



With a Little Help from My Friends: How HLASM Interacts with LE, Binder, C, CICS and MQ

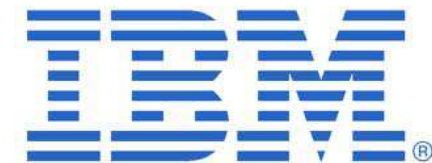
Barry Lichtenstein and Sharuff Morsa
IBM

smorsa@uk.ibm.com

barryl@us.ibm.com

Thursday, August 15, 2013
Session Number **14084**

Insert
Custom
Session
QR if
Desired.



HLASM and friends: Introduction

Assembler application programming with

Binder

C

CICS

MQ

Language Environment

HLASM and friends: C (or is this LE?)

Inter language communication (ILC) for ‘small, simple assembler module’.....

*C provides the **#pragma linkage** directive*

It identifies the entry point of modules that are used in interlanguage calls from C programs

#pragma linkage(*PROG1*, OS)

Designates *PROG1* as an OS linkage entry point.

If the C program is compiled with NOXPLINK, on entry to the called routine, R13 points to a standard Language Environment stack frame, beginning with a 72-byte save area.

HLASM and friends: C ...

Lets start with a simple C program as an example.
Our C program: **CLOCKC** invoking **GETSTCK**, our
assembler routine

```
#pragma linkage(GETSTCK,OS)
```

GETSTCK:

- returns the current *STCK* or *STCKF* or *STCKE* value
- Does not have any working storage

HLASM and friends: C ...

```
// define parms passed to GETSTCK
typedef struct tagGETSTCK_INPUT_PARMS // input parms structure
{
char eyecatcher??(4??); // eye catcher c structure
+char eyecatcherY4"; // eye catcher
long version; // version
long function; // request
long returnValue??(3??); // returned value
+long returnValueY3"; // returned value
} GETSTCK_INPUT_PARMS;

#define Return_STCK 1
#define Return_STCKF 2
#define Return_STCKE 3
```

```
2075 drop r12
2076 GETSTCK_INPUT_PARMS DSECT input parms structure HLASM dsect
2077 eyeCatcher DC CL4'GETS' eye catcher
2078 version DC F'1' input parm version
2079 function DC F'0' request
2080 returnValue DC 3F'0' returned value
2081 returnValue_L equ *-returnValue returned value length
2082 END
ibol and Literal Cross Reference Page 5
.m Program Defn References HLASM R6.0 2013/08/10 20.09
2002 1000R
```

HLASM and friends: C ...

```

*...+....1....+....2....+....3....+....4....+....5....+....6....+...
/* copyright IBM UK LTD 2013 */
/* sample c program to invoke assembler routine */
/* the assembler routine does not require any working storage*/

#include <stdio.h>
#include <leawi.h>
#include <stdlib.h>
#include <string.h>
#include <ceedcct.h>

#pragma linkage(GETSTCK,OS) // use OS linkage

// define CEEDUMP options
#define CEEDUMP_OPT "THREAD(CURRENT) TRACEBACK FILES"

// define parms passed to GETSTCK
typedef struct tagGETSTCK_INPUT_PARMS // input parms structure
{
char eyecatcher??(4??); // eye catcher
+char eyecatcherY4; // eye catcher
long version; // version
long function; // request
long returnValue??(3??); // returned value
+long returnValueY3; // returned value
} GETSTCK_INPUT_PARMS;

#define Return_STCK 1
#define Return_STCKF 2
#define Return_STCKE 3

// prototype GETSTCK
int GETSTCK(GETSTCK_INPUT_PARMS *);

int main() // main c routine

```

HLASM and friends: C ...

```

clockc_listing_and_run.txt - Notepad
File Edit Format View Help
46
47 // say we starting
48 5 printf("CLOCKC started \n");
49
15694A01 V1.13 z/OS XL C 'SMORSA.BOSTON.ASM.SOURCE(CLOCKC)'

                                * * * * *   S O U R C E   * * * * *

LINE  STMT
50      // initialise our parameter block
51      // this is the area we pass to GETSTCK
52      6 memcpy(&parms.eyecatcher,"GETS",4);
52      6 + __memcpy(&parms.eyecatcher,"GETS",4);
53      7 parms.version = 1; // version 1 of interface
54      8 parms.function = Return_STCKF; // return STCKF value
54      8 + parms.function = 2; // return STCKF value
55
56      // invoke GETSTCK
57      9 rc = GETSTCK(&parms);
58
59      // dbebugging - use CEEDUMP to dump our storage area
60      10 printf("GETSTCK_INPUT_PARMS located at %p \n",&parms);
61      11 CEE3DMP(title,options,&fc);
62      printf("returnedvalue %8.8x %8.8x %8.8x \n",
63      parms.returnValue??(0??),
63      + parms.returnValue?0,
64      parms.returnValue??(1??),
64      parms.returnValue?d

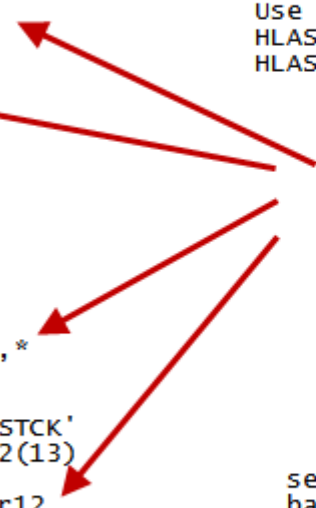
```

HLASM and friends: C ...

```

:      STMT  Source statement                                HLASM R6.0  2013/0
1      /** copyright IBM 2013
2      /** GETSTCK - Sample program to return STCK, STCKF or STCKE
3          SYSSTATE ARCHLVL=2                               Arch level
4+*     THE VALUE OF SYSSTATE IS NOW SET TO ASCENV=P AMODE64
+       L=2 OSREL=00000000
5       IEABRCX DEFINE                                     Use relative branch
324     IEABRCX ENABLE                                     Use relative branch
349     ASMMREL ON                                         HLASM Structured Macros
350     COPY ASMMSP                                         HLASM Structured Macros
.O 1957 GETSTCK CSECT
1958 GETSTCK AMODE 31
1959 Return_STCK EQU 1
1960 Return_STCKF EQU 2
1961 Return_STCKE EQU 3
1962 Return_ERROR EQU 16
1963 r01 EQU 1,,,,GR32
1964 r12 EQU 12,,,,GR32
1965 r14 EQU 14,,,,GR32
1966 r15 EQU 15,,,,GR32
1967     save (14,12),,*
XC 1969+     J      *+12                                     BRANCH AROUND
1970+     DC     AL1(7)
1971+     DC     CL7'GETSTCK'                               IDENTIFIER
XC 1972+     STM   14,12,12(13)                             SAVE REGISTER
1973     lr     r12,r15                                     set base reg
1974     using GETSTCK,r12                                  base reg
.O 1975     L     r01,0(r01)                                  address input parms
1976     using GETSTCK_INPUT_PARMS,r01                    input parms
.O 1977     LHI  r15,Return_ERROR                          preset rc
1978     IF (c1c,eyeCatcher,NE,=C'GETS')                  validate field

```



HLASM and friends: C ...

```

nd ==>
1      L      r01,0(,r01)          address input parms
2      using GETSTCK_INPUT_PARMS,r01  input parms
3      LHI   r15,Return_ERROR      preset rc
4      IF (clc,eyeCatcher,NE,=C'GETS') validate field
5          return (14,12),RC={15}
6      ENDIF
7      SELECT clc,function,EQ      what was requested?
8          WHEN (=A(Return_STCK))
9              XC  returnValue(returnValue_L),returnValue
10             STCK returnValue
11             LHI  r15,Return_STCK
12             WHEN (=A(Return_STCKF))
13                 XC  returnValue(returnValue_L),returnValue
14                 STCKF returnValue
15                 LHI  r15,Return_STCKF
16                 WHEN (=A(Return_STCKE))
17                     XC  returnValue(returnValue_L),returnValue
18                     STCKE returnValue
19                     LHI  r15,Return_STCK
20                 OTHERWISE
21             ENDSEL
22             return (14,12),RC={15}

```

HLASM and friends: C ...

```

2002+#@LB1    DC      0H                                01-000094
2004          SELECT  CLC,function,EQ                what was requested?
2011          WHEN  (=A(Return_STCK))
2017+        CLC   function,=A(Return_STCK)          01-0000152
2018+        BRC   7, #@LB4                          01-0000153
2020          XC   returnValue(returnValue_L),returnValue
2021          STCK returnValue
2022          LHI  r15, Return_STCK
2023          WHEN  (=A(Return_STCKF))
2027+        BRC   15, #@LB3                          SKIP TO END          01-0000150
2028+#@LB4    DC      0H                                01-0000150
2031+        CLC   function,=A(Return_STCKF)          01-0000152
2032+        BRC   7, #@LB6                          01-0000153
2034          XC   returnValue(returnValue_L),returnValue
2035          STCKF returnValue
2036          LHI  r15, Return_STCKF
2037          WHEN  (=A(Return_STCKE))
2041+        BRC   15, #@LB3                          SKIP TO END          01-0000150
                                                    Page      4
TSTCK, R12
  Stmt Source Statement                                HLASM R6.0  2013/08/10 20.09
2042+#@LB6    DC      0H                                01-0000150
2045+        CLC   function,=A(Return_STCKE)          01-0000152
2046+        BRC   7, #@LB8                          01-0000153
2048          XC   returnValue(returnValue_L),returnValue
2049          STCKE returnValue
2050          LHI  r15, Return_STCK
2051          OTHERWISE

```

HLASM and friends: C ...

```

clockc_listing_and_run.txt - Notepad
File Edit Format View Help
./Z/OS V1 R13 BINDER      20:09:43 SATURDAY AUGUST 10, 2013
BATCH EMULATOR  JOB(GACUC  ) STEP(LKED  ) PGM= IEWBLINK
IEW2278I B352 INVOCATION PARAMETERS - XREF,LIST,MAP
IEW2322I 1220  1      INCLUDE  SYSLIB(CLOCKC)
IEW2322I 1220  2      INCLUDE  SYSLIB(GETSTCK)
IEW2322I 1220  3      NAME    CLOCKC(R)

          *** M O D U L E M A P ***

-----
CLASS  C_CODE          LENGTH =      384  ATTRIBUTES = CAT,    LOAD, RMODE=ANY
          OFFSET =      0  IN SEGMENT 001                ALIGN = DBLWORD
-----

SECTION  CLASS  NAME          TYPE      LENGTH  DDNAME  SOURCE  MEMBER
  OFFSET  OFFSET
         0      0  CLOCKC#C          CSECT      308  SYSLIB   01  CLOCKC
         0      0      CLOCKC#C          LABEL
        88     88      main              LABEL

         0     308  CEESTART          CSECT       7C  SYSLIB   01  CLOCKC
         0     308  CEESTART          LABEL

-----
CLASS  C_DATA          LENGTH =      1C  ATTRIBUTES = CAT,    LOAD, RMODE=ANY
          OFFSET =    388  IN SEGMENT 001                ALIGN = DBLWORD
-----

```

HLASM and friends: C ...

```

0      0      IEWBLIT      CSECT      LABEL      EV      NOEL
*** M O D U L E M A P ***

-----
CLASS B_TEXT          LENGTH =      9F2  ATTRIBUTES = CAT,      LOAD, RMODE= 2
                      OFFSET =      0  IN SEGMENT 002      ALIGN = DBLWORE
-----
SECTION  CLASS  NAME      TYPE      LENGTH  DDNAME  SOURCE  MEMBER
OFFSET  OFFSET
0      0      GETSTCK      CSECT      A0      SYSLIB  01      GETSTCK
0      A0      CEERootA      * CSECT      1F0     SYSLIB  02      CEERootA
      A0      CEERootD      LABEL
290    CEEBETBL      * CSECT      28      SYSLIB  02      CEEBETBL
2B8    EDCINPL      * CSECT      24      SYSLIB  02      EDCINPL
2E0    CEE3DMP      * CSECT      14      SYSLIB  02      CEE3DMP
2F8    CEESG003     * CSECT      12B     SYSLIB  02      CEESG003
428    CEEBPUBT     * CSECT      70      SYSLIB  02      CEEBPUBT

```

HLASM and friends: C ...

```

=====JOB OUTPUT=====
CLOCKC Started
GETSTCK_INPUT_PARMS located at 20117450
returnedValue cbcadc19 4c591680 00000000
CLOCKC ended rc(2)
1CEE3DMP V1 R13.0: CLOCKC CEEDUMP - storage -diagnostics
ASID: 002B Job ID: JOB20890 Job name: GACUC Step name: RUN US

CEE3845I CEEDUMP Processing started.
CEE3DMP called by program unit (entry point main) at offset +000000FE.

Registers on Entry to CEE3DMP:

PM..... 0100
GPR0..... 00000000_20117440 GPR1..... 00000000_201172E0 GPR2..... 00000000
GPR4..... 00000000_201172F0 GPR5..... 00000000_00007D60 GPR6..... 00000000
GPR8..... 00000000_00000030 GPR9..... 00000000_80000000 GPR10.... 00000000
GPR12.... 00000000_20111BD0 GPR13.... 00000000_20117248 GPR14.... *****
FPR0..... 26100000 00000000 FPR2..... 18000000 00000000
FPR4..... 00000000 00000000 FPR6..... 00000000 00000000

GPREG STORAGE:
Storage around GPR0 (20117440)
-0020 20117420 40404040 40404040 40404040 40404040 40404040 40404040
+0000 20117440 00000000 00000000 00000000 00000000 C7C5E3E2 00000000
+0020 20117460 4C591680 00000000 00000002 00000000 00000000 20117340
Storage around GPR1 (201172E0)
-0020 201172C0 00000000 00000000 00000000 00000000 00000000 00000000

```

HLASM and friends: C ...

```

GPREG STORAGE:
Storage around GPR0 (20117440)
-0020 20117420 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404000
+0000 20117440 00000000 00000000 00000000 00000000 C7C5E3E2 00000001 00000002 CBCADC19
+0020 20117460 4c591680 00000000 00000002 00000000 00000000 20117340 00000000 00000000
Storage around GPR1 (201172E0)
-0020 201172C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+0000 201172E0 201172F0 20117340 20117440 00000000 C3D3D6C3 D2C340C3 C5C5C4E4 D4D74060
+0020 20117300 40A2A396 99818785 40608489 81879596 89A2A389 83A24000 00000000 00000000
Storage around GPR2 (20117340)
-0020 20117320 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+0000 20117340 E3C8D9C5 C1C44DC3 E4D9D9C5 D5E35D40 E3D9C1C3 C5C2C1C3 D240C6C9 D3C5E240
+0020 20117360 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040

```

```

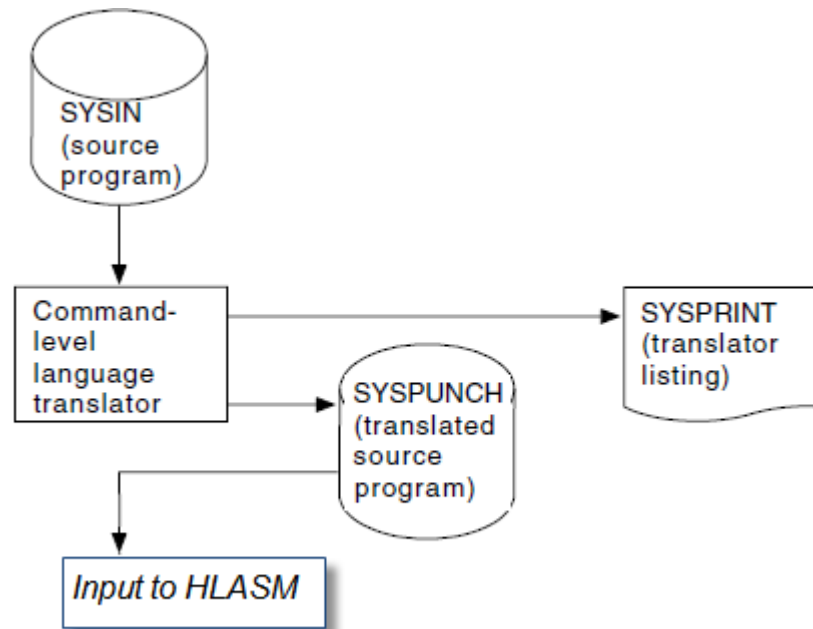
2075          drop r12
2076 GETSTCK_INPUT_PARMS DSECT      input parms structure
2077 eyeCatcher      DC      CL4'GETS'      eye catcher
2078 version         DC      F'1'          input parm version
2079 function        DC      F'0'          request
2080 returnValue     DC      3F'0'         returned value
2081 returnValue_L  equ *-returnValue     returned value length
2082          END

```

bol and Literal Cross Reference Page 5
 m Program Defn References HLASM R6.0 2013/08/10 20.09

HLASM and friends: CICS

CICS assembler programs need to be translated (no integrated translator)



HLASM and friends: CICS ...

What does the translator do for me?

converts EXEC CICS commands into assembler macro invocations

inserts macro definitions for working storage

converts return code checking

Assembler arguments passed in CICS commands

Generally, an argument can be either

the address of the data or

the data itself

ie either a relocatable expression or an absolute expression

HLASM and friends: CICS ...

```
File Edit Format View Help
.CICS X.X.X COMMAND LANGUAGE TRANSLATOR
OPTIONS USED ARE:-

CICS
SPIE
EDF
LINECOUNT(60)
TABLE(DFHEITAB)
NATLANG(EN)
OPTIONS
NOPROLOG
NOEPILOG
NOSYSEIB
NOFEPI
NOCPM
LENGTH
NO MESSAGES PRODUCED BY TRANSLATOR
TRANSLATION TIME:- 0.00 MINS.
```

HLASM and friends: CICS ...

Translator inserts

DFHEIGBL

This macro sets globals

DFHEIENT

This macro is inserted after the first CSECT. It performs prolog code:

- Saves registers

- Gets an initial allocation of the storage defined by DFHEISTG)

- Sets up a base register (default register 3)

- Sets up a dynamic storage register (default register 13)

- Sets up a register to address the EIB (default register 11)

HLASM and friends: CICS ...

Translator inserts

DFHEIRET

This macro performs epilog code:

- Restores registers

- Allows a RC to be passed back to the caller

- Returns control to the address in register 14

DFHEISTG and DFHEIEND

These macros define dynamic storage:

- Define the storage required for the parameter list

- Define a save area

DFHEIBLK

Copybook containing a DSECT that describes the EIB

HLASM and friends: CICS ...

...and the translator converts...

...assembler arguments: examples

```
CLC EIBRESP,DFHRESP(NOSPACE)
```

...becomes

```
CLC EIBRESP,=F'18'
```

HLASM and friends: CICS ...

...and the translator converts...

*...a simple CICS command **EXEC CICS RETURN** to a macro invocation: DFHECALL ...*

```
1741 *      NEEDS TO BROWSE FORWARD OR BACK FROM THE CURRENT POSITION IN
1742 *      THE FILE. THE "COMMAREA" IS PASSED TO THE NEXT INVOCATION OF
1743 *      THIS PROGRAM.
1744 *
1745 *      EXEC CICS RETURN TRANSID(EIBTRNID)
1746 *              COMMAREA(COMMAREA)
1747 *              LENGTH(13)
1748 *      DFHECALL =X'0E08E0002800001000', (CHA4, EIBTRNID), (____RF, COMM
              AREA), (FB_2, =Y(13))
1750+*****
```

HLASM and friends: CICS ...

...which get assembled as...

```

L750+*****
L751+      DS      OH
L752+      LA      1,DFHEIPL
L753+      LA      14,=X'0E08E0002800001000'
L754+      LA      15,EIBTRNID
L755+      LA      0,COMMAREA
L756+      STM     14,0,0(1)
L757+      LA      14,=Y(13)
L758+      ST      14,12(,1)
L759+      OI      12(1),X'80'          LAST ARGUMENT
L760+      LLGT    15,=V(DFHEG1)          @R3141
L761+      BASR    14,15                  @R3141
L762+*****

```

... CICS macros can be found in cicshlq.SDFHMAC

HLASM and friends: CICS ...

```
Command ==> _____ Scroll ==> C
***** ***** Top of Data *****
000001      MACRO                                000001
000002 &LABEL  DFHECALL                          000002
000003 .*                                         000003
000004 .****** 000004
000005 .*                                         * 000005
000006 .* £MAC(DFHECALL),COMP(COMMAND),PROD(CICS TS ); * 000006
000007 .*                                         * 000007
000008 .* DESCRIPTIVE NAME = CICS/MVS EXEC Interface call macro * 000008
000009 .*                                         * 000009
000015 .*                                         * 000015
000016 .*                                         * 000016
000017 .*      Licensed Materials - Property of IBM * 000017
000018 .*                                         * 000018
000019 .*      "Restricted Materials of IBM" * 000019
000020 .*                                         * 000020
000021 .*      5655-Y04 * 000021
000022 .*                                         * 000022
000023 .*      (C) Copyright IBM Corp. 1982, 2012" * 000023
000024 .*                                         * 000024
```

CICS
Green
Screen

HLASM and friends: CICS ...

...the 'glue code' ...

... load register 15

```
LLGT 15,=V(DFHEG1)
```

... DFHEG1 is resolved by the inclusion of the DFHEAG stub routine...

```
INCLUDE SYSLIB(DFHEAG)
```

...which is later resolved by the Binder so that at execution time we branch to it...

```
BASR 14,15
```


HLASM and friends: CICS ...

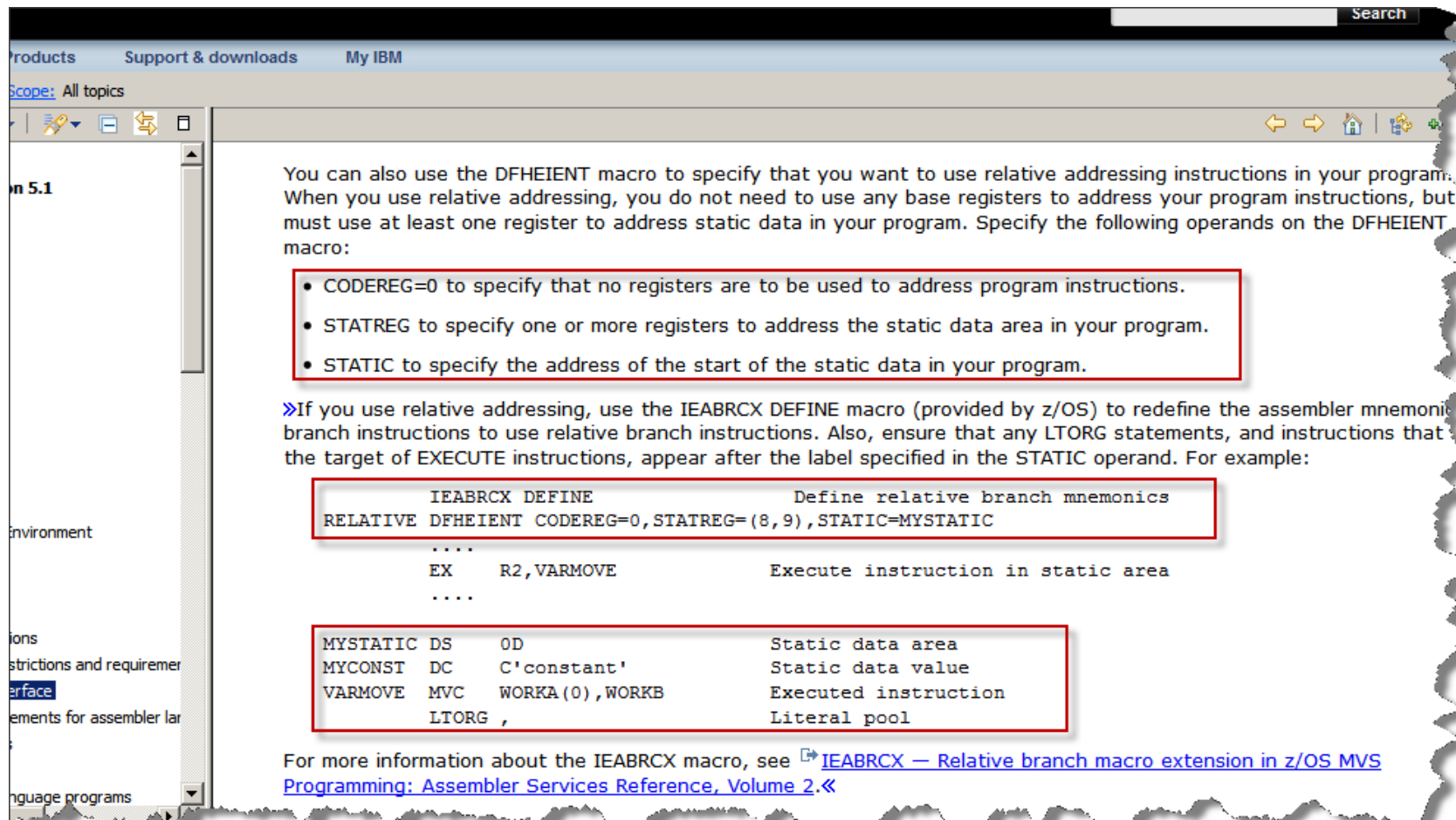
```

382 *****
383      DFHEIGBL ,,,NOLE          INSERTED BY TRANSLATOR
384+*,&DFHEIDL SETB 0 1 MEANS EXEC DLI IN PROGRAM
385+*,&DFHEIDB SETB 0 1 MEANS BATCH PROGRAM
386+*,&DFHEIRS SETB 0 1 MEANS RSECT
387+*,&DFHEILE SETB 0 1 MEANS LE MAIN
388      DFHEISTG DSECT
390+DFHEISTG DSECT          EXEC Interface Storage      @R31410A
391+      USING *,DFHEIPLR      Establish Addressability @R31410A
392+DFHEISA DS 18FD          F4SA                        @R31410A
393+DFHEIBP DS AD           EIB pointer                 @R31410A
394+DFHEICAP DS AD          COMMAREA pointer           @R31410A
395+DFHEIR13 DS AD          Register 13                 @R31410A 0
396+DFHEIVER DS H           Version                    @R31410A 0
397+DFHEIV00 DS H           Used by DFHECALL           @R31410A 0
398+DFHEIRS3 DS F           Used by DFHECALL           @R31410A 0
399+DFHEIPL DS 64A          Parameter list              @R31410A 0
400+DFHEITP1 DS A           Used by DFHECALL           @R31410A 0
401+DFHEITP2 DS A           Used by DFHECALL           @R31410A 0
402+DFHEITP3 DS A           Used by DFHECALL           @R31410A 0
403+DFHEITP4 DS A           Used by DFHECALL           @R31410A 0
404+DFHEITG1 DS AD          Used by DFHECALL           @R31410A 0
405+DFHEITG2 DS AD          Used by DFHECALL           @R31410A 0
406+DFHEITG3 DS AD          Used by DFHECALL           @R31410A 0
- ASSEMBLER
                                           Page 5

tmt Source Statement          HLASM R6.0 2013/08/10 20.09
407+DFHEITG4 DS AD          Used by DFHECALL           @R31410A
408+      DS 4FD             Reserved                  @R31410A
409+*****

```

HLASM and friends: CICS ...



products Support & downloads My IBM

Scope: All topics

You can also use the DFHEIENT macro to specify that you want to use relative addressing instructions in your program. When you use relative addressing, you do not need to use any base registers to address your program instructions, but must use at least one register to address static data in your program. Specify the following operands on the DFHEIENT macro:

- CODEREG=0 to specify that no registers are to be used to address program instructions.
- STATREG to specify one or more registers to address the static data area in your program.
- STATIC to specify the address of the start of the static data in your program.

»If you use relative addressing, use the IEABRCX DEFINE macro (provided by z/OS) to redefine the assembler mnemonic branch instructions to use relative branch instructions. Also, ensure that any LTORG statements, and instructions that the target of EXECUTE instructions, appear after the label specified in the STATIC operand. For example:

```

IEABRCX DEFINE          Define relative branch mnemonics
RELATIVE DFHEIENT CODEREG=0, STATREG=(8,9), STATIC=MYSTATIC
....
EX    R2,VARMOVE      Execute instruction in static area
....

MYSTATIC DS    0D          Static data area
MYCONST DC    C'constant'  Static data value
VARMOVE MVC    WORKA(0),WORKB Executed instruction
LTORG ,              Literal pool
  
```

For more information about the IEABRCX macro, see [IEABRCX — Relative branch macro extension in z/OS MVS Programming: Assembler Services Reference, Volume 2](#).

HLASM and friends: CICS ...

```

Stmt  Source Statement                                HLASM R6.0  2013/08/10 20.09
1166 DFHfABRW RMODE 31
1167          DFHEIENT DATAREG=13, EIBREG=11, STATREG=3, STATIC=STATR, CODEREG=0
1258+*****
1259+*
1260+* CONTROL BLOCK NAME = DFHEIBLK
1261+*
1262+* NAME OF MATCHING PL/AS CONTROL BLOCK = None
1263+*
1264+* DESCRIPTIVE NAME = CICS TS EXEC Interface Block.

```

```

1474+*****
1475+*          SETUP STATIC STORAGE ADDRESSABILITY
1476+*****
1477+          LARL 3, STATR          Load static storage base          @D3C
1478+          USING STATR, 3          @D2A
1479+*****
1480+*          ESTABLISH DATA ADDRESSIBILITY
1481+*****
1482+ DFHEIPLR EQU 13          PARAMETER LIST REGISTER          @BBAC81A
1483+          LGR DFHEIPLR, 1          @R31410A
1484+          USING DFHEISTG, 13          @BBAC81A
1485+*****
1486+*          ESTABLISH EIB ADDRESSIBILITY
1487+*****
1488+          LG DFHEIBR, DFHEIBP          @R31410A
1489+          USING DFHEIBLK, DFHEIBR          @BBAC81A
1490+*****
1491+*          END OF PROLOG CODE FOR EXEC INTERFACE
1492+*****
E - ASSEMBLER

```

HLASM and friends: CICS ...

```

2261 *
DFHfABRW - CICS SAMPLE FILEA BROWSE - ASSEMBLER
Active Usings: STATR,R3 DFHEIBLK,R11 DFHEISTG,R13
Loc Object Code Addr1 Addr2 Stmt Source Statement HLASM R6.0 201
00000698 2262 STATR DS 0D
00000698 FFFFFFFFFFFFFFFF 2263 CHEKTAB DC 256X'FF'
00000798 00000798 00000788 2264 ORG CHEKTAB+X'F0'
00000788 0000000000000000 2265 DC 10X'00'
00000792 00000792 00000798 2266 ORG
00000798 2267 LTORG
00000798 C1C3C3D6E4D5E340 2268 =CL(L'MESSAGES)'ACCOUNT NUMBER MUST BE NUME
000007E8 C5D5C440D6C640C6 2269 =CL(L'MESSAGES)'END OF FILE - PLEASE RESTAR
00000838 D7D9C5E2E240C3D3 2270 =CL(L'MESSAGES)'PRESS CLEAR TO EXIT'
00000888 E3D9C1D5E2C1C3E3 2271 =CL(L'MESSAGES)'TRANSACTION TERMINATED'
000008D8 C6C9D3C5C1404040 2272 =CL8'FILEA'
000008E0 00000000 2273 =V(DFHEGL)
000008E4 00000024 2274 =F'36'
000008E8 00000000 2275 =F'0'
000008EC 00000000 2276 =A(0)
000008F0 0000000D 2277 =F'13'
000008F4 00000014 2278 =F'20'
000008F8 C5D9D9E2 2279 =CL4'ERRS'
000008FC 0000 2280 =H'0'
000008FE F9F9F9F9F9F9 2281 =C'999999'
00000904 FFFFFFFFFFFFFFFF 2282 =6X'FF'
0000090A F0F0F0F0F0F0 2283 =C'000000'
00000910 000D 2284 =Y(13)
00000912 0050 2285 =Y(L'FILEA)
00000914 1802C00028000000 2286 =X'1802C0002800000000040900000020'
00000923 C4C6C85BC1C7C1 2287 =CL7'DFHfAGA'
0000092A 060CB000280020A0 2288 =X'060CB000280020A000'

```

HLASM and friends: CICS ...

```

Command ==> _____ Scroll ==> C
***** 0001
*      If CODEREG value specified check that relative addressing  * 0001
*      is specified.                                           * 0001
***** 0001
      AIF  ('&CODEREG' EQ '').AM64RA                            @R31410A 0001
      AIF  ('&CODEREG' EQ '0' OR                               *0001
           '&CODEREG' EQ '*').AM64RA                            @R31410A 0001
      MNOTE 12,'DFHEIENT - AMODE 64 - MUST USE RELATIVE ADDRESSING.' 0001
*                                                                 @R31410A 0001
AM64RA ANOP ,                                                @R31410A 0001
***** 0001
*      Check that static specified.                             * 0001
***** 0001
      AIF  ('&STATIC' NE '').AM64ST2                          @R31410A 0001
      MNOTE 12,'DFHEIENT - AMODE 64 - STATIC REQUIRED.'          @R31410A 0001
AM64ST2 ANOP ,                                                @R31410A 0001
***** 0001
      Check that CMCSTATE AMODE 64 MFC specified.             * 0001

```

(another)
CICS
Green
Screen

HLASM and friends: CICS ...

```

000028          00000028      1370+DFH0008B EQU *          End of eyecatcher
000028          1461+DFH0008C DS 0H
000028          1462+          DS 0S((DFHEISTG+65264-DFHEIEND-4096)/4096) L
000028          1463+*
000028 EBEC D008 0024          00000008      1464+          STMG 14,12,8(13)      Save callers registers
00002E 0700          1465+          CNOP 0,4          Force word alignment
000030 A715 0007          0000003E      1466+          JAS 1,DFH0008D      Setup base and skip co
000034 0000042E          1467+*
000038 00000000          1468+          DC AL4(DFHEIEND-DFHEISTG) Length of storage
00003C 0000          1469+          DC V(DFHEAG0)      Entry point
00003E          1470+          DC AL2(0)          Version
00003F          1471+DFH0008D DS 0H
00003E E3F0 1004 0017          00000004      1472+          LLGT 15,4(,1)      Load entry point
000044 0DEF          1473+          BASR 14,15        Call DFHEAG0
000044          1474+*****

```

V-type symbols: DFHEAG0 and DFHEG1

```

00838 D7D9C5E2E240C3D3          2270          =CL(L'MESSAGES)'PRESS CL
00888 E3D9C1D5E2C1C3E3          2271          =CL(L'MESSAGES)'TRANSACT
008D8 C6C9D3C5C1404040          2272          =CL8'FILEA'
008E0 00000000          2273          =V(DFHEG1)|
008E4 00000024          2274          =F'30'
008E8 00000000          2275          =F'0'
008FC 00000000          2276          =F(0)

```

HLASM and friends: CICS ...

```

Z/OS V1 R13 BINDER      20:09:40 SATURDAY AUGUST 10, 2013
BATCH EMULATOR  JOB(GACUA  ) STEP(B  ) PGM= IEWL      PROCEDURE(LKED  )
IEW2278I 0352 INVOCATION PARAMETERS LIST,XREF
IEW2322I 1220 1      ORDER DFHEAG
IEW2322I 1220 2      INCLUDE SYSLIB(DFHEAG)
IEW2322I 1220 3      NAME DFHEABRW(R)
IEW2650I 5102 MODULE ENTRY NOT PROVIDED.  ENTRY DEFAULTS TO SECTION DFHEAG.

                                C R O S S - R E F E R E N C E   T A B L E

TEXT CLASS = B_TEXT

----- REFERENCE ----- TARGET -----
CLASS      ELEMENT      | SYMBOL (ABBREV)  SECTION (ABBREV)  ELEMENT
OFFSET SECT/PART(ABBREV)  |                   |                   | OFFSET CLASS
-----|-----|-----|-----|-----|
60 DFHEABRW      | 38 V-CON | DFHEAG0  | DFHEAG0  | 0 B_TEX
908 DFHEABRW     | 8E0 V-CON | DFHEG1   | DFHEAG   | 8 B_TEX
*** END OF CROSS REFERENCE ***

```

HLASM and friends: MQ

Data structures

provided as macros:

CMQODA	<i>object descriptor</i>
CMQMDA	<i>message descriptor</i>
CMQGMOA	<i>get message options</i>
CMQA	<i>constants</i>
...	

Found in *mqhlq.SCSQMACS*

HLASM and friends: MQ ...

Data structures ...

```
3592 ZEROREC  DC    F'0'          check for zero length
3593 *
3594 OBJDESC   CMQODA  DSECT=NO,LIST=NO  object descriptor
3595+         PUSH  PRINT
3596+         PRINT OFF
3647+         POP   PRINT
3648 MSGDESC   CMQMDA  DSECT=NO,LIST=NO  Message descriptor
3649+         PUSH  PRINT
3650+         PRINT OFF
3684+         POP   PRINT
3685 GETMSGOPTS CMQGMOA DSECT=NO,LIST=NO  Get message options
3686+         PUSH  PRINT
3687+         PRINT OFF
3708+         POP   PRINT
3709 *
```

HLASM and friends: MQ ...

```

BROWSE   MQM.V710.SCSQMACS(CMQODA)           Line 00000000 Col 001
Command ==>                               Scroll ==> C
***** Top of Data *****
MACRO
&INAME   CMQODA &DCLVER=,&DSECT=NO,&LIST=NO,&NESTED=NO,           X
          &STRUCID=00,                                           X
          &VERSION=MQOD_VERSION_1,                               X
          &OBJECTTYPE=MQOT_Q,                                   X
          &OBJECTNAME=,                                         X
          &OBJECTQMGRNAME=,                                     X
          &DYNAMICQNAME=CSQ.*,                                  X
          &ALTERNATEUSERID=,                                   X
          &RECSPRESENT=0,                                       X
          &KNOWNDDESTCOUNT=0,                                 X
          &UNKNOWNDESTCOUNT=0,                               X
          &INVALIDDESTCOUNT=0,                               X
          &OBJECTRECOFFSET=0,                                  X
          &RESPONSERECOFFSET=0,                                X
          &OBJECTRECPTTR=0,                                    X
          &RESPONSERECPTTR=0,                                  X
          &ALTERNATESECURITYID=00,                             X

```

&VERSION=MQOD_VERSION_1,
&OBJECTTYPE=MQOT_Q,

constants defined in CMQA

```

BROWSE   MQM.V710.SCSQMACS(CMQA)
Command ==>
.* Structure Version Number
MQOD_VERSION_1 EQU 1
MQOD_VERSION_2 EQU 2
MQOD_VERSION_3 EQU 3
MQOD_VERSION_4 EQU 4
MQOD_CURRENT_VERSION EQU 4
.*

```

```

BROWSE   MQM.V710.SCSQMACS(CMQODA)
Command ==>
&LNAME._STRUCID DC CL4'&STRUCID'
&LNAME._VERSION DC A(&VERSION)
&LNAME._OBJECTTYPE DC A(&OBJECTTYPE)
&LNAME._OBJECTNAME DC CL48'&OBJECTNAME'
&LNAME._OBJECTQMGRNAME DC CL40'&OBJECTQMGRNAME'

```

HLASM and friends: MQ ...

API calls

... uses standard assembler CALL macro

MQOPEN	<i>open a queue</i>
MQGET	<i>get a message</i>
MQPUT	<i>put a message</i>
MQCLOSE	<i>close a queue</i>

HLASM and friends: MQ ...

```

2718 ****
2719 * DSECTS USED BY THIS PROGRAM
2720 ****
2721 WORKAREA DSECT
2722 *
2723 PARMLIST CALL , (0,0,0,0,0,0,0,0,0,0,0,0), VL, MF=L
2727+PARMLIST DS      0F
2728+          DC      A(0)          PROB. PROG. PARAMETER
2729+          DC      A(0)          PROB. PROG. PARAMETER
2730+          DC      A(0)          PROB. PROG. PARAMETER
2731+          DC      A(0)          PROB. PROG. PARAMETER
2732+          DC      A(0)          PROB. PROG. PARAMETER
2733+          DC      A(0)          PROB. PROG. PARAMETER
2734+          DC      A(0)          PROB. PROG. PARAMETER
2735+          DC      A(0)          PROB. PROG. PARAMETER
2736+          DC      A(0)          PROB. PROG. PARAMETER
2737+          DC      A(0)          PROB. PROG. PARAMETER
2738+          DC      A(0+X'80000000')
2739 *

```

HLASM and friends: MQ ...

```

3031 *****
3032 MAINCONN DS    0H
3033          XC    HCONN,HCONN          Null connection
3034 *
3035          CALL  MQCONN,
              (MQMNAME,
              HCONN,
              COMPCODE,
              REASON),
              MF=(E, PARMLIST), VL
3037+          DS    0H
3042+          LA   1, PARMLIST          LOAD PARAM
3043+          LA   14, MQMNAME         PICKUP PARAMETER
3044+          LA   15, HCONN          PICKUP PARAMETER
3045+          LA   0, COMPCODE        PICKUP PARAMETER
3046+          STM  14, 0, 0(1)        STORE INTO
3047+          LA   14, REASON         PICKUP PARAMETER
3048+          ST   14, 12(0, 1)       STORE INTO
3049+          OI   12(1), X'80'      SET LAST WORD BI
3050+          CNOP 0, 4
3051+          BRAS 15, *+8            BRANCH AR
3052+          DC   V(MQCONN)          ENTRY POINT
3053+          L    15, 0(15, 0)       LOAD 15 W
3054+          BALR 14, 15             BRANCH TO
3055 *
3056          LA   R0, MQCC_OK         Expected compc
3057          C    R0, COMPCODE        As expected ?
3058          BER  R6                 Yes .. return
3059 *
3060          MVC  INF4_TYP, =CL10'CONNECT

```

MQCONN
with four
parameters

HLASM and friends: MQ ...

```

3091 *
3092      CALL MQOPEN,
          (HCONN,
           OBJDESC,
           OPTIONS,
           HOBJ,
           COMPCODE,
           REASON),
          MF=(E, PARMLIST), VL
3094+      DS      0H
3099+      LA      1, PARMLIST
3100+      LA      14, HCONN
3101+      LA      15, OBJDESC
3102+      LA      0, OPTIONS
3103+      STM     14, 0, 0(1)
3104+      LA      14, HOBJ
3105+      LA      15, COMPCODE
3106+      LA      0, REASON
3107+      STM     14, 0, 12(1)
3108+      OI      20(1), X'80'
3109+      CNOP    0, 4
3110+      BRAS   15, *+8
3111+      DC      V(MQOPEN)
3112+      L       15, 0(15, 0)
3113+      BALR   14, 15
3114 *
3115      LA      R0, MQCC_OK
3116      C       R0, COMPCODE

```

MQOPEN
with six
parameters

HLASM and friends: MQ ...

```
000001E4 9680 100C          0000000C          3049+         OI    12(1),X'80'  
000001E8                   3050+         CNOP   0.4  
000001E8 A7F5 0004          000001F0          3051+         BRAS  15,*+8  
000001EC 00000000          3052+         DC    V(MQCONN)  
000001F0 58FF 0000          00000000          3053+         L     15,0(15,0)  
000001F4 05EF                   3054+         BALR  14,15  
                               3055 *  
000001F6 4100 0000          00000000          3056         LA    R0,MQCC_OK  
000001FA 5900 A0D4          000000D4          3057         C     R0,COMPCODE  
000001FE 0786                   3058         BER   R6
```

- BRAS r15 addresses v-conn
- DC V(MQCONN) v-conn resolved at load
- L 15,0(15,0) r15 loads vconn
- BALR ..and go invoke

HLASM and friends: MQ ...

- The MQ api is a call interface
 - Assembler programs must observe the OS linkage convention
 - Before you issue any MQ apl call, your programs must point R13 at a save area of at least 18 full words
 - Create static versions of the structures
 - Allocate storage for dynamic structures
 - Copy static versions to dynamic storage
 - Invoke API

MQ sample can be found in *mqlq.SCSQASMS*

HLASM and friends: MQ ...

```

BROWSE  MQM.V710.SCSQMACS(CMQGMOA)          Line 00000000 Col 001 00
Command ==> _____ Scroll ==> CSF
***** Top of Data *****
      MACRO
&INAME  CMQGMOA &DCLVER=,&DSECT=NO,&LIST=NO,&NESTED=NO,          X
          &STRUCID=GMO,                                         X
          &VERSION=MQGMO_VERSION_1,                             X
          &OPTIONS=(MQGMO_NO_WAIT+MQGMO_PROPERTIES_AS_Q_DEF),  X
          &WAITINTERVAL=0,                                     X
          &SIGNAL1=0,                                          X
          &SIGNAL2=0.                                          X

```

&OPTIONS=(MQGMO_NO_WAIT+MQGMO_PROPERTIES_AS_Q_DEF),

```

BROWSE  MQM.V710.SCSQMACS(CMQQA)
Command ==> _____
.* Get Message Options
MQGMO_WAIT EQU X'00000001'
MQGMO_NO_WAIT EQU X'00000000'

```

```

BROWSE  MQM.V710.SCSQMACS(CMQQA)
Command ==> _____
MQGMO_PROPERTIES_IN_HANDLE EQU X'00000000'
MQGMO_PROPERTIES_COMPATIBILITY EQU X'10000000'
MQGMO_PROPERTIES_AS_Q_DEF EQU X'00000000'
MQGMO_NONE EQU X'00000000'

```

OPTIONS = (ZERO + ZERO) !

HLASM and friends: MQ ...

```

00000286          3128 MSGSINIT DS      0H
00000286 4100 0060      00000060 3129      LA      R0,MQGMO_BROWSE_NEXT+MQGMO_ACCEPT_TRUNCATED_MSG
0000028A 5A00 C20C      0000120C 3130      A       R0,=A(MQGMO_FAIL_IF QUIESCING)
0000028E 5000 B9E0      000009E0 3131      ST      R0,GETMSGOPTS_OPTIONS      Indicate get options
          3132 *
  
```

```

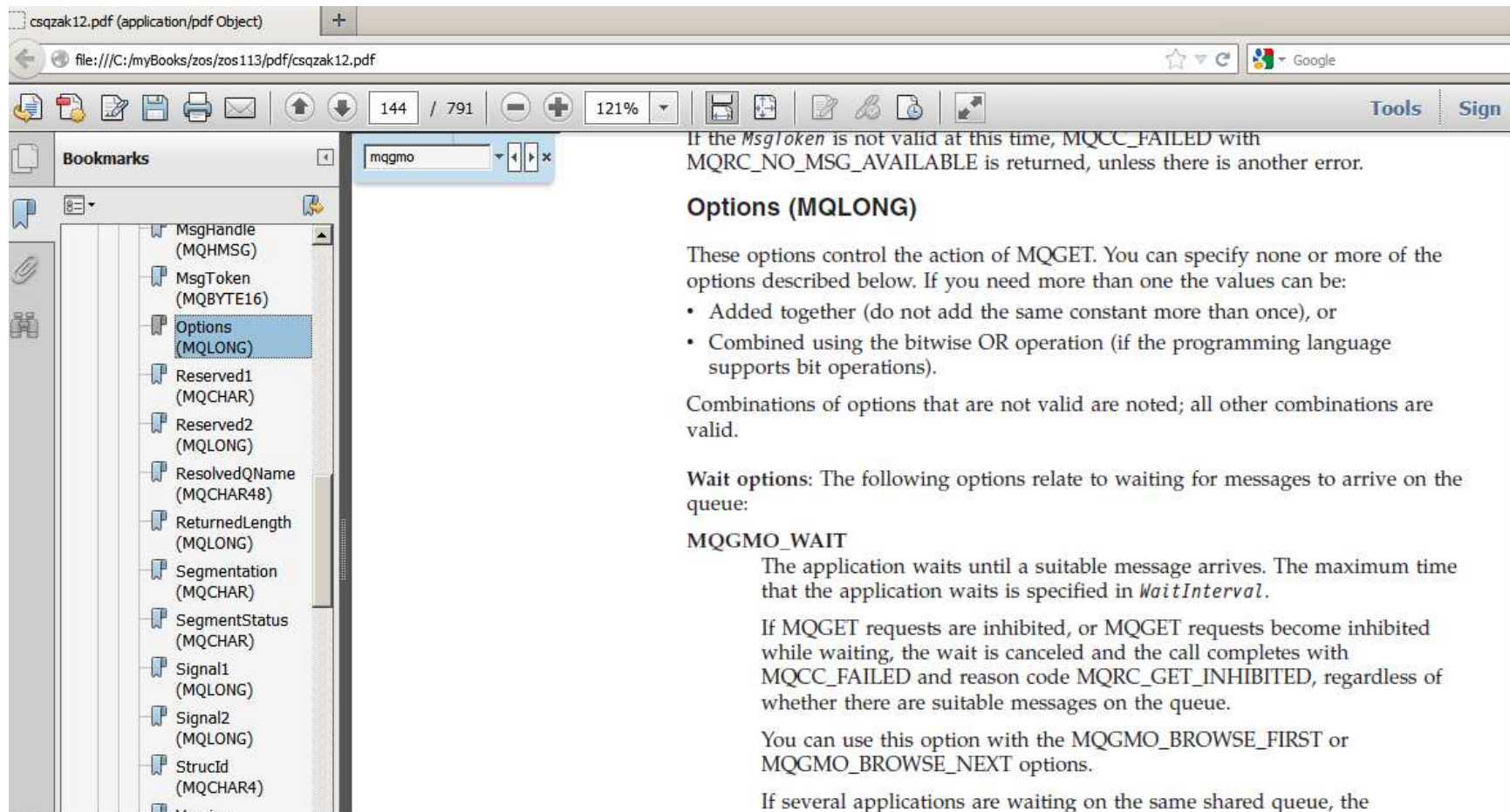
MQGMO_BROWSE_NEXT      EQU X'00000020'
MQGMO_ACCEPT_TRUNCATED_MSG EQU X'00000040'
MQGMO_FAIL_IF QUIESCING EQU X'00002000'
  
```

...and save the result ...

```

ST  R0,GETMSGOPTS_OPTIONS      Indicate get options
  
```

HLASM and friends: MQ ...



csqzak12.pdf (application/pdf Object)

file:///C:/myBooks/zos/zos113/pdf/csqzak12.pdf

144 / 791 121%

Tools Sign

Bookmarks

mqgmo

MsgHandle (MQHMSG)

MsgToken (MQBYTE16)

Options (MQLONG)

Reserved1 (MQCHAR)

Reserved2 (MQLONG)

ResolvedQName (MQCHAR48)

ReturnedLength (MQLONG)

Segmentation (MQCHAR)

SegmentStatus (MQCHAR)

Signal1 (MQLONG)

Signal2 (MQLONG)

StrucId (MQCHAR4)

Version

If the *MsgToken* is not valid at this time, MQCC_FAILED with MQRC_NO_MSG_AVAILABLE is returned, unless there is another error.

Options (MQLONG)

These options control the action of MQGET. You can specify none or more of the options described below. If you need more than one the values can be:

- Added together (do not add the same constant more than once), or
- Combined using the bitwise OR operation (if the programming language supports bit operations).

Combinations of options that are not valid are noted; all other combinations are valid.

Wait options: The following options relate to waiting for messages to arrive on the queue:

MQGMO_WAIT

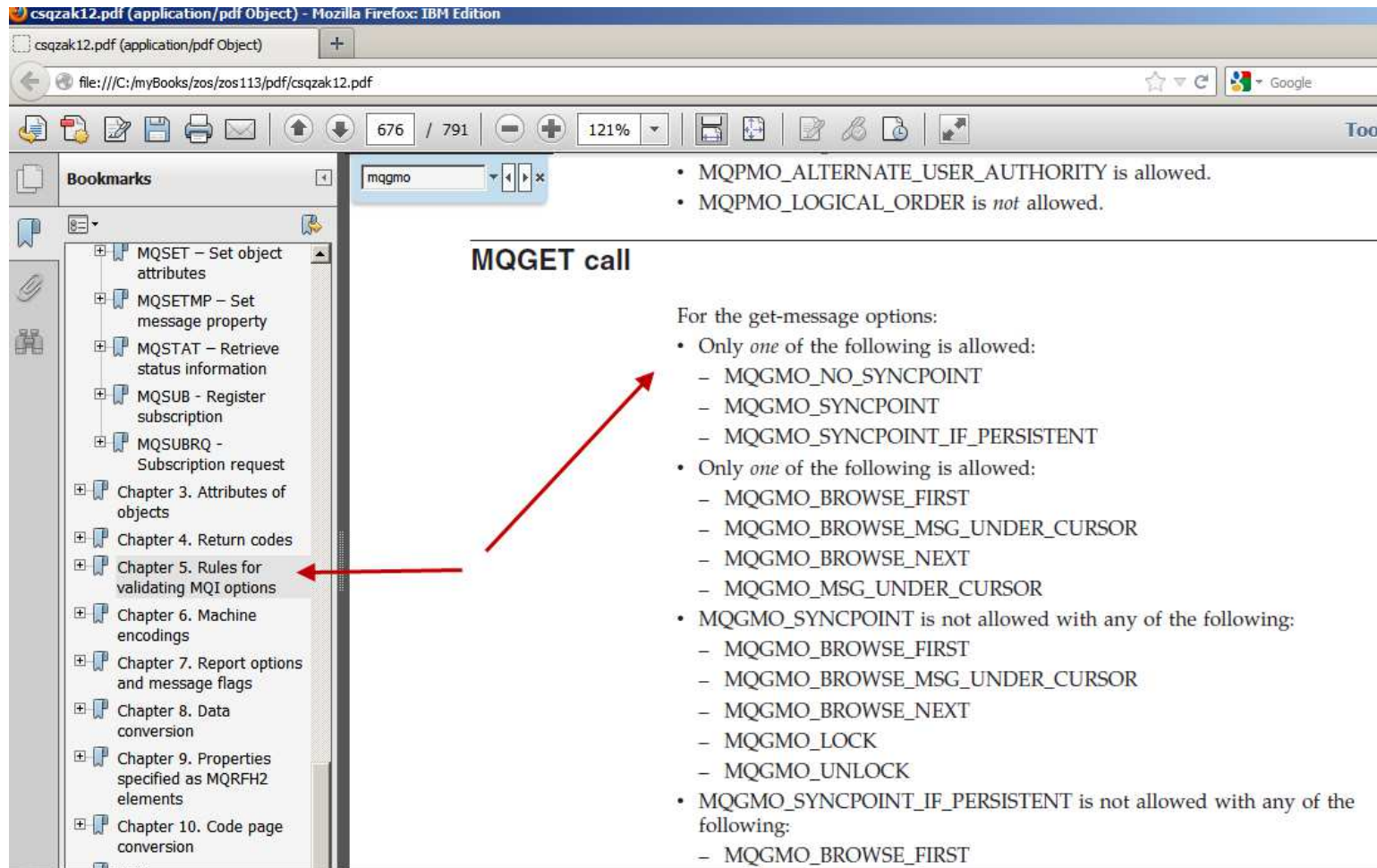
The application waits until a suitable message arrives. The maximum time that the application waits is specified in *WaitInterval*.

If MQGET requests are inhibited, or MQGET requests become inhibited while waiting, the wait is canceled and the call completes with MQCC_FAILED and reason code MQRC_GET_INHIBITED, regardless of whether there are suitable messages on the queue.

You can use this option with the MQGMO_BROWSE_FIRST or MQGMO_BROWSE_NEXT options.

If several applications are waiting on the same shared queue, the

HLASM and friends: MQ ...



csqzak12.pdf (application/pdf Object) - Mozilla Firefox IBM Edition

file:///C:/myBooks/zos/zos113/pdf/csqzak12.pdf

676 / 791 121%

Bookmarks

- MQSET – Set object attributes
- MQSETMP – Set message property
- MQSTAT – Retrieve status information
- MQSUB - Register subscription
- MQSUBRQ - Subscription request
- Chapter 3. Attributes of objects
- Chapter 4. Return codes
- Chapter 5. Rules for validating MQI options
- Chapter 6. Machine encodings
- Chapter 7. Report options and message flags
- Chapter 8. Data conversion
- Chapter 9. Properties specified as MQRFH2 elements
- Chapter 10. Code page conversion

mqgmo

- MQPMO_ALTERNATE_USER_AUTHORITY is allowed.
- MQPMO_LOGICAL_ORDER is *not* allowed.

MQGET call

For the get-message options:

- Only *one* of the following is allowed:
 - MQGMO_NO_SYNCPOINT
 - MQGMO_SYNCPOINT
 - MQGMO_SYNCPOINT_IF_PERSISTENT
- Only *one* of the following is allowed:
 - MQGMO_BROWSE_FIRST
 - MQGMO_BROWSE_MSG_UNDER_CURSOR
 - MQGMO_BROWSE_NEXT
 - MQGMO_MSG_UNDER_CURSOR
- MQGMO_SYNCPOINT is not allowed with any of the following:
 - MQGMO_BROWSE_FIRST
 - MQGMO_BROWSE_MSG_UNDER_CURSOR
 - MQGMO_BROWSE_NEXT
 - MQGMO_LOCK
 - MQGMO_UNLOCK
- MQGMO_SYNCPOINT_IF_PERSISTENT is not allowed with any of the following:
 - MQGMO_BROWSE_FIRST

HLASM and friends: MQ ...

```

csg4baa1_listing.txt - Notepad
File Edit Format View Help
1z/OS V1 R13 BINDER      20:09:41 SATURDAY AUGUST 10, 2013
BATCH EMULATOR  JOB(GACUA  ) STEP(B  ) PGM= HEWL      PROCEDURE(L  )
IEW2278I B352 INVOCATION PARAMETERS - MAP,LET,LIST
IEW2322I 1220 1  INCLUDE MQM(CSQBSTUB)
IEW2322I 1220 2  NAME CSQ4BAA1(R)

1          *** M O D U L E  M A P ***

-----
CLASS  B_TEXT          LENGTH =      1728  ATTRIBUTES = CAT,    LOAD, RMODE= 24
                        OFFSET =           0 IN SEGMENT 001    ALIGN = DBLWORD
-----

SECTION  CLASS          NAME              TYPE      LENGTH  DDNAME  SOURCE  -----
OFFSETS  OFFSET          NAME              TYPE      LENGTH  DDNAME  SEQ  MEMBER
-----
          0  CSQ4BAA1          CSECT      1243  SYSLIN   01  CSQ4BAA1
1248  CSQAA          CSECT      100  MQM      01  CSQBSTUB
E4    132C  CS#AA          LABEL
FC    1344  CSQAA@          LABEL

1348  CSQBPTUB          CSECT     2E0  MQM      01  CSQBSTUB
4     134C  CS#BPTUB          LABEL
1C    1364  MQDISC          LABEL
1C    1364  CSQBDISC          LABEL
5C    13A4  MQOPEN          LABEL
5C    13A4  CSQBOPEN          LABEL
64    13AC  MQCLOSE          LABEL
64    13AC  CSQBCLOS          LABEL
6C    13B4  MQGET           LABEL
6C    13B4  CSQBGET          LABEL
74    13BC  MQPUT           LABEL
74    13BC  CSQBPUT          LABEL
7C    13C4  MQPUT1          LABEL
7C    13C4  CSQBPUT1          LABEL

```

HLASM and friends: MQ ...

```

Menu Functions Confirm Utilities Help
BROWSE MQM.V710.SCSQLOAD Row
Command ==>

```

Name	Prompt	Alias-of	Size	TTR
CSQBRRSI			000008A0	00003F
CSQBRSTB			00000898	000040
CSQBSET		CSQBSTUB	000004E0	00003B
CSQBSTAT		CSQBSTUB	000004E0	00003B
CSQBSTMP		CSQBSTUB	000004E0	00003B
CSQBSTUB			000004E0	00003B
CSQBSTU6			00001090	000024
CSQBSUB		CSQBSTUB	000004E0	00003B
CSQBSUBR		CSQBSTUB	000004E0	00003B
CSQBWSTB			00000858	00003D
CSQBWST6			00001170	00003C
CSQBXEP		CSQBSTUB	000004E0	00003B
CSQBXTUB			00000398	000018
CSQCCLDS		CSQCSTUB	00000768	00001A
CSQCCONN		CSQCSTUB	00000768	00001A
CSQCCONX		CSQCSTUB	00000768	00001A
CSQCDISC		CSQCSTUB	00000768	00001A
CSQCEAI0			00000018	000016

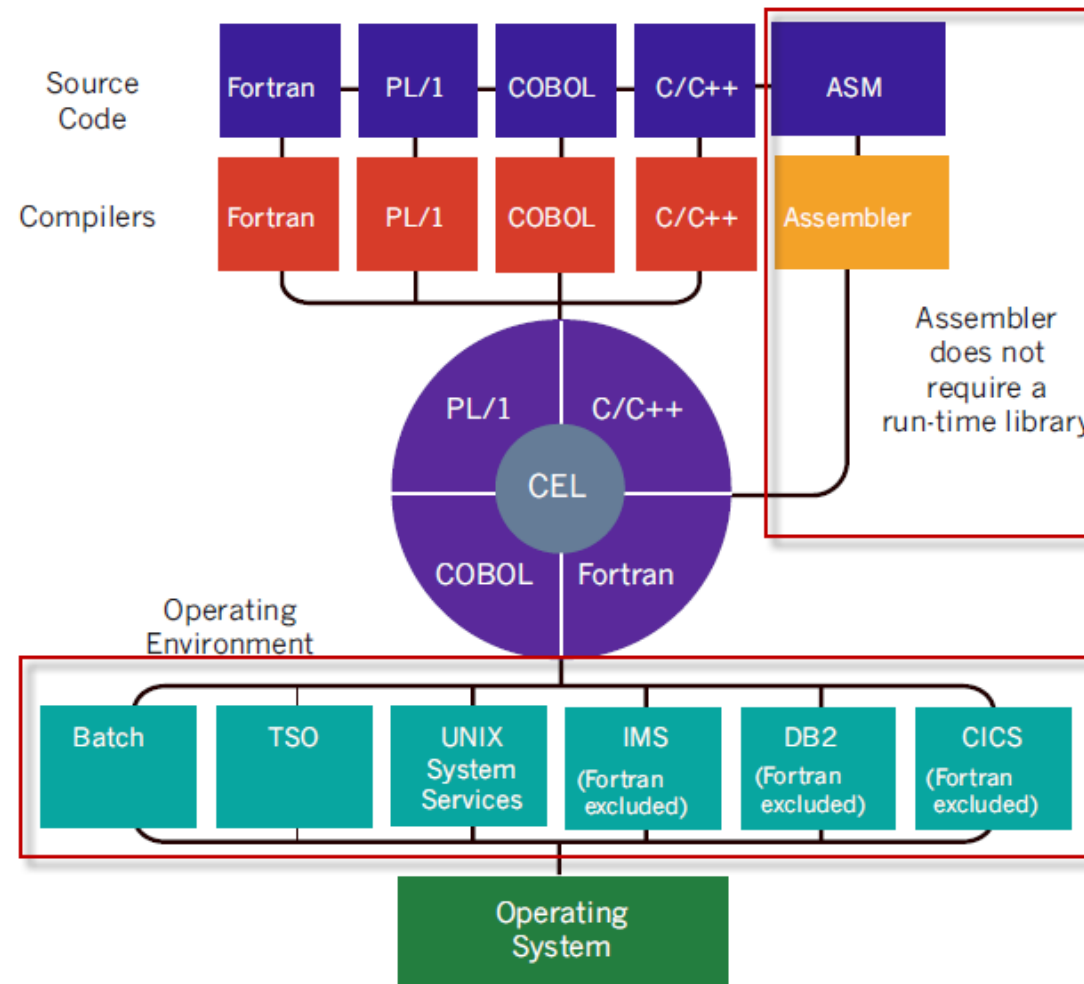
```

Menu Functions Confirm Utilities Help
BROWSE MQM.V710.SCSQLOAD Row
Command ==>

```

Name	Prompt	Alias-of	Size	TTR
CSQASTUB			00000410	00003B
CSQBASTB			00000830	00003B
CSQBBACK		CSQBSTUB	000004E0	00003B
CSQBBFMH		CSQBSTUB	000004E0	00003B
CSQBCB		CSQBSTUB	000004E0	00003B
CSQBCLOS		CSQBSTUB	000004E0	00003B
CSQBCOMM		CSQBSTUB	000004E0	00003B
CSQBCONN		CSQBSTUB	000004E0	00003B
CSQBCONX		CSQBSTUB	000004E0	00003B
CSQBCTL		CSQBSTUB	000004E0	00003B
CSQBCTMH		CSQBSTUB	000004E0	00003B
CSQBDISC		CSQBSTUB	000004E0	00003B
CSQBDMH		CSQBSTUB	000004E0	00003B
CSQBDMT		CSQBSTUB	000004E0	00003B
CSQBGET		CSQBSTUB	000004E0	00003B

HLASM and friends: LE



HLASM and friends: LE ...

LE-conforming assembler

An LE-conforming assembler program is a program that uses the LE provided macros to generate the required prolog and epilog code, and follows the register conventions of LE.

Macros... CEEENTRY and CEETERM

Do you remember ...

*DFHEIENT DATAREG=... and DFHEIRET ... and DFHEISTG ... and
DFHEIEND ... from the earlier slides ?*

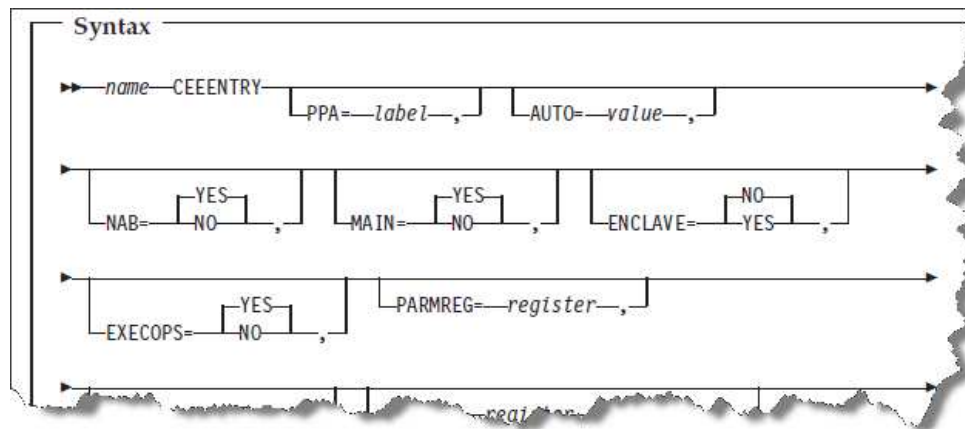
Is there anything new under the sun ?

HLASM and friends: LE ...

```

365 CEEIVP  CEEENTRY PPA=MAINPPA,AUTO=WORKSIZE
366+CEEIVP CSECT ,
367+CEEIVP RMODE ANY @D2
368+CEEIVP AMODE ANY @D2A
369+      ENTRY CEEIVP
370+      PUSH USING
371+      DROP , @02A
372+      USING *,15
375+      BRC 15,CEEZ0004 (BC)
376+      DC X'00C3C5C5'
377+CEEY0004 EQU ((WORKSIZE+7)/8)*8
+      . Size of automatic storage. @P1A
378+      DC A(CEEY0004) . Size of automatic storage. @P1
379+      DC A(MAINPPA-CEEIVP) . Address of PPA for this program
3

```



AUTO=WORKSIZE

Your programs working storage

HLASM and friends: LE ...

```

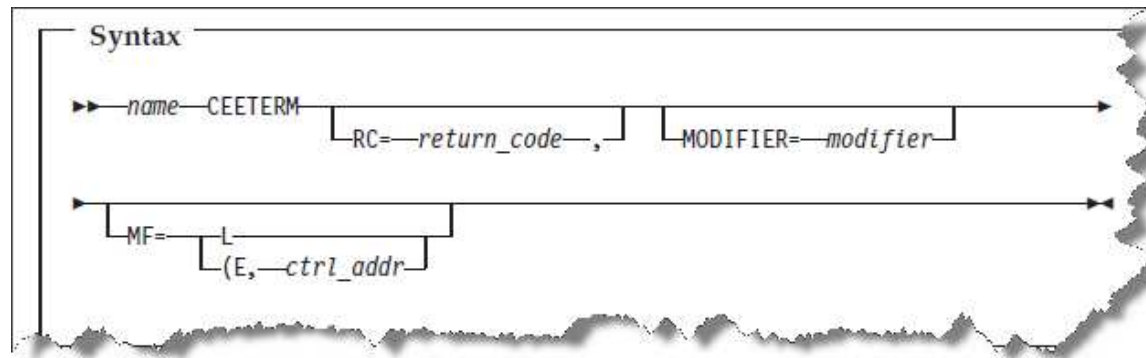
502          CEETERM  RC=0
503+         LA      1,CEET0010      Get address of termination
504+         L       15,=V(CEETREC)  Get address of termination
505+         BALR   14,15            Call termination routine

506+CEET0010 DC      A(*+8)          Parm 1
507+         DC      A(*+8+X'80000000') Parm 2

```

Par

Source: ... HLASM ...



return code can be a fixed constant, variable, or register 2-12

HLASM and friends: LE ...

```

BROWSE  PP.ADL370.ZOS113.SCEEMAC(CEEENTRY)      Line 00000000 Col 00
Command ==>                                     Scroll ==>
***** Top of Data *****
*/*****
*/
*/ Licensed Materials - Property of IBM
*/
*/ 5694-A01 5688-198
*/
*/ Copyright IBM Corp. 1991, 2009
*/
*/ Status = HLE7760
*/
*/*****
*
      MACRO
&NAME  CEEENTRY &PPA=PPA,      Label of PPA.          X
        &AUTO=CEEDSASZ,      Size of automatic storage.  X
        &MAIN=YES,           Main entry                      X
        &ENCLAVE=NO,         Nested enclave, main only      X
        &NAB=YES,           OK to use previous NAB value     X
        &PARMREG=1,         Register to hold parms.         X
        &BASE=11,           Base reg for this module.       X

```

```

BROWSE  PP.ADL370.ZOS113.SCEEMAC(CEETERM)      Line 00000000 Col 00
Command ==>                                     Scroll ==>
***** Top of Data *****
      MACRO
&NAME  CEETERM &RC=0,          Return code.          X
        &MODIFIER=0,         ENCL_MODIFIER             X
        &MF=
.*      &MF=                  LIST/EXECUTE FORM      @NX0179A
*****
.*
.* CEETERM - LE/370 Termination MACRO.
.*
.* LICENSED MATERIALS - PROPERTY OF IBM
.*
.* OS/390 LANGUAGE ENVIRONMENT
.*
.* 5647-A01
.*
.* (C) COPYRIGHT IBM CORP. 1991, 2000
.*

```

*LE macros are in
your 1eh1q..SCEEMAC*

HLASM and friends: LE ...

Using LE callable services... examples

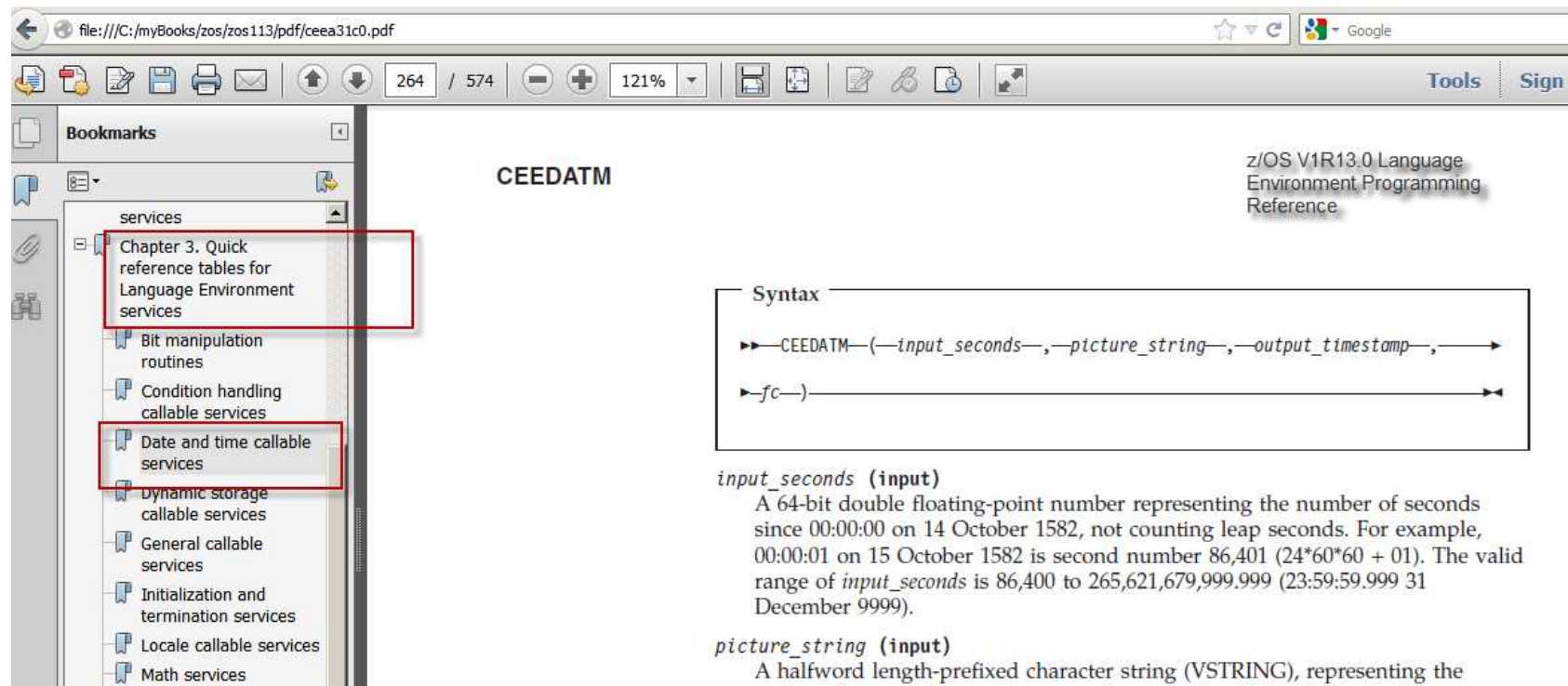
CEELOCT Gets current date and time

CEEDATM Converts number of seconds to character timestamp

CEEMOUT Write a message

The LE callable services are also available to other languages

HLASM and friends: LE ...



file:///C:/myBooks/zos/zos113/pdf/ceea31c0.pdf

264 / 574 121%

Tools Sign

Bookmarks

- services
 - Chapter 3. Quick reference tables for Language Environment services
 - Bit manipulation routines
 - Condition handling callable services
 - Date and time callable services**
 - Dynamic storage callable services
 - General callable services
 - Initialization and termination services
 - Locale callable services
 - Math services

CEEDATM

z/OS V1R13.0 Language Environment Programming Reference

Syntax

```
CEEDATM(—input_seconds—,—picture_string—,—output_timestamp—,—fc—)
```

input_seconds (input)
A 64-bit double floating-point number representing the number of seconds since 00:00:00 on 14 October 1582, not counting leap seconds. For example, 00:00:01 on 15 October 1582 is second number 86,401 (24*60*60 + 01). The valid range of *input_seconds* is 86,400 to 265,621,679,999.999 (23:59:59.999 31 December 9999).

picture_string (input)
A halfword length-prefixed character string (VSTRING), representing the

HLASM and friends: LE ...

```

519 MOUT      DC      V(CEEMOUT)      The CEL Message service
520 LOCT      DC      V(CEELOCT)      The CEL Local date/time
521 DATM      DC      V(CEEDATM)     The CEL date formatter
522 *

```

```

422
423 *  Setup the parameter list to CEELOCT. Omit the feedback
424 *
425      LA      R02,LILIAN
426      LA      R03,SECS
427      LA      R04,GREG
428      LA      R05,00      No FBCode here! wait for
429      STM     R02,R05,PLIST
430 *
431 *  Point to the parameter list and call CEELOCT
432 *
433      LA      R01,PLIST
434      L       R15,LOCT
435      BALR   R14,R15
436 *
437 *  Top of the loop. Use R3 to index thru the different

```

```

664 WORKAREA DSECT
665      ORG     *+CEEDSASZ      Leave space for the DSA fix
666 PLIST     DS      0D
667 PARM1    DS      A
668 PARM2    DS      A
669 PARM3    DS      A
670 PARM4    DS      A
671 PARM5    DS      A

```

...or simply use the standard call macro...

```
CALL CEELOCT(..., ..., ..., ...), VL, MF=(..)
```

HLASM and friends: LE ...

```

000011C 00000000      519 MOUT   DC      V(CEEMOUT)  The CEL Message serv
0000120 00000000      520 LOCT   DC      V(CEELOCT)  The CEL Local date/t
0000124 00000000      521 DATM  DC      V(CEEDATM)  The CEL date format
522 *
523 *
524 *
525 *
526 *
527 *
528 *
529 *
530 *
531 *
532 *
533 *
534 *
535 *
536 *
537 *
538 *
539 *
540 *
541 *
542 *
543 *
544 *
545 *
546 *
547 *
548 *
549 *
550 *
551 *
552 *
553 *
554 *
555 *
556 *
557 *
558 *
559 *
560 *
561 *
562 *
563 *
564 *
565 *
566 *
567 *
568 *
569 *
570 *
571 *
572 *
573 *
574 *
575 *
576 *
577 *
578 *
579 *
580 *
581 *
582 *
583 *
584 *
585 *
586 *
587 *
588 *
589 *
590 *
591 *
592 *
593 *
594 *
595 *
596 *
597 *
598 *
599 *
600 *
601 *
602 *
603 *
604 *
605 *
606 *
607 *
608 *
609 *
610 *
611 *
612 *
613 *
614 *
615 *
616 *
617 *
618 *
619 *
620 *
621 *
622 *
623 *
624 *
625 *
626 *
627 *
628 *
629 *
630 *
631 *
632 *
633 *
634 *
635 *
636 *
637 *
638 *
639 *
640 *
641 *
642 *
643 *
644 *
645 *
646 *
647 *
648 *
649 *
650 *
651 *
652 *
653 *
654 *
655 *
656 *
657 *
658 *
659 *
660 *
661 *
662 *
663 *
664 *
665 *
666 *
667 *
668 *
669 *
670 *
671 *
672 *
673 *
674 *
675 *
676 *
677 *
678 *
679 *
680 *
681 *
682 *
683 *
684 *
685 *
686 *
687 *
688 *
689 *
690 *
691 *
692 *
693 *
694 *
695 *
696 *
697 *
698 *
699 *
700 *
701 *
702 *
703 *
704 *
705 *
706 *
707 *
708 *
709 *
710 *
711 *
712 *
713 *
714 *
715 *
716 *
717 *
718 *
719 *
720 *
721 *
722 *
723 *
724 *
725 *
726 *
727 *
728 *
729 *
730 *
731 *
732 *
733 *
734 *
735 *
736 *
737 *
738 *
739 *
740 *
741 *
742 *
743 *
744 *
745 *
746 *
747 *
748 *
749 *
750 *
751 *
752 *
753 *
754 *
755 *
756 *
757 *
758 *
759 *
760 *
761 *
762 *
763 *
764 *
765 *
766 *
767 *
768 *
769 *
770 *
771 *
772 *
773 *
774 *
775 *
776 *
777 *
778 *
779 *
780 *
781 *
782 *
783 *
784 *
785 *
786 *
787 *
788 *
789 *
790 *
791 *
792 *
793 *
794 *
795 *
796 *
797 *
798 *
799 *
800 *
801 *
802 *
803 *
804 *
805 *
806 *
807 *
808 *
809 *
810 *
811 *
812 *
813 *
814 *
815 *
816 *
817 *
818 *
819 *
820 *
821 *
822 *
823 *
824 *
825 *
826 *
827 *
828 *
829 *
830 *
831 *
832 *
833 *
834 *
835 *
836 *
837 *
838 *
839 *
840 *
841 *
842 *
843 *
844 *
845 *
846 *
847 *
848 *
849 *
850 *
851 *
852 *
853 *
854 *
855 *
856 *
857 *
858 *
859 *
860 *
861 *
862 *
863 *
864 *
865 *
866 *
867 *
868 *
869 *
870 *
871 *
872 *
873 *
874 *
875 *
876 *
877 *
878 *
879 *
880 *
881 *
882 *
883 *
884 *
885 *
886 *
887 *
888 *
889 *
890 *
891 *
892 *
893 *
894 *
895 *
896 *
897 *
898 *
899 *
900 *
901 *
902 *
903 *
904 *
905 *
906 *
907 *
908 *
909 *
910 *
911 *
912 *
913 *
914 *
915 *
916 *
917 *
918 *
919 *
920 *
921 *
922 *
923 *
924 *
925 *
926 *
927 *
928 *
929 *
930 *
931 *
932 *
933 *
934 *
935 *
936 *
937 *
938 *
939 *
940 *
941 *
942 *
943 *
944 *
945 *
946 *
947 *
948 *
949 *
950 *
951 *
952 *
953 *
954 *
955 *
956 *
957 *
958 *
959 *
960 *
961 *
962 *
963 *
964 *
965 *
966 *
967 *
968 *
969 *
970 *
971 *
972 *
973 *
974 *
975 *
976 *
977 *
978 *
979 *
980 *
981 *
982 *
983 *
984 *
985 *
986 *
987 *
988 *
989 *
990 *
991 *
992 *
993 *
994 *
995 *
996 *
997 *
998 *
999 *
1000 *

```

Binder XREF ..

```

COMMAND INPUT ==>
SCROLL ==> CSR
----- REFERENCE ----- TARGET -----
CLASS          ELEMENT
OFFSET SECT/PART(ABBREV)  OFFSET TYPE  SYMBOL(ABBREV) SECTION (ABBREV)  ELEMENT
                                     CLASS NAME
54 CEEIVP      54 V-CON     CEEINT      CEEBPIRA          @ B_TEXT
11C CEEIVP     11C V-CON     CEEMOUT     CEEMOUT           @ B_TEXT
120 CEEIVP     120 V-CON     CEELOCT     CEELOCT           @ B_TEXT
124 CEEIVP     124 V-CON     CEEDATM     CEEDATM           @ B_TEXT
200 CEEIVP     200 V-CON     CEESTART    CEESTART          @ B_TEXT
204 CEEIVP     204 V-CON     CEEBETBL    CEEBETBL          @ B_TEXT
208 CEEIVP     208 A-CON     CEESTART    CEESTART          @ B_TEXT
300 CEEIVP     300 V-CON     CEETREC     CEETREC           @ B_TEXT
308 CEEMAIN    0 A-CON      CEEIVP      CEEIVP            @ B_TEXT
320 CEEBETBL    4 V-CON     CEEBETBL    CEEBETBL          @ B_TEXT

```

HLASM and friends: LE ...

CLASS OFFSET	NAME	TYPE	LENGTH	DDNAME	SEQ	SOURCE MEMBER
0	CEEIVP	CSECT	304	SYSLIN	01	CEEIVP
308	CEEMAIN	CSECT	8	SYSLIN	01	CEEIVP
310	CEETREC	* CSECT	14	SYSLIB	02	CEETREC
328	CEEBETBL	* CSECT	28	SYSLIB	02	CEEBETBL
350	CEESTART	* CSECT	80	SYSLIB	02	CEESTART
400	CEEDATM	* CSECT	14	SYSLIB	02	CEEDATM
418	CEELOCT	* CSECT	14	SYSLIB	02	CEELOCT
430	CEEMOUT	* CSECT	14	SYSLIB	02	CEEMOUT

```

-----
*** DATA SET SUMMARY ***
DDNAME   CONCAT  FILE IDENTIFICATION
SYSLIB   02     PP. ADLE370.ZOS113.SCEELKED
SYSLIN   01     SMORSA.BOSTON.ASM.OBJ
*** END OF MODULE MAP ***

```

```

ND INPUT ==>          SCROLL ==> CSF
29 //L.SYSLIB DD
   //SYSLIB DD DISP=SHR,DSN=&LOAD
   IEF653I SUBSTITUTION JCL - DISP=SHR,DSN=SMORSA.BOSTON.ASM.LOAD
80 //          DD DISP=SHR,DSN=PP.ADLE370.ZOS113.SCEELKED
31 //XOBJLIB DD DSN=&OBJ,DISP=SHR
   IEF653I SUBSTITUTION JCL - DSN=SMORSA.BOSTON.ASM.OBJ,DISP=SHR
32 //XSYSLMOD DD DISP=SHR,DSN=&LOAD.(&MEM)
   IEF653I SUBSTITUTION JCL - DSN=SMORSA.BOSTON.ASM.LOAD,DISP=SHR

```


HLASM and friends: LE ...

And finally

```
Today is Tuesday, 10 August 2013.  
Today is TUESDAY, AUGUST 10, 2013.  
Today is Tuesday, 08/10/13 20:19:40.41  
Today is day 222 of 2013  
Program Complete.
```