

z390 and zCOBOL Portable Mainframe Assembler and COBOL with zCICS Support

Don Higgins and Melvyn Maltz
Automated Software Tools Corporation

Tuesday, August 9, 2011
11:00-12:00 am

Trademark Acknowledgments



- **IBM Corporation**
 - z/OS, HLASM, CICS, VSAM
- **Microsoft Corporation**
 - Windows Vista, XP, and 2000
 - Visual Express C++
- **Sun Microsystems**
 - J2SE, J2RE

What is it ?



- A tool to develop and test Assembler and COBOL programs
- A training tool for those wishing to learn those languages
- CICS and VSAM are supported
- Runs under Windows and Linux

And it's free !!



Presentation Outline



- **z390 Portable Mainframe Assembler**
Assemble, link, execute HLASM compatible programs
- **zCOBOL Portable Mainframe COBOL**
Compile, link, execute COBOL programs
- **zCICS Support**
Support EXEC CICS COBOL and Assembler
Run local and remote TN3270 CICS transactions
- **Questions and Answers**

z390 Portable Mainframe Assembler



- **z390 Open Source Java Project**
- **Execute HLASM compatible macro code**
- **Assemble HLASM compatible programs**
- **Link object code into z390 load modules**
- **Execute load modules on J2SE platforms:**
 - **Windows (XP, Vista and 7) and Linux**
 - **24/31 bit AMODE/RMODE**
 - **32/64 bit GPR/FPR, HFP/BFP/DFP**
 - **QSAM, VSAM, SOA, CICS, TN3270**

z390 Portable Mainframe Assembler

What's new



- **Enhanced trace**

```
800FD8F2 0 D203A0085307 MVC S1(000FD6E8)=40404040 S2(00002307)=C4D6D5F0='DON0'  
800FD8F8 0 D203A014530B MVC S1(000FD6F4)=00000000 S2(0000230B)=D3D6C7D6='LOGO'  
800FD8FE 0 D2003DEE530F MVC S1(000FF5D6)=F6 S2(0000230F)=D2='K'  
800FD904 0 D503A0083184 CLC S1(000FD6E8)=C4D6D5F0 S2(000FE96C)=E2D8F0F1='SQ01'  
800FD90A 1 4770D1B2 BNE S2(000FD99A)=D2033E61A008 MVC
```

```
800FD8F2 0 D203A0085307 MVC S1(000FD6E8)=40404040 S2(00002307)=C4D6D5F0='DON0'  
Z390KCP 000112 Z390PARM MVC TCTTETI,ZCVTPARM+3 SET TERMID  
800FD8F8 0 D203A014530B MVC S1(000FD6F4)=00000000 S2(0000230B)=D3D6C7D6='LOGO'  
Z390KCP 000118 MVC TCTTEINT,ZCVTPARM+7 SET INITIAL TRANSID  
800FD8FE 0 D2003DEE530F MVC S1(000FF5D6)=F6 S2(0000230F)=D2='K'  
Z390KCP 00011E MVC INIDATE,ZCVTPARM+11 SET DATE FORMAT  
800FD904 0 D503A0083184 CLC S1(000FD6E8)=C4D6D5F0 S2(000FE96C)=E2D8F0F1='SQ01'  
Z390KCP 000124 CLC TCTTETI,=C'SQ01' SEQUENTIAL TERMINAL ?  
800FD90A 1 4770D1B2 BNE S2(000FD99A)=D2033E61A008 MVC  
Z390KCP 00012A BNE OPEN3270 EXIT IF NOT
```

z390 Portable Mainframe Assembler

What's new



- **ZSORT**

```
ZSORT ISORT,LRECL=15,FIELDS=(1,15,CH,A),MEMORY=500000
```

```
LOOP1 EQU *
```

```
...
```

```
ZSORT PUT,REC=REC
```

```
B LOOP1 LOOP
```

```
...
```

```
DOSORT EQU *
```

```
ZSORT GET,REC=REC INVOKE SORT
```

```
CHI R15,4
```

```
BE END_OF_FILE
```

```
B DOSORT
```

Batch sorting is implemented

The zCOBOL SORT verb is under development



Z390 Structured macro code



Example of conditional macro code:

```
:&I SETA 1  
AWHILE (&I LE &LIMIT)  
  AIF ('&ID(&I)' EQ 'DSH')  
    MNOTE 'FOUND ID'  
  AEXIT AWHILE  
AEND  
:&I SETA &I+1  
AEND
```

Originally from HLASM Tools

Integrated in MZ390 macro processor

Other commands can be found in the ZSTRMAC documentation



Z390 Structured Programming Macros



Example of structured macros:

FIND SUBENTRY

LA R1,ID

LA R2,ID_END

WHILE (CLR,R1,LT,R2)

IF (CLC,0(3,R1),EQ,=C'DSH')

WTO 'FOUND ID'

SUBEXIT RC=0

ENDIF

AHI R1,3

ENDDO

WTO 'NOT FOUND'

SUBEXIT RC=1

z390 Compatibility Options



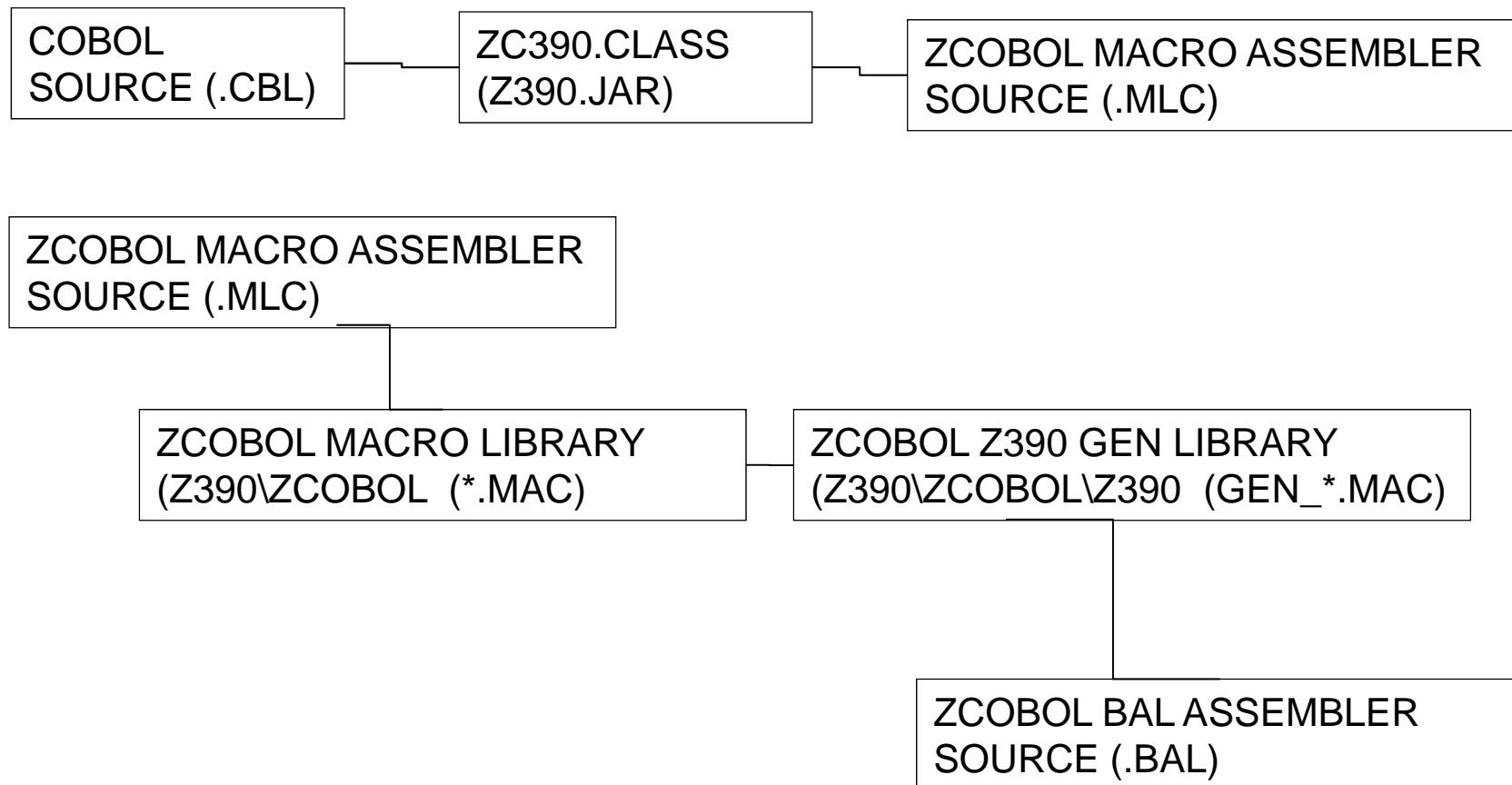
- VSE macros which map to MVS compatible z390 macros including CDLOAD, COMRG, EOJ, DTFPR, DTFSD, OPEN, CLOSE, GETIME, GETVIS
- Optional ASCII mode which generates ASCII character constants and does compares in ASCII collating sequence
- RECFM=FT/VT for ASCII to/from EBCDIC for QSAM file compatibility with ASCII text files
- Regression tests showing use of options

zCOBOL Portable Mainframe COBOL



- **Compiler architecture**
- **Compiler examples of source code generation**
- **Compiler code generation**
- **Compiler commands**
- **Demo and regression test programs**
- **What's new**
 - DISPLAY extended to support hex display of any field**

zCOBOL Portable Mainframe COBOL



zCOBOL Compiler Architecture



- **zc390.java parser CBL to MLC macro assembler**
- **zcobol library for all COBOL verb macros (139)**
- **zcobol\z390 library for all HLASM gen macros (102)**

- **Base-free code**
**Temporary regs used for WS and Linkage, some fixed
regs used for zCICS**

- **zcobol\java, vce, i586 for all java, C++, HLA/MASM code gen macros (11
each)***

*** Note: Once the z390 code gen macros are stabilized,
they will be copied to other target language libraries
and modified to support other source code generation
options.**

zCOBOL to z390 code gen Example 1



COBOL SOURCE:

77 CTR-1 COMP PIC S9(9).

01 SYSTEM-DATE.

02 SYSTEM-DD PIC 99.

02 SYSTEM-MM PIC 99.

HLASM > MACROS > BAL:

WS 77,CTR_1,COMP,PIC,S9(9)

- GEN_WS
CTR_1 DS FL4

WS 01,SYSTEM_DATE

WS 02,SYSTEM_DD,PIC,99

WS 02,SYSTEM_MM,PIC,99

- GEN_WS
 - SYSTEM_DATE DS 0CL4
 - SYSTEM_DD DS ZL2
 - SYSTEM_MM DS ZL2

zCOBOL to z390 code gen Example 2

SHARE
Technology · Connections · Results

IF CTR_1 = 2 GO TO OPT_2 .

IF CTR_1,=,2

- **GEN_COMP**

L R0,CTR_1

CHI R0,2

- **GEN_BC 7,PG_IF_1**

JNE PG_IF_1

GO TO,OPT_2

- **GEN_B PG_OPT_2**

J PG_OPT_2

PERIOD

- **GEN_LABEL PG_IF_1,ENDIF**

PG_IF_1 DS 0H

ENDIF

zCOBOL Compile Commands



- **ZC390C** – compile to z390 relocatable object code
- **ZC390CL** – compile and link z390 load module
- **ZC390CLG** – compile, link, and execute z390 pgm

- **ZCJAVCLG** – compile and execute J2SE java pgm
- **ZCVCECLG** – compile, link, and execute C++ pgm
- **ZC586CLG** – compile, link, and execute MASM pgm
- **Note other system software requirements (all free):**
 - All require J2SE and z390 installs
 - **ZCVCECLG** requires MS Visual Express C++ install
 - **ZC586CLG** requires HLA and MASM installs

zCOBOL Demo and Regression Tests



- Demos in `zcobol\demo` include:
 - HELLO.CBL - display "Hello World"
 - DATETIME.CBL- display current time and date
 - COPYFILE.CBL- copy line sequential file
- Regression tests in `zcobol\test` include:
 - TESTCMP1 – test ADD,SUBTRACT,MULTIPLY,DIVIDE
 - TESTFUN1 – test functions NUMERIC, etc.
 - TESTIF1 – test IF ELSE ENDIF
 - TESTISP1 - test INSPECT TALLY, REPLACING, etc.
 - TESTMOV1 – test MOVE including EDIT for DISPLAY
 - TESTPM1 – test PERFORM THRU, TIMES, VARYING
 - TESTSIX1 - test 2 dimensional subscripting

z390 and zCOBOL Documentation



All z390 and zCICS support documentation is on www.z390.org

Download link for z390 includes zCOBOL and zCICS

- Support link to submit RPI's for fixes and enhancements
- Documentation on assembler, linker, emulator, zCICS support

All the zCOBOL documentation is on www.zcobol.org

- Demo Programs User Guide
 - NIST ANSI 85 COBOL Test Suite Results
 - Options
 - Regression Test Programs

zCOBOL Group – join zcobol-subscribe@yahoogroups.com

zCICS Support for TN3270 local and remote terminals processing z390 and zCOBOL transactions over TCP/IP

Melvyn Maltz
Automated Software Tools Corporation

Tuesday, August 9, 2011
11:00-12:00 am

zCICS V8-V10



What's New...

BMS support for colour plus EXTATT, DSATT and MAPATT mapping parms

Non-terminal attach support

EXEC CICS INQUIRE/SET FILE

EXEC CICS ASSIGN

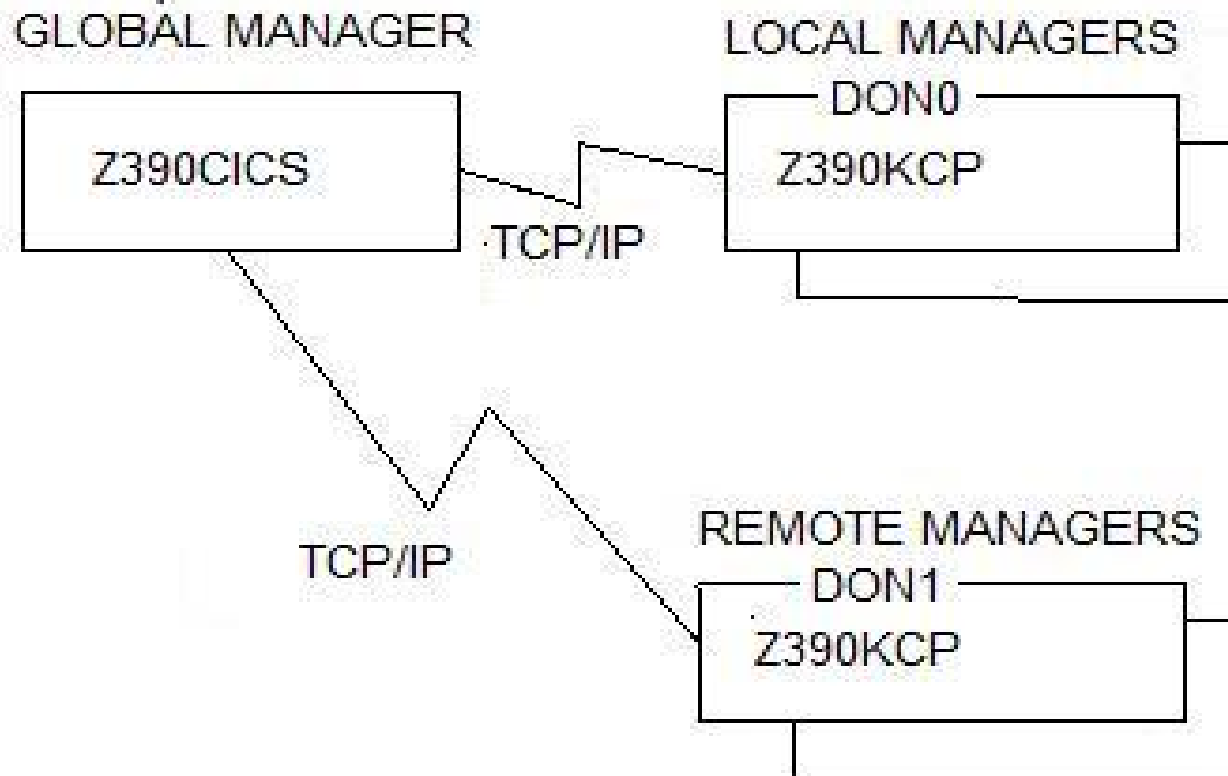
CEMT INQUIRE/SET FILE fully implemented using BMS

CEDF Intercepts, Working Storage and Redisplay Mode

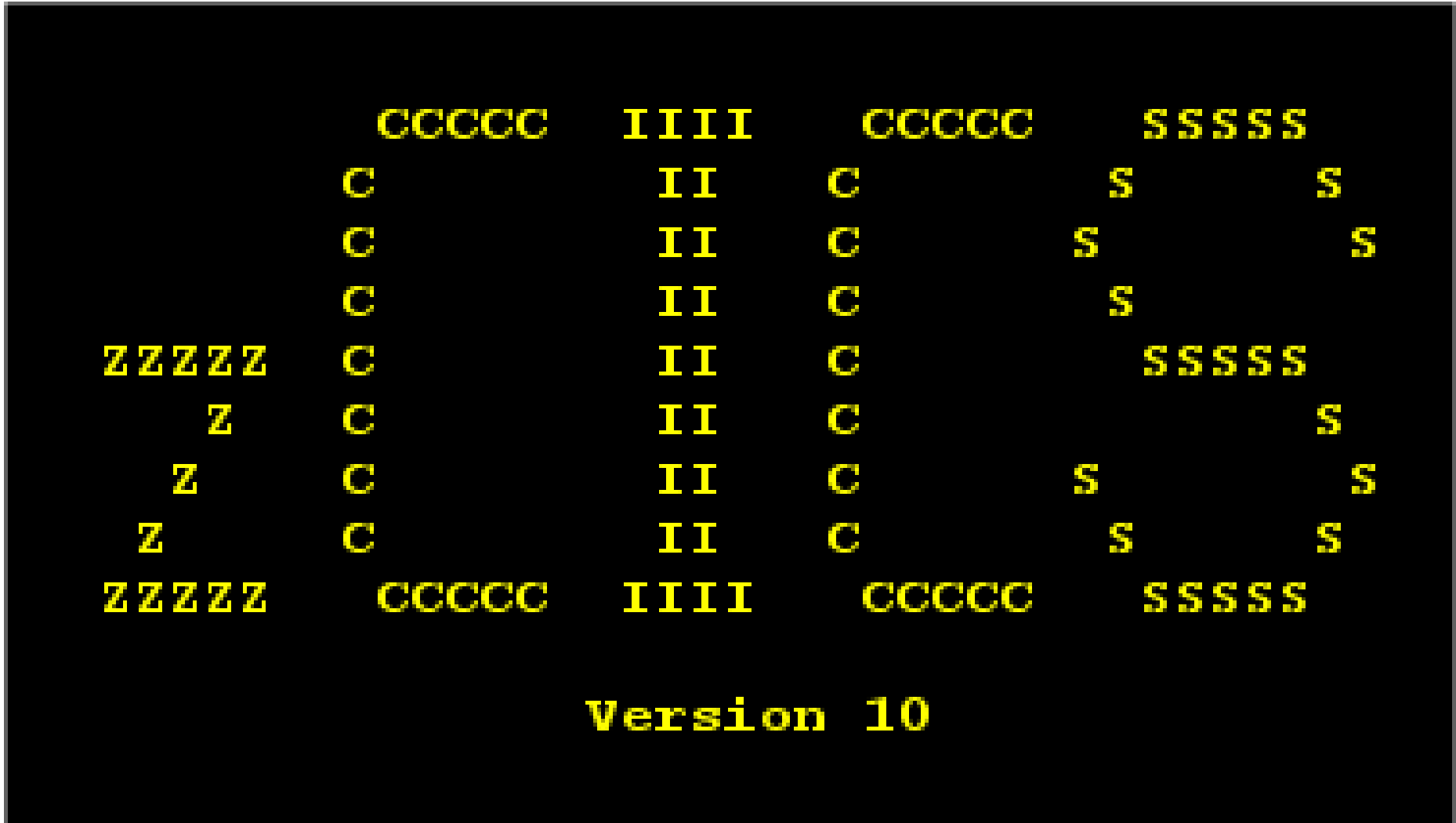
Corrections made to the current CICS manuals as a result of this Project.



zCICS Overview



zCICS GUI Screen



zCICS V10 supported commands



General

ADDRESS
ASSIGN
HANDLE AID
HANDLE CONDITION
IGNORE CONDITION
POP HANDLE
PUSH HANDLE

SC

FREEMAIN
GETMAIN

IC

ASKTIME
ASKTIME ABSTIME
DELAY
FORMATTIME
START
RETRIEVE
CANCEL

TC

RECEIVE
SEND

PC

ABEND
HANDLE ABEND
LINK
LOAD
RELEASE
RETURN
XCTL

KC

ENQ
DEQ

FC

READ
STARTBR
READNEXT
READPREV
ENDBR
RESETBR

BMS

SEND MAP
RECEIVE MAP
SEND CONTROL

DC

DUMP

System

INQUIRE FILE
SET FILE

→

zCICS BMS Extensions



- More cross-checking for Macro and execution
MAPFAIL now uses EIBRESP2.

ATTRB=(ALPHA)

XINIT=FFhh

PICIN/PICOUT supported by Assembler as an edit word

PICOUT=5C20216B202020

Data is 12345, displayed as *12,345

PICOUT=5B20216B202020

Data is 1234 , displayed as \$1,234

zCICS BMS Map Layout Example



```

•
•           1           2           3           4           5           6           7           8
•   1...+...0...+...0...+...0...+...0...+...0...+...0...+...0...+...0
• *****
• 1 *      @TESTGUI6 UPDATE NAME, ADDR, AND/OR ZIP (PF1=HELP PF2=ERASE INPUT PF3=EXIT)* 1
• 2 *                                                                                               * 2
• 3 *      @ENTER NAME@_____@                                                                 * 3
• 4 *                                                                                               * 4
• 5 *      @ENTER ADDR@_____@                                                                 * 5
• 6 *                                                                                               * 6
• 7 *      @ENTER ZIP @____@                                                                 * 7
• 8 *                                                                                               * 8
• 9 *      @...@..... * 9
• 10 *                                                                                               * 10
• 11 *      @...@..... * 11
• 12 *                                                                                               * 12
• 13 *      @...@..... * 13
• 14 *                                                                                               * 14
• 15 *      @PRESS F1 FOR HELP * 15
• 16 *                                                                                               * 16
• 17 *      @..... * 17
• 18 *                                                                                               * 18
• 19 *@TEST OCCURS      @...@...@...@...@...@...@SUM=@..... * 19
• 20 *@TEST GRPNAME     @..-..-..@      @..... * 20
• 21 *@TEST PICS        @.....@ * 21
• 22 *@.....@..... * 22
• 23 *@CURSOR LOCATION=@..... * 23
• *****
•           1           2           3           4           5           6           7           8
•   1...+...0...+...0...+...0...+...0...+...0...+...0...+...0
•
• →

```

zCICS Supplied Transactions



- **Many test transactions**
- **CEMT I TERm CEMT S TER OUT**
- **CEMT I TRAn CEMT P SHU**
- **CEMT I FILE CEMT P SHU IMM**
- **CEMT S FILE**
- **CEMT I SYStem**
- **CEMT I ENQueue**

- **CEBR**
- **CEDF**

zCICS CEMT INQUIRE/SET FILE 1



TERMINAL DON0 06/23/11 20:27:49

File Edit View Help

```
? FI AL
```

FILENAME	-----	STATUS	-----	DSNAME	-----
AIXNAME	CLO ENA REA	BRO ...	FIX E:\Z390\CICS\VSAM\Z390CAT1.AIXNAME		
AIXSURN	CLO ENA REA	BRO ...	FIX E:\Z390\CICS\VSAM\Z390CAT1.AIXSURN		
MYFILE01	OPE ENA REA	BRO ...	VAR E:\Z390\CICS\VSAM\Z390CAT1.MYFILE01		
MYFILE02	CLO ENA	ADD	FIX E:\Z390\CICS\VSAM\Z390CAT1.MYFILE02		
MYFILE03	CLO UNE REA UPD ADD	BRO DEL	FIX E:\Z390\CICS\VSAM\Z390CAT1.MYFILE03		
MYFILE04	CLO DIS REA UPD ADD	BRO DEL	FIX		
MYFILE05	CLO ENA REA UPD ADD	BRO DEL	FIX E:\Z390\CICS\VSAM\Z390CAT1.MYFILE02		
MYFILE06	CLO ENA REA UPD ADD	BRO DEL	FIX E:\Z390\CICS\VSAM\Z390CAT1.MYFILE06		
MYFILE07	CLO ENA REA UPD ADD	BRO DEL	VAR E:\Z390\CICS\VSAM\Z390CAT1.MYFILE07		
MYFILE08	CLO ENA REA UPD ADD	BRO DEL	FIX E:\Z390\CICS\VSAM\Z390CAT1.MYFILE08		
MYFILE09	CLO ENA REA UPD ADD	BRO DEL	VAR E:\Z390\CICS\VSAM\Z390CAT1.MYFILE09		
MYKSDS01	CLO ENA REA	BRO ...	FIX E:\Z390\CICS\VSAM\Z390CAT1.MYKSDS01		
MYWORD01	CLO ENA REA	BRO ...	FIX E:\Z390\CICS\VSAM\Z390CAT1.MYWORD01		
MYWORD02	CLO ENA REA	BRO ...	FIX E:\Z390\CICS\VSAM\Z390CAT1.MYWORD02		

CURSOR SELECT FILE CLEAR:END

Command:

Status:

Screen View Ready for input



zCICS CEMT INQUIRE/SET FILE 2



TERMINAL DON0 06/23/11 20:36:09

File Edit View Help

```
I F I A L

FILENAME ----- STATUS ----- DSNAME-----
AIXNAME  CLO ENA REA ... .. BRO ... FIX E:\Z390\CICS\VSAM\Z390CAT1.AIXNAME
AIXSURN  CLO ENA REA ... .. BRO ... FIX E:\Z390\CICS\VSAM\Z390CAT1.AIXSURN
MYFILE01 OPE ENA REA ... .. BRO ... VAR E:\Z390\CICS\VSAM\Z390CAT1.MYFILE01
MYFILE02 CLO ENA ... .. ADD ... .. FIX E:\Z390\CICS\VSAM\Z390CAT1.MYFILE02
MYFILE03 CLO UNE REA UPD ADD BRO DEL FIX E:\Z390\CICS\VSAM\Z390CAT1.MYFILE03
MYFILE04 CLO DIS REA UPD ADD BRO DEL FIX
MYFILE05 CLO ENA REA UPD ADD BRO DEL FIX E:\Z390\CICS\VSAM\Z390CAT1.MYFILE02
MYFILE06 CLO ENA REA UPD ADD BRO DEL FIX E:\Z390\CICS\VSAM\Z390CAT1.MYFILE06
MYFILE07 CLO ENA REA UPD ADD BRO DEL VAR E:\Z390\CICS\VSAM\Z390CAT1.MYFILE07
MYFILE08 CLO ENA REA UPD ADD BRO DEL FIX E:\Z390\CICS\VSAM\Z390CAT1.MYFILE08
MYFILE09 CLO ENA REA UPD ADD BRO DEL VAR E:\Z390\CICS\VSAM\Z390CAT1.MYFILE09
MYKSDS01 CLO dNA REA u.. a.. BRO ... FIX E:\Z390\CICS\VSAM\Z390CAT1.MYKSDS01
MYWORD01 CLO dNA REA u.. .. BRO d.. FIX E:\Z390\CICS\VSAM\Z390CAT1.MYWORD01
MYWORD02 CLO dNA REA ... a.. BRO ... FIX E:\Z390\CICS\VSAM\Z390CAT1.MYWORD02
```

CURSOR SELECT FILE CLEAR:END

Command:

Status:

Screen View Ready for input

zCICS CEMT INQUIRE/SET FILE 3



TERMINAL DON0 06/23/11 20:39:04

File Edit View Help

```
? FI AL
```

FILENAME	-----	STATUS	-----	DSNAME	-----
AIXNAME	CLO ENA REA ...	BRO ...	FIX E:\Z390\CICS\VSAM\Z390CAT1.AIXNAME		
AIXSURN	CLO ENA REA ...	BRO ...	FIX E:\Z390\CICS\VSAM\Z390CAT1.AIXSURN		
MYFILE01	OPE ENA REA ...	BRO ...	VAR E:\Z390\CICS\VSAM\Z390CAT1.MYFILE01		
MYFILE02	CLO ENA ...	ADD ...	FIX E:\Z390\CICS\VSAM\Z390CAT1.MYFILE02		
MYFILE03	CLO UNE REA UPD	ADD BRO DEL	FIX E:\Z390\CICS\VSAM\Z390CAT1.MYFILE03		
MYFILE04	CLO DIS REA UPD	ADD BRO DEL	FIX		
MYFILE05	CLO ENA REA UPD	ADD BRO DEL	FIX E:\Z390\CICS\VSAM\Z390CAT1.MYFILE02		
MYFILE06	CLO ENA REA UPD	ADD BRO DEL	FIX E:\Z390\CICS\VSAM\Z390CAT1.MYFILE06		
MYFILE07	CLO ENA REA UPD	ADD BRO DEL	VAR E:\Z390\CICS\VSAM\Z390CAT1.MYFILE07		
MYFILE08	CLO ENA REA UPD	ADD BRO DEL	FIX E:\Z390\CICS\VSAM\Z390CAT1.MYFILE08		
MYFILE09	CLO ENA REA UPD	ADD BRO DEL	VAR E:\Z390\CICS\VSAM\Z390CAT1.MYFILE09		
MYKSDS01	CLO DIS REA UPD	ADD BRO ...	FIX E:\Z390\CICS\VSAM\Z390CAT1.MYKSDS01		
MYWORD01	CLO DIS REA UPD	... BRO DEL	FIX E:\Z390\CICS\VSAM\Z390CAT1.MYWORD01		
MYWORD02	CLO DIS REA ...	ADD BRO ...	FIX E:\Z390\CICS\VSAM\Z390CAT1.MYWORD02		

CURSOR SELECT FILE CLEAR:END

Command:

Status:

Screen View Ready for input

zCICS CEMT INQUIRE/SET FILE 4



 **TERMINAL DON0 10/12/10 15:09:16**

File Edit View Help

INQUIRE FILE

```
File          ( MYFILE01)
Accessmethod( Vsam)
Dsname        ( E:\Z390\CICS\VSAM\Z390CAT1.MYFILE01)
Basedsname    ( E:\Z390\CICS\VSAM\Z390CAT1.MYFILE01)
Openstatus    ( Open)
Enablestatus  ( Enabled)
Readstatus    ( Readable)
Updatestatus  ( Notupdatable)
Addstatus     ( Notaddable)
Browsestatus  ( Browsable)
Deletestatus  ( Notdeletable)
Keylength     (      n/a      )
Keyposition   (      n/a      )
Object        ( Base)
Recordformat  ( Variable)
Recordsize    (           26)
Type          ( Esds)
```

PF3:RETURN TO LIST CLEAR:END

zCICS CEBR 1



TERMINAL DON0 01/03/08 22:40:14

File Edit View Help

```
CEBR                                                    EBCDIC
QNAME----- ITEMS | QNAME----- ITEMS | QNAME----- ITEMS
MYQUEUE1          16
MYQUEUE2          50
□□□□              80
VSM1              31
VSM2              27
VSM3              21
```

CURSOR SELECT QNAME : PF2=EBCDIC/ASCII/HEX : CLEAR TO END

Command:

Status:

Screen View

zCICS CEBR 2



```
TERMINAL DONO 12/08/07 21:45:11
File Edit View Help
CEBR VSM1          REC 17 OF 31 COL 1 OF 50 EBCDIC
ENTER COMMAND ===>

00017 abcdefgh
00018 abcdefghi
00019 Bill Brewer
00020 abcdefghi
00021 abcdefgh
00022 Jan Stewer
00023 abcdefghij
00024 abcdefg
00025 Peter Gurney
00026 Jan Stewer
00027 Peter Gurney
00028 Peter Davy
00029 Tom Cobley
00030 Harry Hawk
00031 Daniel Whiddon
***** BOTTOM OF QUEUE *****

PF1 : HELP          PF2 : EBCDIC/ASCII/HEX    PF3 : RETURN TO QNAMES
PF4 : VIEW TOP
PF7 : SCROLL BACK HALF  PF8 : SCROLL FORWARD HALF
PF10: SCROLL BACK FULL

Command:  Status:
Screen View Ready for input
```


zCICS CEDF 1



TERMINAL DON0 06/24/11 16:05:07

File Edit View Help

```
?TRANSACTION: ASGN PROGRAM: TESTASGN TASK: 0000004
STATUS: COMMAND EXECUTION COMPLETE
EXEC CICS ASSIGN
  ABCODE          (AEI0)
  ABDUMP          (X'00')
  ABPROGRAM      (TESTASGN)
  APLKYBD        (X'00')
  APLTEXT        (X'00')
  ASRAPSW        (X'0000000000000000')
  ASRAREGS R0 -R3 (00000000 00000000 00000000 00000000)
  ASRAREGS R4 -R7 (00000000 00000000 00000000 00000000)
  ASRAREGS R8 -R11 (00000000 00000000 00000000 00000000)
  ASRAREGS R12-R15 (00000000 00000000 00000000 00000000)
  BTRANS         (X'00')
  CMDSEC         ( )

PROGRAM: TESTASGN OFFSET: X'001AB4'   EIBFN: X'0208'
RESPONSE: NORMAL      EIBRESP2: 000

ENTER: CONTINUE

                                PF3 :END EDF SESSION
                                PF5 :WORKING STORAGE
                                PF8 :SCROLL FORWARD HALF
                                PF11:SCROLL FORWARD FULL PF12:REDISPLAY MODE
```

Command: Status:

Screen View Ready for input

zCICS CEDF 2



TERMINAL DON0 06/24/11 16:07:19

File Edit View Help

```
?TRANSACTION: ASGN PROGRAM: TESTASGN TASK: 0000004 EBCDIC
CICS DSA @ 000DA060/X'0070' USER DSA @ 000DA0D0/X'0000' DSA LENGTH=X'0070'
000DA060 000000 00000000 800FD818 00000000 00000000  □□□□□□□□□□□□□□□□
000DA070 000010 00000000 00000000 00000000 00000000  □□□□□□□□□□□□□□□□
000DA080 000020 00000000 00000000 00000000 00000000  □□□□□□□□□□□□□□□□
000DA090 000030 00000000 00000000 00000000 00000000  □□□□□□□□□□□□□□□□
000DA0A0 000040 00000000 00000000 000DAA18 00000000  □□□□□□□□□□□□□□□□
000DA0B0 000050 00000000 00000070 00000000 00000000  □□□□□□□□□□□□□□□□
000DA0C0 000060 40404040 40404040 40404040 40404040
```

ENTER: CURRENT DISPLAY

PF2 :EBCDIC/ASCII

Command:

Status:

Screen View Ready for input

zCICS CEDF 3



TERMINAL DONO 06/24/11 16:09:16 _ □ X

File Edit View Help

```
TRANSACTION: ASGN PROGRAM: TESTASGN TASK: 0000004          DISPLAY- 000 /025
STATUS:  COMMAND EXECUTION COMPLETE
EXEC CICS ASSIGN
  ABCODE          (AEI0)
  ABDUMP          (X'00')
  ABPROGRAM      (TESTASGN)
  APLKYBD        (X'00')
  APLTEXT        (X'00')
  ASRAPSW        (X'0000000000000000')
  ASRAREGS R0 -R3 (00000000 00000000 00000000 00000000)
  ASRAREGS R4 -R7 (00000000 00000000 00000000 00000000)
  ASRAREGS R8 -R11 (00000000 00000000 00000000 00000000)
  ASRAREGS R12-R15 (00000000 00000000 00000000 00000000)
  BTRANS         (X'00')
  CMDSEC         ( )

PROGRAM: TESTASGN OFFSET: X'001AB4'      EIBFN: X'0208'
RESPONSE: NORMAL          EIBRESP2: 000

ENTER: END REDISPLAY MODE

                                PF3 :END EDF SESSION
                                PF5 :WORKING STORAGE

PF7 :REDISPLAY BACK 1
PF10:REDISPLAY BACK 5          PF12:PAGING KEYS
```

Command: Status:

Screen View Ready for input

zCICS CEDF 4



TERMINAL DON0 06/24/11 16:11:44

File Edit View Help

```
TRANSACTION: ASGN PROGRAM: TESTASGN TASK: 000004 EBCDIC      DISPLAY- ?06 /025
STATUS:  COMMAND EXECUTION COMPLETE
EXEC CICS SEND
FROM  (C□G/R12:8008A402 R13:000DA060 R14:0008C314 R15:00000000)
LENGTH (00055)
```

```
PROGRAM: TESTASGN OFFSET: X'000726'      EIBFN: X'0404'
RESPONSE: NORMAL      EIBRESP2: 000
```

ENTER: END REDISPLAY MODE

```
PF2 :EBCDIC/ASCII/HEX      PF3 :END EDF SESSION
PF5 :WORKING STORAGE
PF7 :REDISPLAY BACK 1      PF8 :REDISPLAY FORWARD 1
PF10:REDISPLAY BACK 5      PF11:REDISPLAY FORWARD 5
```

Command:

Status:

Screen View Ready for input

zCICS Sequential Terminal Support



- **Regression test your transactions.**
- **Run a transaction with INI parm SEQ_TERM=TRACE**
- **Run the extract program Z390SEQ to build the data streams**
- **Sequence all of your data streams**
- **Application changes occur**
- **Set INI parm SEQ_TERM=YES**
- **Run the simulation, you can see it happen on screen**
- **Your whole life will flash before your eyes**

- **Run the comparator Z390CMPG, review the output**
- **Refine the comparator by building an exclusion file for variable data like dates and times**

zCICS Documentation 1



- There's a lot of it.
 - None of it is meant to replace IBM's Manuals.
 - The information given refers to zCICS, its implementation, workings, extensions and command/parameter support.

zCICS Documentation 2



- Readme
- Application Programming Guide
- Diagnosis Reference
- History
- Sequential Terminal Support
- Supplied Transactions
- System Programmer's Guide
- VSAM Guide
- Basic Mapping Support

And it's free

z390 zCOBOL zCICS Q and A Time



- Can I compile and test EXEC CICS COBOL programs using z390 zCICS?
- Which zCOBOL extension is highest priority?
- Which zCICS extension is highest priority?
- Which z390 extension is highest priority?
- How do I request a bug fix or enhancement?
- How can I volunteer to help?

z390 and zCOBOL Direction



The user community helps set direction

- z390 major priorities
 - Full VSAM update and alternate index support
 - SQL support
- zCOBOL major priorities are as follows:
 - NIST ANSI 85 test suite completion
 - Full VSAM update and alternate index support
 - SQL support
- zCICS priorities
 - VSAM AIX and update
 - CHANNELS and CONTAINERS

Submit RPI's for fixes and enhancements

- Join z390 and zCOBOL user groups for updates

