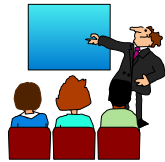




Language Environment Overview & Basic Debug Education

John Monti - March 1999
IBM Poughkeepsie
jmonti@us.ibm.com



► Basic Introduction

Its not easy!



■ "To succeed, preparation alone is insufficient. One must improvise as well. "
Salvor Hardin - Terminus City.



- ▶ It can be helpful to improvise with LE
- ▶
- ▶ The quote is by Salvor Hardin or Terminus City. From Issac Asimov's Science Fiction Foundation Series
- ▶
- ▶ Go from Science Fiction to Fantasy land...

Reflection

- "Language Environment is 100% compatible with previous runtime environments." - An unnamed misinformed marketing rep.
- "Read the LE migration guide for details of migration problems you **WILL** encounter." - An unnamed LE developer



- ▶ Really stress that LE is NOT 100% compatible - READ THE MIGRATION GUIDES!!!!

Agenda

- Introduction
- Language Environment Overview
- LE Messages and ABENDs
- LE Condition Handling
- LE Runtime Options
- CEEDUMPs
- SYSTEM DUMPs
- Summary
- Sources and Additional Information



► Agenda

Introduction

■ Language Environment

- A common runtime environment across multiple high level languages.

- COBOL
 - C/C++
 - PL/I
 - FORTRAN
 - Assembler (not HLL)

- Supported across multiple platforms

- OS/390
 - VM
 - VSE



- Detail the cross language support and various platforms
-
- Disavow any knowledge of VSE!

Introduction

- What is a runtime environment?
 - High level languages encapsulate complicated function into simple "functions".
 - Obtaining storage
 - Obtaining the system date and time
 - Outputting messages
 - Math functions
 - etc.
 - The runtime environment is "called" by the user program to do these pieces of work.
 - LE combines several runtimes.
 - Binding occurs at runtime not link time.

- ▶ Runtime is just a way to get the work done, without making the decision until runtime as to how the work will get done.

Language Environment (LE)

- Made up of several components

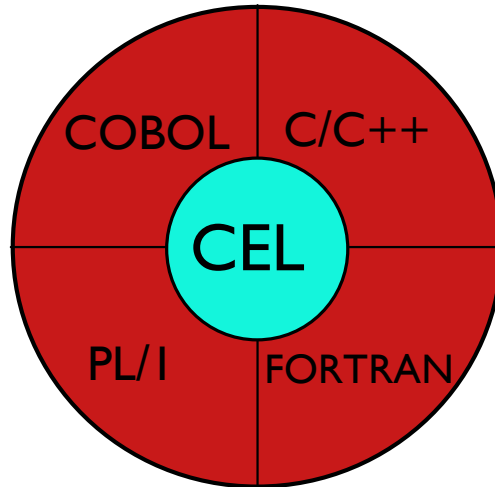
- Common Execution Library (CEL)
 - ▶ compid 568819801
- C/C++ Runtime Library (CRTL)
 - HFS header files
 - ▶ compid 568819805
- COBOL Runtime Library
 - ▶ compid 568819802
- PL/I Runtime Library
 - ▶ compid 568819803
- FORTRAN Runtime Library
 - ▶ compid 568819804



- ▶ Stress LE is still made up of separate runtimes

Language Environment (LE)

■ Often shown as:

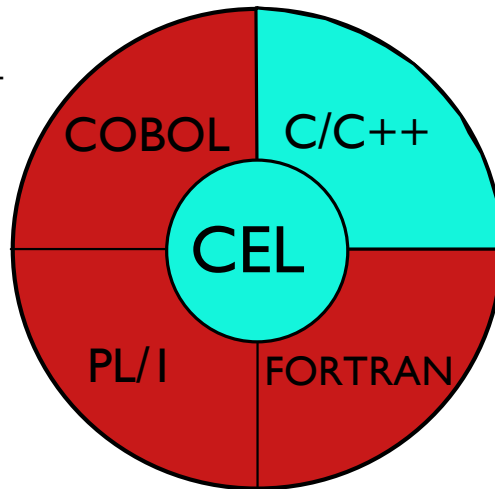


- Picture shows how CEL (LE) is in the middle and the various LE runtimes call CEL for common services.

Language Environment (LE)

■ Poughkeepsie Support:

- CEL
- C/C++



- C/C++ and CEL are supported in Pok
-
- COBOL, FORTRAN, PL1 - in STL

Language Environment (LE)

■ LE Release Chart

LE Release Level	Supported MVS/OS390 Release Level	FMID(s)
LE 1.5	MVS 4.3 MVS 5.1 MVS 5.2.2 OS/390 R1.1	HMWL510 JMWL5xx (each language has a separate FMID)
LE 1.6	OS/390 R1.2	HMWL610 JMWL6xx (each language has a separate FMID)
LE 1.7 (LE for OS/390 R3)	OS/390 R1.3	HMWL710 (languages in base FMID)
LE 1.8 (LE for OS/390 R4 and LE for OS/390 R5)	OS390 R2.4 OS390 R2.5	HMWL810 (Languages in base FMID)
LE 1.9 (LE for OS/390 R6)	OS390 R2.6	HMWL910 (Languages in base FMID)
LE 2.7 (LE for OS/390 R7)	OS390 R2.7	HLE607 (Languages in base FMID)

NOTE:

1. LE is upwardly compatible but not downwardly compatible. Applications linked with a particular release

of LE will run on all higher levels of LE (See APAR II11316)

2. OS/390 base elements require LE to be at the level shipped with that release of OS/390.

- ▶ Release chart for reference.
- ▶
- ▶ Mention the notes LE is upwardly compatible - NOT downwardly!!!

Language Environment (LE)

■ Which Libraries When?

- C/C++ Compile steps
 - ▶ SCEEH... in SYSLIB
- Prelink steps
 - ▶ SCEE OBJ and/or SCEECPP in SYSLIB
 - NOTE: Never SCEELKED or SCEELKEX.
- Link Edit/Binder step
 - ▶ SCEELKED in SYSLIB
 - ▶ SCEELKEX then SCEELKED in SYSLIB (Binder)
(LE 1.8 and up only)
 - NOTE: Never SCEERUN
- Run/Go step
 - ▶ SCEERUN in STEPLIB/LPA/LINKLST
 - NOTE: Never SCEELKED



▶ Concatenations:

▶

▶ Biggies are:

▶

▶ Don't use SCEERUN at link/binder time

▶

▶ Don't use SCEELKED and/or SCEELKEX at runtime

Common Execution Library (CEL)

- The **CEE** prefix!
- Set of common functions and routines used by all member languages of LE.
 - Initialization
 - Storage Management
 - Condition Handling (error processing)
 - ▶ Includes CEEDUMPs
 - Messaging
 - Date/Time services (Year 2000!!!)
 - Math Functions
 - Termination



- ▶ Highlevel of what the CEE prefix does

Common Execution Library (CEL)

■ Important modules to remember:

- **CEEHDSP**: Always the top module in CEEDUMPs. This module schedules the CEEDUMP to be taken, **IGNORE CEEHDSP!**
 - ▶ LE Condition handling modules start with CEEHxxxx.
- **CEEPLPKA**: LE's main load module. Contains most of the parts for CEL. Most LE reported ABENDs occur in CEEPLPKA. This does **NOT** indicate an LE problem. Use the data from a U4039 and/or CEEDUMP to determine what the problem is. (more later!)
- **CEEBINIT**: LE's main initialization load module.
 - ▶ LE Initialization modules will start with CEEBxxxx.
- **CEECCICS**: LE's main interface module with CICS.
 - ▶ LE CICS interface modules will start with CEECxxxx.

▶ Make sure they ignore CEEHDSP in CEEDUMP!!!!

▶

▶ Make sure they know CEEPLPKA is not at fault!

▶

▶

Common Execution Library (CEL)

■ Important modules to remember:

- CEEEVxxx: LE event handlers. The **xxx** is the member number of the language. Much like CEEPLPKA in that they contain many routines.

NOTE: you do not have to be a language to have an event handler!

- 003 C/C++ Runtime
- 005 COBOL
- 007 FORTRAN
- 008 DCE
- 010 PL/I
- 012 Debug Tool



- Event handlers
-
- Don't need to be languages

Messages and ABENDs

- Prefixes: Messages and Modules
 - CEE - Output by CEL, but may be reporting a problem elsewhere.
 - IGZ - Output by COBOL
 - IBM - Output by PL/I
 - AFH - Output by FORTRAN
 - EDC - Output by C/C++
- See Language Environment Debug Guide and Runtime Messages for exact details on LE messages and/or ABENDs.



► IGZ, EDC, IBM where around before LE

Messages and ABENDs

■ Common CEL messages:

- CEE3201S - Indicates ABEND0C1
- CEE3204S - Indicates ABEND0C4
- CEE32xxS - Indicates ABEND0Cy
 - ▶ where y is the hex equivalent of decimal xx
- CEE3250C - Indicates some non-0Cx ABEND occurred.
- CEE1000S - CICS only. Report LE ABEND to operator console.
- CEE0802C and CEE0813S - Error with **HEAP** storage (normally user problem).
- CEE0374C - Indicates some error (may or may not be with LE) occurred (examine token).



- ▶ Make sure they understand the CEE32xx messages are just ABEND0Cy
- ▶
- ▶ Token on next page for CEE0374C message

Messages and ABENDs

■ LE Condition Token (Feedback Code)

■ Example: 00030C84 59C3C5C5 xxxxxxxx

● 0003 | 0C84 | 59 | C3C5C5 | xxxxxxxx

▶ 0003 Severity

—0000 Informational (I)

—0001 Warning(W)

—0002 Error(E)

—0003 Severe(S)

—0004 Critical(C)

▶ 0C84 Hex message number (3204)

▶ 59 Flags (ignore)

▶ C3C5C5 Prefix (Facility ID)

▶ xxxxxxxx Used internally

● This token represents Message CEE3204S

- ▶ Break down the token with them.
- ▶
- ▶ Re-emphasize that this is an ABEND0C4!!!!

Messages and ABENDs

■ Common CEL ABENDs:

- U4038 - Some "severe" error occurred but no dump was requested (useless).
- U4039 - Some "severe" error occurred and a dump was requested (see CEEDUMP).
- U4083* - Savearea backchain in error.
- U4087* - Error during error processing.
 - U4083 and U4087 may mask original error.
(more later)
- U4093* - Error during initialization.
- U4094* - Error during termination.

* Need reason code to be meaningful.

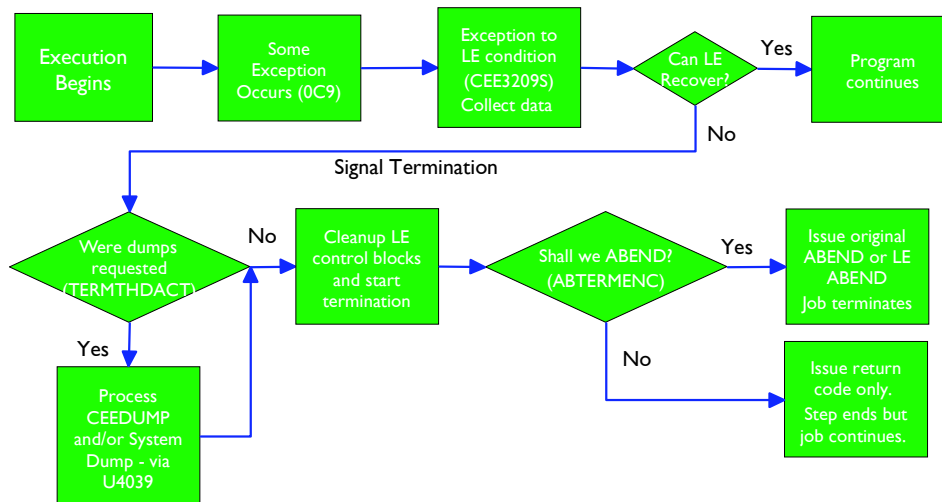


- ▶ U4038 no good
- ▶ U4039 Good - and may not be LE - We will go over it
- ▶
- ▶ U4083 and U4087 are errors during error processing. Be carefull they may mask the original problem. More later on how we might be fixing this.
- ▶
- ▶ U4093 and U4094 Need the reason codes. The book is pretty good.

Messages and ABENDs

■ Condition Handling Flow

● (Extremely over simplified)



- ▶ Go through the chart twice.
- ▶
- ▶ Stress the timing and how the re-issue of the ABEND is not the dump we want. We want the U4039 dump.
- ▶
- ▶ Stress the ABTERMENC(RETCODE) problem. Use a 3 step payroll job with the second step failing.
 - ▶ 1. First step reads the master file and creates a temp file
 - ▶ 2. Second step reads the raises and creates another temp file
 - ▶ 3. Third step combines the files and writes the checks
- ▶ If the second step fails with RETCODE 3rd step still runs.
- ▶

Messages and ABENDs

- Getting useful information
 - A CEEDUMP and/or a U4039 Dump
 - ▶ Remember U4038 does not generate a dump.
 - Use following LE runtime options to ask LE to take a dump (described later):
 - ▶ TERMTHDACT
 - ▶ TRAP
 - U4039 Dump
 - Allocate SYSMDUMP DD card.
 - ▶ Must point to a dataset (not SYSOUT)
 - ▶ LRECL 4160, RECFM FBS
 - See APAR III0573 for details



- ▶ This is the data we want.
- ▶
- ▶ SYSMDUMPs must go to data sets

Messages and ABENDs

- Getting useful information...
 - CEEDUMP
 - Batch
 - CEEDUMP DD card
 - May be to SYSOUT or Dataset
 - If not specified Dynamically allocated to SYSOUT=*
 - `_CEE_DUMPTARG` to SYSOUT=x (LE 2.7)
 - CICS - CESE Queue - (APAR PQ20892)
 - Unix System Services - Current working directory
 - `_CEE_DUMPTARG` to redirect (LE 1.9)

- CEEDUMPs and where they go.

Messages and ABENDs

■ Getting **NON-USEFUL** information:

- Remember: when ABTERMENC(ABEND) is set the "original" ABEND (like 0C4) is reissued.

- **DO NOT SLIP ON THIS ABEND!**

- LE reissues this ABEND at the end of LE termination.
- LE environment has been already cleaned-up and therefore a dump at this point is useless.
- Work with dump of U4039 instead.



► Again we don't want the re-issued ABEND dump!!!!

Runtime Options

■ Changing Runtime Options

- Installation Defaults - Samples in SCEESAMP
 - CEEWCOPT - CICS only
 - CEEWDOPT - all non-CICS environments
- Region Level (LE 2.7 and back to LE 1.8 with PQ22683)
 - CEEWROPT - CICS and IMS only
 - Unlike Installation Defaults, loaded at runtime
- Application Level
 - CEEWUOPT
 - CEEUOPT **MUST** be linked with the application
- Program Level
 - Compiled into program
 - #pragma runopts() in C
 - PLIXOPT for PL/I



- Runtime options!
-
- Region level is great if CICS is in the audience.
-
- UOPT must be LINKED

Runtime Options

■ Changing Runtime Options continued.

■ Program invocation

- Not in CICS but on a wish list.
- In PARM= statement on EXEC card
 - ▶ COBOL
 - PARM='program opts/LE runtime opts' (last slash)
 - ▶ C, PLI, FORTRAN, LE enabled Assembler
 - PARM='LE runtime opts/program opts' (first slash)
 - ▶ The '/' delimits the 2 sets of arguments and is required when LE opts are being set.

● Unix System Services shell

- ▶ Set the _CEE_RUNOPTS environment variable (use export as needed) no slash needed!

- ▶ No CICS invocation
- ▶
- ▶ COBOL backward from C-PLI-FORTRAN
- ▶
- ▶ OE not need for slash.

Runtime Options

■ Some important runtime options (See LE Customization for complete list)

- ABTERMENC(option)
 - RETCODE-Step ends with return code (job continues)
 - ABEND-Step will be ABENDED (job terminates)
 - ▶ Recommendations:
 - CICS - ABTERMENC(ABEND)
 - Others - ABTERMENC(ABEND)
- ALL31()
 - off for environments with AMODE 24 pgms.
 - ▶ In the future may be determined dynamically (see APARs PQ17931/PQ20527)
 - on for environments with all AMODE 31 pgms.
 - ▶ Recommendations:
 - CICS - ALL31(ON) whenever possible.
 - Others - ALL31(ON) whenever possible.



- ▶ ABTERMENC we talked about
- ▶
- ▶ ALL31 - use ON whenever possible
- ▶ Save below the line storage
- ▶ Save performance - NO AMODE switching code.

Runtime Options

■ Some important runtime options ...

■ ERRCOUNT()

- Terminates environment if number of conditions is exceeded.
- 0 - Allows infinite number of conditions to occur.
 - Required by PL/I for "on unit" processing - Each invocation of an "on unit" is an LE condition.
 - Required by C/C++ for signal processing - Signals are processed as LE conditions.
- >0 - Indicates the number of conditions allowed before LE terminates the process
 - Recommendations:
 - CICS - 0 (required)
 - Others - 0



- ERRCOUNT - Just make it 0

Runtime Options

■ Some important runtime options...

- HEAP(int,inc,where,type,.....)
 - int - Minimum size of initial heap segment
 - inc - Minimum size of any additional heap segments
 - where - BELOW, ANYWHERE
 - type
 - KEEP - do not free when empty (better performance)
 - FREE - free when empty (better memory management)
 - NOTES:
 - Used for COBOL working storage
 - Dynamic storage (C malloc/PLI allocate)
 - Recommendations
 - CICS - HEAP(4K,4080,ANYWHERE,KEEP,4K,4080)
 - Other - HEAP(32K,32K,ANYWHERE,KEEP,8K,4K)
 - When BELOW use FREE
 - When ANYWHERE use KEEP



- Heap
- Explain initial size
- Explain increment size
-
- Above or below
-
- Explain Keep and Free here.

Runtime Options

■ Some important runtime options...

■ STACK(int,inc,where,type)

- int - Size of initial stack segment
- inc - Minimum size of any additional stack segments
- where - BELOW, ANYWHERE - Must be BELOW if ALL31 (OFF)
- type
 - ▶ KEEP - do not free when empty (better performance)
 - ▶ FREE - free when empty (better memory management)
- NOTES:
 - ▶ Used for Dynamic Save Area
 - ▶ Used for C/C++ and PL/I local variables
- Recommendations:
 - ▶ CICS - STACK(4K,4080,ANYWHERE,FREE)
 - ▶ Other - STACK(128K,128K,ANYWHERE,KEEP)
 - Use KEEP when ANYWHERE is specified
 - Use FREE when BELOW is specified



► Stack - Same as Heap

Runtime Options

■ Some important runtime options...

■ POSIX(option)

- ON - Indicates POSIX semantics should be used.
- OFF - Indicates ANSI semantics should be used.
- NOTES:
 - ▶ Where ANSI and POSIX conflict this setting is used to clear up any ambiguities
 - ▶ ANSI programs may access the Open Edition HFS without POSIX(ON)
 - ▶ POSIX(ON) must be set to use POSIX only functions like `pthread_create()`.
 - ▶ POSIX(ON) is the default in the USS Shell.
- Recommendations:
 - ▶ CICS - POSIX(OFF)
 - ▶ Other - POSIX(OFF)



- ▶ POSIX - Depends on the audience....
- ▶
- ▶ Use it if you need it.

Runtime Options

■ Some important runtime options...

- STORAGE(getheap,freeheap,stack,reserve allocation)
 - getheap - One byte value used to initialize every heap allocation.
 - ▶ 00 is equivalent to WSCLEAR (COBOL)
 - freeheap - One byte value used to initialize every heap free.
 - ▶ Debug purposes
 - ▶ Security purposes
 - stack - One byte value used to initialize every stack (DSA) allocation
 - reserve allocation
 - ▶ Reserve stack storage (doesn't really belong on this option).
 - Recommendations:
 - ▶ CICS - STORAGE(00,None,00,0K)
 - ▶ Other - STORAGE(None,None,None,8K)
 - Note:
 - ▶ Setting getheap or stack to 00 has been know to correct problems with programs.

- ▶ Storage -
- ▶ Explain Getheap - WSCLEAR
- ▶ Freeheap - Debug/Security
- ▶ Getstack - Each DSA

Runtime Options

■ Some important runtime options...

- TERMTHDACT(option) - The action a thread(task) takes when terminating - Use to get CEEDUMP and/or U4039.
 - QUIET - Messages off
 - MESSAGE - Just messages no dump
 - TRACE - Traceback only (no dump)
 - DUMP - CEEDUMP only
 - UADUMP - CEEDUMP/system dump
 - UAONLY - system dump no CEEDUMP (LE 2.7)
 - UATRACE - system dump and traceback (LE 2.7)
 - UAImm - system dump of the original error (LE 2.7)
 - Requires TRAP(ON,NOSPIE)
 - Takes dump in LE ESTAE via SETRP prior to LE condition handling for program checks only.
 - Recommendations:
 - CICS - TERMTHDACT(MSG)
 - Other - TERMTHDACT(TRACE)



- ▶ TERMTHDACT
- ▶
- ▶ Cover UAONLY/UATRACE/UAImm at first since they did not yet GA.
- ▶
- ▶ Explain what is currently available.
- ▶
- ▶ Explain UAONLY - CICS!!!!
- ▶ UATRACE - Not sure why
- ▶ UAImm - Really cool. Get the original abend and dump!
- ▶ Helps with U4083 and U4087 we talked about earlier.

Runtime Options

■ Some important runtime options...

■ TRAP

- Turn LE condition handling on or off
- TRAP(ON,SPIE) recommended
 - Simply TRAP(ON) prior to LE 1.8
- TRAP(ON,NOSPIE) (LE 1.8)
 - Allows user apps to have their own ESPIE routine
 - LE condition handling will still take place via an ESTAE
 - Required if using TERMTHDACT(UAIMM) (LE 2.7)
- TRAP(OFF)
 - LE condition handling is disabled (almost)
 - SPIE/NOSPIE suboption is ignored
- Recommendations:
 - CICS - TRAP(ON,SPIE)
 - Other - TRAP(ON,SPIE)



- TRAP
-
- Use TRAP(ON)!!!



Hands on debug

With a COBOL example



CEEDUMPs

■ 2 simple(?) COBOL program. (JMONTI.TEST.C(COBOL1))

```
000100 CBL NOLIB,APOST,NODYNAM,NOOPT,TEST
000200     PROCESS QUOTE,MAP
000300     IDENTIFICATION DIVISION.
000400     PROGRAM-ID. COBOLED1.
000500     ENVIRONMENT DIVISION.
000600     DATA DIVISION.
000700     WORKING-STORAGE SECTION.
000800     01 WS-VARS.
000900         05 WS-COMP1 PIC S9(4) COMP-4 VALUE ZEROES.
001000         05 WS-COMP2 PIC S9(4) COMP-4 VALUE ZEROES.
001100         05 WS-COMP3 PIC S9(4) COMP-4 VALUE ZEROES.
001200     PROCEDURE DIVISION.
001300         MOVE 33 TO WS-COMP3.
001400         MOVE 22 TO WS-COMP2.
001500         MOVE 11 TO WS-COMP1.
001600         CALL "COBOLED2".
001700         STOP RUN.
001800     END PROGRAM COBOLED1.
```



- ▶ Explain the program.
- ▶
- ▶ Show #pragma runopts - we show it later in the options format.
- ▶
- ▶ Show divide by 0 in func_with_error()

CEEDUMPs

■ 2 simple(?) COBOL program. (JMONTI.TEST.C(COBOL1))

```
001900      IDENTIFICATION DIVISION.
002000      PROGRAM-ID. COBOLED2.
002100      ENVIRONMENT DIVISION.
002200      DATA DIVISION.
002300      WORKING-STORAGE SECTION.
002400          01 WS-VARS.
002500              05 WS-COMP1 PIC S9(4) COMP-4 VALUE ZEROES.
002600              05 WS-COMP2 PIC S9(4) COMP-4 VALUE ZEROES.
002700              05 WS-COMP3 PIC S9(4) COMP-4 VALUE ZEROES.
002800      PROCEDURE DIVISION.
002900          MOVE 32 TO WS-COMP3.
003000          MOVE 10 TO WS-COMP1.
003100          DIVIDE WS-COMP1 BY WS-COMP2 GIVING WS-COMP3.
003200          STOP RUN.
003300      END PROGRAM COBOLED2.
```



- ▶ Explain the program.
- ▶
- ▶ Show #pragma runopts - we show it later in the options format.
- ▶
- ▶ Show divide by 0 in func_with_error()

CEEDUMPs

■ Job Log

- ▶ Shows U4039 ABEND in CEEPLPKA
- ▶ ABEND=S0C9 reissued at termination

```
-
- STEPNAME PROCSTEP PGMNAME      CC      USED      CPU TIME  ELAPSED TIME
- STEP1      COBOL    IGYCRCTL    00      2048K    00:00:00.16  00:03:51.62
- STEP1      LKED     HEWL        00      204K     00:00:00.08  00:00:01.21
IEA995I SYMPTOM DUMP OUTPUT
USER COMPLETION CODE=4039 REASON CODE=00000000
TIME=21.45.36 SEQ=03447 CPU=0000 ASID=0153
PSW AT TIME OF ERROR 078D1000 A3E207B0 ILC 2 INTC 0D
ACTIVE LOAD MODULE      ADDRESS=23E19D30 OFFSET=00006A80
NAME=CEEPLPKA
DATA AT PSW 23E207AA - 00181610 0A0D58D0 D00498EC
GPR 0-3 84000000 84000FC7 00024478 23E207B0
GPR 4-7 23E178A0 00000000 00024478 00025017
GPR 8-11 23E238A5 23E228A6 000243D0 A3E206E0
GPR 12-15 00015910 00026180 A3E22F1E 00000000
END OF SYMPTOM DUMP
IEA993I SYSMDUMP TAKEN TO JMONTI.LECOBED1.SYSMDUMP
IEF450I JMONTI@ GO - ABEND=S0C9 U0000 REASON=00000009
```



- ▶ Joblog
- ▶
- ▶ Show U4039 and CEEPLPKA - But this is not the error
- ▶ These regs are for SVC D in not the real error!!!!
- ▶
- ▶ We re-issued the ABEND0C9

CEEDUMPs

■ Program Output

```
CEE3209S The system detected a fixed-point divide exception.  
From compile unit COBOLED2 at entry point COBOLED2 at statement 13 at  
compile unit offset +00000308 at address 23E029E0.
```

■ For further diagnosis lets start with the CEEDUMP!!!!

- ▶ CEEDUMPs are formatted dumps and may simply be browsed.
 - SDSF for SYSOUT
 - CEBR CESE
 - Your favorite browser/editor in USS.



- ▶ Read the message!!!! This is the error!
- ▶
- ▶ statement 27 only with the TEST compiler option

CEEDUMPs

■ Sample CEEDUMP: (See handout)

CEE3DMP V1 R8.0: Condition processing resulted in the unhandled condition. 02/26/99 9:48:42 PM Page: 1

Information for enclave COBOLED1

Information for thread 8000000000000000

Traceback:

DSA Addr	Program Unit	PU Addr	PU Offset	Entry	E Addr	E Offset	Statement	Load Mod	Service	Status
00024018	CEEHDSP	23E208A8	+000026A6	CEEHDSP	23E208A8	+000026A6		CEEPLPKA	UQ24548	Call
000260C8	COBOLED2	23E026D8	+00000308	COBOLED2	23E026D8	+00000308	13	COBOL1		Exception
00026018	COBOLED1	23E00978	+0000033E	COBOLED1	23E00978	+0000033E	14	COBOL1		Call

For additional info see LE Debug Guide and Run-time Messages.

- Show
 - Release 1.8
 - Traceback headers
 - STATUS - Why we left the program
- Go through the entire traceback
- EDCZMINV (Main Invocation) out of CEEEV003 (C/C++ Event handler)
- CEEBBEXT - Initialization CEEBINIT

CEEDUMPS

■ Sample CEEDUMP:

```
Condition Information for Active Routines
Condition Information for COBOLED2 (DSA address 000260C8)
CIB Address: 00024478
Current Condition:
    CEE0198S The termination of a thread was signaled due to an unhandled condition.
Original Condition:
    CEE3209S The system detected a fixed-point divide exception.
Location:
    Program Unit: COBOLED2 Entry: COBOLED2 Statement: 13 Offset: +00000308
Machine State:
    ILC..... 0002      Interruption Code..... 0009
    PSW..... 078D2000 A3E029E2
    GPR0..... 00026180 GPR1..... 00000000 GPR2..... 00000000 GPR3..... 0000000A
    GPR4..... 00000000 GPR5..... 00046038 GPR6..... 00000000 GPR7..... 00FCCBF0
    GPR8..... 23F1B100 GPR9..... 23F17700 GPR10..... 23E027E0 GPR11..... 23E028B0
    GPR12..... 23E027D4 GPR13..... 000260C8 GPR14..... A3E029D4 GPR15..... A3E02916
Storage dump near condition, beginning at location: 23E029D0
+000000 23E029D0 45E0913A 48208000 8E200020 48408002 1D244030 800445E0 913A9140 905558B0 [...j.....j.....j.....]
```

- ▶ CEE0198S - Single for termination ignore!
- ▶ Use original condition CEE3209S!
- ▶
- ▶ Location !!!!
- ▶
- ▶ Machine State - ILC PSW REGS
- ▶
- ▶ Storage dump.
- ▶
- ▶ LE knows the rules. Failing instruction always at the break (second set of 4 words)
- ▶
- ▶ 5D60D09C Divide R6 by 9C bytes past R13 - R6 is even odd pair - A = 10. Later in CEEDUMP we could find the storage
- ▶ for 9C past R13 but we got lucky. 1st instruction is 5050D098 and R5 is 0!!!
- ▶
- ▶ CEEDUMPS are great for existing problems. We have failing CSECT, REGS, OFFSET, Instruction. Should be able to find an existing APAR or PDDB item.

CEEDUMPs

■ Sample CEEDUMP:

Parameters, Registers, and Variables for Active Routines:

```
CEEHDSP (DSA address 00024018):
  Saved Registers:
    GPR0..... 000000000  GPR1..... 000243B4  GPR2..... A3E22F52  GPR3..... 00000003
    GPR4..... 23E178A0  GPR5..... 00000000  GPR6..... 00024FA4  GPR7..... 00025017
    GPR8..... 23E238A5  GPR9..... 23E228A6  GPR10..... 23E218A7  GPR11..... 23E208A8
    GPR12..... 00015910  GPR13..... 00024018  GPR14..... 8001C0E2  GPR15..... A3E675A0
  GPREG STORAGE:
    Storage around GPR0 (00000000)
      +0000 00000000  Inaccessible storage.
      +0020 00000020  Inaccessible storage.
      +0040 00000040  Inaccessible storage.
    Storage around GPR1 (000243B4)
      -0020 00024394  23E17934 00025017 23E238A5 23E228A6 23E218A7 23E208A8 00015910 00000000 |.....&..S.v.S.w.S.x.S.y.....|
      +0000 000243B4  00025097 000250E7 00024FA4 00024FA4 00024E04 00024ED4 00025218 00024478 |..&p..&X..|u..|u..+...+M.....|
      +0020 000243D4  00024478 00000000 00000000 00000000 00000001 00000005 A3EEBAA0 00000000 |.....t.....|
  ...
  Local Variables:
    6 01 WS-VARS      AN-GR
    7 02 WS-COMP1     S9999 COMP      +00010
    8 02 WS-COMP2     S9999 COMP      +00000
    9 02 WS-COMP3     S9999 COMP      +00032
  ...
  Program COBOLED2 was compiled 02/26/99 9:45:32 PM
  COBOL Version = 01 Release = 02 Modification = 02      User Level = ' '
  TGT for COBOLED2: 23F17700
```

- We start displaying storage around the registers, local variables for each entry point and then other storage.
- Also the COBOL TGT is displayed.

- We start to display storage....
-
- For C/C++ we show Errno and Errnojr

SYSTEM DUMPS

- Use system dumps (SYSMDUMP/CICS dump) for additional debug
 - Use VERBX CEEERRIP (alias LEDATA) from IPCS - CICS users must specify CAA(xxxxxxxx)
 - ▶ No options (or 'SUMMARY') for general info and runtime options.
 - ▶ 'CEEDUMP' for report like CEEDUMP's traceback information.
 - ▶ 'CM' for condition management
 - Dump title is most often ABENDU4039

TITLE=JOBNAME JMONTI@B STEPNAME GO

USER 4039



- ▶ We need to use a system dump when the CEEDUMP is not sufficient. Use the VERBX
- ▶
- ▶ CICS must specify CAA

SYSTEM DUMPs

■ VERBX LEDATA Example - VERBX CEEERRIP

■ Command: VERBX CEEERRIP

■ Command: VERBX CEEERRIP 'SUMMARY'

```
*****
                        LANGUAGE ENVIRONMENT DATA
*****

TCB: 008ADE88          LE Level: 08          ASID: 0153

Active Members: COBOL

CEECAA: 00015910
+000000 FLAG0:00      LANGP:08      BOS:00026000      EOS:00046000
+000044 TORC:00000000      TOVF:8000F100      ATTN:23E16B10
+00015C HLEXIT:00000000      HOOK:50C0D064 05C058C0 C00605CC
+0001A4 DIMA:0000F316      ALLOC:0700C198      STATE:0700C198
+0001B0 ENTRY:0700C198      EXIT:0700C198      MEXIT:0700C198
+0001BC LABEL:0700C198      BCALL:0700C198      ACALL:0700C198
+0001C8 DO:0700C198      IFTRUE:0700C198      IFFALSE:0700C198
+0001D4 WHEN:0700C198      OTHER:0700C198      CGOTO:0700C198
+0001F4 CRENT:00000000      EDCV:00000000      TCASRV_USERWORD:00000000
+00025C TCASRV_WORKAREA:23E16460      TCASRV_GETMAIN:00000000
+000264 TCASRV_FREEMAIN:00000000      TCASRV_LOAD:8000F840
```

► Show LE level. Not much else is useful

SYSTEM DUMPs

■ VERBX CEEERRIP ...

■ COMMAND: VERBX CEEERRIP continued...

```
CEERCB: 00013918
+000000 EYE:CEERCB      SYSTM:03   HRDWR:03   SBSYS:02   FLAGS:80
+000014 DMEBR:23E09BB8   ZLOD:23ED4320   ZDEL:23E7D108
+000020 ZGETST:23ED5EC8   ZFREEST:23ED3FA0   VERSION_ID:000000B4

CEEEDB: 000148B0
+000000 EYE:CEEEDB      FLAG1:D0    BIPM:00    BPM:00
+00000B CREATOR_ID:01    MEMBR:000157D0   OPTCB:00014F00
+000014 URC:00000000    RSNCD:00000000   DBGEH:00000000
+000020 BANHP:00014D78   BBEHP:00014DA8   BCELV:0001B038
+00002C PCB:00014558     ELIST:00000000   PL_ASTRPTR:00000000
+000038 DEFPLPTR:000149D0    CXIT_PAGE:00000000
+000040 DEBUG_TERMID:00000000   PARENT:00000000   R13_PARENT:00005F88
+000054 LEOV:00000000    ENVAR:23E16708   ENVIRON:00014908
+000064 OTRB:00000000    PSA31:00000000   PSL31:00000000
+000070 PSA24:00000000    PSL24:00000000   PSRA:23ED3DB0
+00007C CAACHAIN@:00015910    FLAG1A:70    MEMBERCOMPAT1:00
+000090 THREADSACTIVE:00000001    CURMSGFILEDCBPTR:00013B80
+000098 CEEINT_INPUT_R1:00005FD8    LAST_RBADDR:008ADD78
+0000A0 LAST_RBCNT:00000001
```

► PARENT - points to previous CAA in nested enclaves.

► PARENT in EDB may be used to find other CAAs

SYSTEM DUMPs

■ VERBX CEEERRIP ... (COMMAND: VERBX CEEERRIP)

```
LAST WHERE SET      Override  OPTIONS
*****
INSTALLATION DEFAULT OVR      ABPERC(NONE)
PROGRAM INVOCATION   OVR      ABTERMENC(ABEND)
INSTALLATION DEFAULT OVR      NOAIXBLD
INSTALLATION DEFAULT OVR      ALL31(OFF)
INSTALLATION DEFAULT OVR      ANYHEAP(00016384,00008192,ANY ,FREE)
INSTALLATION DEFAULT OVR      NOAUTOTASK
INSTALLATION DEFAULT OVR      BELOWHEAP(00008192,00004096,FREE)
INSTALLATION DEFAULT OVR      CELOPTS(ON)
INSTALLATION DEFAULT OVR      CBLPSHPOP(ON)
INSTALLATION DEFAULT OVR      CBLQDA(ON)
INSTALLATION DEFAULT OVR      CHECK(ON)
INSTALLATION DEFAULT OVR      COUNTRY(US)
INSTALLATION DEFAULT OVR      DEBUG
INSTALLATION DEFAULT OVR      DEPTHCONDLMT(00000010)
INSTALLATION DEFAULT OVR      ENVAR(" ")
INSTALLATION DEFAULT OVR      ERRCOUNT(00000020)
INSTALLATION DEFAULT OVR      ERRUNIT(00000006)
INSTALLATION DEFAULT OVR      FILEHIST
DEFAULT SETTING      OVR      NOFLOW
INSTALLATION DEFAULT OVR      HEAP(00032768,00032768,ANY ,
                                KEEP,00008192,00004096)
INSTALLATION DEFAULT OVR      HEAPCHK(OFF,00000001,00000000)
```

► Runtime options:

►

► ABPERC(NONE) was an installation default

► ABTERMENC(ABEND) I specified in my JCL on EXEC card as part of PARM= - Therefore PROGRAM INVOCATION

SYSTEM DUMPs

■ VERBX CEEERRIP ... (COMMAND: VERBX CEEERRIP)

```
INSTALLATION  DEFAULT  OVR      HEAPPOOLS(OFF,
                                00000008,00000010,
                                00000032,00000010,
                                00000128,00000010,
                                00000256,00000010,
                                00001024,00000010,
                                00002048,00000010)

REGION_DEFAULT  OVR      INFOMSGFILTER(OFF)
INSTALLATION  DEFAULT  OVR      INQPCOPN
INSTALLATION  DEFAULT  OVR      INTERRUPT(OFF)
INSTALLATION  DEFAULT  OVR      LIBRARY(SYSCEE)
INSTALLATION  DEFAULT  OVR      LIBSTACK(00008192,00004096,FREE)
INSTALLATION  DEFAULT  OVR      MSGFILE(SYSOUT ,FBA ,00000121,00000000,
                                NOENQ)
INSTALLATION  DEFAULT  OVR      MSGQ(00000015)
INSTALLATION  DEFAULT  OVR      NATLANG(ENU)
INSTALLATION  DEFAULT  OVR      NONONIPSTACK(00004096,00004096,BELOW,KEEP)
INSTALLATION  DEFAULT  OVR      OCSTATUS
INSTALLATION  DEFAULT  OVR      NOPC
INSTALLATION  DEFAULT  OVR      PLITASKCOUNT(00000020)
INSTALLATION  DEFAULT  OVR      POSIX(OFF)
INSTALLATION  DEFAULT  OVR      PROFILE(OFF," ")
INSTALLATION  DEFAULT  OVR      PRTUNIT(00000006)
INSTALLATION  DEFAULT  OVR      PUNUNIT(00000007)
```

► Not much here, just the options

SYSTEM DUMP_s

■ VERBX CEEERRIP ... (COMMAND: VERBX CEEERRIP)

```
INSTALLATION DEFAULT OVR RDRUNIT(00000005)
INSTALLATION DEFAULT OVR RECPAD(OFF)
INSTALLATION DEFAULT OVR RPTOPTS(OFF)
INSTALLATION DEFAULT OVR RPTSTG(OFF)
INSTALLATION DEFAULT OVR NORTEREUS
INSTALLATION DEFAULT OVR RTLS(OFF)
INSTALLATION DEFAULT OVR NOSIMVRD
INSTALLATION DEFAULT OVR STACK(00131072,00131072,BELOW,KEEP)
INSTALLATION DEFAULT OVR STORAGE(NONE,NONE,NONE,00008192)
PROGRAM INVOCATION OVR TERMTHDACT(UADUMP)
INSTALLATION DEFAULT OVR NOTEST(ALL,*,PROMPT,INSPREF)
INSTALLATION DEFAULT OVR THREADHEAP(00004096,00004096,ANY,KEEP)
INSTALLATION DEFAULT OVR TRACE(OFF,00004096,DUMP,LE=00000000)
INSTALLATION DEFAULT OVR TRAP(ON,SPIE)
INSTALLATION DEFAULT OVR UPSI(00000000)
INSTALLATION DEFAULT OVR NOUSRHDLR()
INSTALLATION DEFAULT OVR VCTRSAVE(OFF)
INSTALLATION DEFAULT OVR VERSION()
INSTALLATION DEFAULT OVR XUFLOW(AUTO)
*****
```

- ▶ Again I specified TERMTHDACT(UADUMP) in JCL.
- ▶
- ▶ TRAP(ON,SPIE) says Programmer Default, Why?
- ▶ #pragma runopts()

SYSTEM DUMP_s

■ VERBX CEEERRIP 'CEEDUMP'

■ COMMAND: VERBX CEEERRIP 'CEEDUMP'

```
*****
                        LANGUAGE ENVIRONMENT DATA
*****

Information for enclave COBOLED1

Information for thread 8000000000000000

Traceback:
  DSA Addr  Program Unit  PU Addr  PU Offset  Entry      E Addr  E Offset  Statement  Load Mod  Service  Status
00026180   CEEHSDMP      23E206E0 +00089AD6 CEEHSDMP   23E206E0 +00089AD6
00024018   CEEHDSP       23E208A8 +00002674 CEEHDSP    23E208A8 +00002674
000260C8   COBOLED2      23E026D8 +00000308 COBOLED2   23E026D8 +00000308 Exception
00026018   COBOLED1      23E00978 +0000033E COBOLED1   23E00978 +0000033E Call
```

- ▶ Not really a CEEDUMP.
- ▶ Missing some info - we've requested it.

SYSTEM DUMPs

■ VERBX CEEERRIP 'CM'

■ COMMAND: VERBX CEEERRIP 'CM'

— (First CIBH is zeroes and should be ignored)

Condition Management Control Blocks

```
HCOM: 23E16AC8
+000008 EYES:HCOM FLAG:60104000 CIBH:000248C0

CIBH: 000248C0
+000000 EYE:CIBH BACK:23E178A0 FRWD:00000000
+000010 PTR_CIB:00000000 FLAG1:00 ERROR_LOCATION_FLAGS:00
+000018 HDLQ:00000000 STATE:00000000 PRM_DESC:00000000
+000024 PRM_PREFIX:00000000
+000028 PRM_LIST:00000000 00000000 00000000 00000000
+000038 PARM_DESC:00000000 PARM_PREFIX:00000000
+000040 PARM_LIST:00000000 00000000 00000000 00000000 FUN:00000000
+000058 FLG_5:00 FLG_6:00 FLG_7:00 FLG_8:00 FLG_1:00
+00005D FLG_2:00 FLG_3:00 FLG_4:00 ABCD:00000000
+000064 ABRC:00000000 OLD_COND_64:00000000 00000000
+000070 OLD_MIB:00000000 COND_64:00000000 00000000
+00007C MIB:00000000 PL:00000000 SV2:00000000
+000088 SV1:00000000 INT:00000000 MID:00000000
+000094 HDL_SF:00000000 HDL_EPT:00000000 HDL_RST:00000000
+0000A0 RSM_SF:00000000 RSM_POINT:00000000 RSM_MACHINE:00000000
```

► Ignore the first one. It is waiting for the next condition

SYSTEM DUMPs

■ VERBX CEEERRIP 'CM'... COMMAND: VERBX CEEERRIP 'CM'

— (Second CIBH contains exception information)

```
CIBH: 23E178A0
+000000 EYE:CIBH BACK:00000000 FRWD:000248C0
+000010 PTR_CIB:00024478 FLAG1:C5 ERROR_LOCATION_FLAGS:1F
+000018 HDLQ:00000000 STATE:00000008 PRM_DESC:00000000
+000024 PRM_PREFIX:00000000
+000028 PRM_LIST:00024490 00024558 00024564 23E17CBC
+000038 PARM_DESC:00000000 PARM_PREFIX:00000000
+000040 PARM_LIST:00024554 00024478 00024564 23E17CBC FUN:00000067
+000058 FLG_5:48 FLG_6:23 FLG_7:00 FLG_8:00 FLG_1:00
+00005D FLG_2:00 FLG_3:00 FLG_4:05 ABCD:940C9000
+000064 ABRC:00000009 OLD_COND_64:00030C89 59C3C5C5
+000070 OLD_MIB:00000001 COND_64:00030C89 59C3C5C5
+00007C MIB:00000001 PL:23E026EC SV2:000260C8
+000088 SV1:000260C8 INT:23E029E0 MID:00000005
+000094 HDL_SF:000161A8 HDL_EPT:A3EEBAA0 HDL_RST:00000000
+0000A0 RSM_SF:000260C8 RSM_POINT:23E029E2 RSM_MACHINE:23E17BE8
+0000B0 COND_DEFAULT:00000005 Q_DATA_TOKEN:23E179D8 FDBK:00000000
+0000BC ABNAME:.....
```

■ OLD_COND_64: 00030C89 59C3C5C5 is an LE condition token

- CEE3209S

- ▶ This one is good.
- ▶
- ▶ +5D line has ABCD
- ▶
- ▶ Then OLD_COND_64 - Break down the token.

SYSTEM DUMPs

- VERBX CEEERRIP 'CM'... COMMAND: VERBX CEEERRIP 'CM'
- (Then machine state information like CEEDUMP)

```
Machine State
+000278 MCH_EYE:ZMCH      MCH_GPR00:00026180      MCH_GPR01:00000000
+000288 MCH_GPR02:00000000      MCH_GPR03:0000000A
+000290 MCH_GPR04:00000000      MCH_GPR05:00046038
+000298 MCH_GPR06:00000000      MCH_GPR07:00FCCBF0
+0002A0 MCH_GPR08:23F1B100      MCH_GPR09:23F17700
+0002A8 MCH_GPR10:23E027E0      MCH_GPR11:23E028B0
+0002B0 MCH_GPR12:23E027D4      MCH_GPR13:000260C8
+0002B8 MCH_GPR14:A3E029D4      MCH_GPR15:A3E02916
+0002C0 MCH_PSW:078D2000 A3E029E2      MCH_ILC:0002      MCH_IC1:00
+0002CB MCH_IC2:09      MCH_PFT:00000000      MCH_FLT_0:00000000 00000000
+0002D8 MCH_FLT_2:00000000 00000000      MCH_FLT_4:00000000 00000000
+0002E8 MCH_FLT_6:00000000 00000000      MCH_EXT:00000000
```

- Machine State (ZMCH)
- Floating points too

SYSTEM DUMPs

■ VERBX CEEERRIP 'CM'... COMMAND: VERBX CEEERRIP 'CM'
— (and CIB)

```
CIB: 00024478
+000000 EYE:CIB BACK:00000000 FRWD:00000000
+000010 PLAT_ID:00000000 COND_64:000300C6 59C3C5C5
+000020 MIB:00000000 MACHINE:00024ED4
+000028 OLD_COND_64:00030C89 59C3C5C5 OLD_MIB:00000001
+000034 FLG_1:00 FLG_2:00 FLG_3:00 FLG_4:04 HDL_SF:00026018
+00003C HDL_EPT:A3EEBAA0 HDL_RST:00000000 RSM_SF:000260C8
+000048 RSM_POINT:23E029E2 RSM_MACHINE:23E17BE8
+000050 COND_DEFAULT:00000005 VSR:00000000 00000000 VSTOR:00000000
+00009C VRPSA:00000000 MCB:00000000 MRN:00000000 00000000
+0000AC MFLAG:00 FLG_5:48 FLG_6:23 FLG_7:00 FLG_8:00
+0000B4 ABCD:940C9000 ABRC:00000009 ABNAME:00000000 00000000
+0000C4 PL:23E026EC SV2:000260C8 SV1:000260C8
+0000D0 INT:23E029E0 Q_DATA_TOKEN:00000000 FDBK:00000000
+0000DC FUN:00000067 TOKE:00026018 MID:00000005
+0000E8 STATE:00000008 RTCC:00000014 PPAV:00000001
+0000F4 AB_TERM_EXIT:00000000 00000000 SDWA_PTR:00000000
+000100 SIGNO:00000000 PPSD:23E17CCC
```

- NOTE: There may be additional CIBH/CIB pairs for prior conditions.

- ▶ CIB
- ▶ COND_64 not the CEE3209S what is it?
- ▶ CEE0198S - current condition in CEEDUMP....

SYSTEM DUMPs

- Getting the name of the failing csect (function) (or any LE enabled csect).
 - From option I (Browse) in IPCS
 1. Get address (2nd word) from MCH_PSW
 2. L xxxxxxxx (where xxxxxxxx is the addr).
 3. F CEE prev
 4. Back up 5 bytes to the 47F0xxxx instruction
 - a. This is the beginning of the csect/function
 - b. Repeat Step 3 and 4 as needed.
 5. Add the value at offset x'0C' to the module address
 6. Go to that location
 7. First byte is offset to the name. Go there.
 8. This is a 2byte prefixed string with the function name

► Read through this. Explain that it still may be confusing.

SYSTEM DUMPs

■ Getting the name of the failing csect (function)

■ Example: I. MCH_PSW: 078D2000 A3E029E2 (or any address)

```
23E026D0  98ECD00C  07FE0000  47F0F028  00C3C5C5  | g.}.....00..CEE |
                                     3. Do a FIND CEE PREV
                                     4. Make sure 5 bytes prior is 47F0xxxx (this is beginning of module)
23E026E0  000000B8  00000014  47F0F001  98CEAC00  | .....00.q... |
                                     5. GO to offset x'C' and pick up offset of PPA1 - (23E026D8 + 14 = 23E02708)
                                     NOTE: Offset might be negative
```

that address.

6. Add the offset to the beginning of the module and go to

7. Use the first byte as an offset to the name (sometimes

```
doubled)
23E026F0  23E0278E  00000000  00000000  00000000  | .\..... |
23E02700  90ECD00C  4110F038  98EFF04C  07FF0000  | ..}...0.q.0<... |
23E02710  23E026D8  00000000  23E04118  23E02786  | .\Q.....\...f |
23E02720  23E026D8  23E0290A  23E044A0  23E027A2  | .\Q.....\...s |
23E02730  00104001  00000008  C3D6C2D6  D3C5C4F2  | .. ..COBOLED2 |
23E02740  F1F9F9F9  F0F2F2F6  F2F1F4F5  F3F2F0F1  | 1999022621453201 |
23E02750  F0F2F0F2  00000000  0000076C  A0687CC4  | 0202.....%..@D |
23E02760  00000000  00040008  00200000  00000000  | ..... |
23E02770  00000000  00000004  00000004  0000F000  | .....0. |
23E02780  40404040  0008C3D6  C2D6D3C5  C4F20500  | ..COBOLED2.. |
```

8. A 2 byte prefixed string with the csect name.

```
.
.
23E029D0  45E0913A  48208000  8E200020  48408002  | .\j..... |
23E029E0  1D244030  800445E0  913A9140  905558B0  | .. ..\j.j .... |
                2. Display the address (23E029E2)
23E029F0  C00847E0  B1544110  00085820  905C58F0  | {..\.....*.0 |
```

► Go through these steps. They should get it now!

SYSTEM DUMPs

■ Other LEDATA (CEEERRIP) options

- 'SM' - VERBX CEEERRIP 'SM'
 - ▶ All Storage Management control blocks
- 'HEAP' - VERBX CEEERRIP 'HEAP'
 - ▶ Just HEAP Storage Management control blocks
- 'STACK' - VERBX CEEERRIP 'STACK'
 - ▶ Just STACK Storage Management control blocks
- 'ALL' - VERBX CEEERRIP 'ALL'
 - ▶ All control blocks including CRTL
- 'CAA(xxxxxxxx)' - Combined with other options
 - ▶ Allows specific CAA to be used as anchor (required for CICS) - VERBX CEEERRIP 'ALL CAA(xxxx)'
- 'DSA(xxxxxxxx)' - Combined with other options
 - ▶ Allows specific DSA to be used for traceback info
- 'TCB(xxxxxxxx)' - Combined with other options
 - ▶ Allows specific TCB to be used as starting point

- ▶ Other options
- ▶ STACK and HEAP in part 2 - It pays to advertise



Hands on debug

With a C example



CEEDUMPs

■ A simple(?) C program.(JMONTI.TEST.C(TESTEDU))

```
000001 #pragma runopts(trap(on))          /* Set TRAP on as PROGRAMMER DEFAULT */
000002 int first_func_called();              /* Function prototypes */
000003 int second_func_called();
000004 int func_with_error();
000005 int main(int argc, char *argv ) /* Main Program */
000006 {
000007     int i = 0;
000008     i = first_func_called();             /* Call to "first_func_called()" */
000009     return i;
000010 }
000011 int first_func_called()                 /* Function: first_func_called() */
000012 {
000013     int j;
000014     j = second_func_called();            /* Call to "second_func_called()" */
000015     return j;
000016 }
000017 int second_func_called()                 /* Function: second_func_called() */
000018 {
000019     int k;
000020     k = func_with_error();                /* Call to "func_with_error()" */
000021     return k;
000022 }
000023 int func_with_error()                   /* Function: func_with_error() */
000024 {
000025     int n = 0;
000026     int m;
000027     m = 10 / n;                           /* Force divide by 0 (ABEND0C9) */
000028     return m;
000029 }
```



- Explain the program.
-
- Show #pragma runopts - we show it later in the options format.
-
- Show divide by 0 in func_with_error()

CEEDUMPs

■ Job Log

- ▶ Shows U4039 ABEND in CEEPLPKA
- ▶ ABEND=S0C9 reissued at termination

```
-
- STEPNAME PROCSTEP PGMNAME CC USED CPU TIME ELAPSED TIME
- CLG COMPILE CBCDRVR 00 176K 00:00:00.21 00:00:05.31
- CLG LKED HEWL 00 216K 00:00:00.18 00:00:01.34
IEA995I SYMPTOM DUMP OUTPUT
USER COMPLETION CODE=4039 REASON CODE=00000000
TIME=13.35.27 SEQ=22849 CPU=0000 ASID=00F3
PSW AT TIME OF ERROR 078D1400 92B7FA1A ILC 2 INTC 0D
ACTIVE LOAD MODULE ADDRESS=12B19AB8 OFFSET=00065F62
NAME=CEEPLPKA
DATA AT PSW 12B7FA14 - 00181610 0A0D58D0 D00498EC
GPR 0-3 84000000 84000FC7 00020478 12B7FA1A
GPR 4-7 12B148A0 00000000 000203D0 00021017
GPR 8-11 12B2A135 12B29136 000203D0 92B7F950
GPR 12-15 00015910 00022460 92B297DE 00000000
END OF SYMPTOM DUMP
IEA993I SYSMDUMP TAKEN TO JMONTI.LEEDU.SYSMDUMP
IEF450I JMONTI@C GO CLG - ABEND=S0C9 U0000 REASON=00000009
TIME=13.36.13
- CLG GO PGM=*.DD *S0C9 296K 00:00:21.68 00:00:48.06
IEF404I JMONTI@C - ENDED - TIME=13.36.13
```



▶ Joblog

▶

▶ Show U4039 and CEEPLPKA - But this is not the error

▶ These regs are for SVC D in not the real error!!!!

▶

▶ We re-issued the ABEND0C9

CEEDUMPs

■ Program Output

```
CEE3209S The system detected a fixed-point divide exception.  
          From compile unit JMONTI.TEST.C(TESTEDU) at entry point func_with_error  
          at statement 27 at compile unit offset +00000062 at address 12B00BE2.
```

■ For further diagnosis lets start with the CEEDUMP!!!!

- ▶ CEEDUMPs are formatted dumps and may simply be browsed.
 - SDSF for SYSOUT
 - CEBR CESE
 - Your favorite browser/editor in USS.



- ▶ Read the message!!!! This is the error!
- ▶
- ▶ statement 27 only with the TEST compiler option

CEEDUMPs

■ Sample CEEDUMP: (See handout)

CEE3DMP V1 R8.0: Condition processing resulted in the unhandled condition.

09/17/98 1:35:27 PM

Page: 1

Information for enclave main

Information for thread 8000000000000000

Traceback:

DSA Addr	Program Unit	PU Addr	PU Offset	Entry	E Addr	E Offset	Statement	Load Mod	Service	Status
00020018	CEEHDSF	12B27138	+0000264E	CEEHDSF	12B27138	+0000264E		CEEPLPKA	UQ19695	Call
000223C0	JMONTI.TEST.C(TESTEDU)	12B00B80	+00000062	func_with_error	12B00B80	+00000062	27 GO			Exception
00022320	JMONTI.TEST.C(TESTEDU)	12B00A00	+0000005E	second_func_called	12B00A00	+0000005E	20 GO			Call
00022280	JMONTI.TEST.C(TESTEDU)	12B00880	+0000005E	first_func_called	12B00880	+0000005E	14 GO			Call
000221E0	JMONTI.TEST.C(TESTEDU)	12B005E8	+00000066	main	12B005E8	+00000066	8 GO			Call
000220C8		12D5BE0E	-12D45A70	EDCZMINV	12D5BE0E	-12D45A70		CEEV003		Call
00022018	CEEBBEXT	0000E690	+0000013C	CEEBBEXT	0000E690	+0000013C		CEEBINIT	UQ09246	Call

For additional info see LE Debug Guide and Run-time Messages.

- Show
 - Release 1.8
 - Traceback headers
 - STATUS - Why we left the program
- Go through the entire traceback
- EDCZMINV (Main Invocation) out of CEEV003 (C/C++ Event handler)
- CEEBBEXT - Initialization CEEBINIT

CEEDUMPs

■ Sample CEEDUMP:

```
Condition Information for Active Routines
Condition Information for JMONTI.TEST.C(TESTEDU) (DSA address 000223C0)
CIB Address: 00020478
Current Condition:
  CEE0198S The termination of a thread was signaled due to an unhandled condition.
Original Condition:
  CEE3209S The system detected a fixed-point divide exception.
Location:
  Program Unit: JMONTI.TEST.C(TESTEDU)
  Entry:      func_with_error
  Statement:   27 Offset: +00000062
Machine State:
  ILC..... 0004      Interruption Code..... 0009
  PSW..... 078D2400 92B00BE6
  GPR0..... 00022460 GPR1..... 12E77040 GPR2..... 92D5BEC2 GPR3..... 92B00BCA
  GPR4..... 8000E774 GPR5..... 00000000 GPR6..... 00000000 GPR7..... 0000000A
  GPR8..... 00000001 GPR9..... 80000000 GPR10..... 92D5BE02 GPR11..... 8000E690
  GPR12..... 00015910 GPR13..... 000223C0 GPR14..... 000223C0 GPR15..... 12B00B80
Storage dump near condition, beginning at location: 12B00BD2
+000000 12B00BD2 5050D098 4400C1AC 4160000A 8E600020 5D60D09C 5070D09C 4400C1AC 58F0D09C |&&.q..A..-...-.)-.q&.....A..0..|
```

- ▶ CEE0198S - Single for termination ignore!
- ▶ Use original condition CEE3209S!
- ▶
- ▶ Location !!!!
- ▶
- ▶ Machine State - ILC PSW REGS
- ▶
- ▶ Storage dump.
- ▶
- ▶ LE knows the rules. Failing instruction always at the break (second set of 4 words)
- ▶
- ▶ 5D60D09C Divide R6 by 9C bytes past R13 - R6 is even odd pair - A = 10. Later in CEEDUMP we could find the storage
- ▶ for 9C past R13 but we got lucky. 1st instruction is 5050D098 and R5 is 0!!!
- ▶
- ▶ CEEDUMPS are great for existing problems. We have failing CSECT, REGS, OFFSET, Instruction. Should be able to find an existing APAR or PDDB item.

CEEDUMPs

■ Sample CEEDUMP:

```
Parameters, Registers, and Variables for Active Routines:
CEEHDSP (DSA address 00020018):
  Saved Registers:
    GPR0..... 00000000  GPR1..... 000203B4  GPR2..... 92B2978A  GPR3..... 00000003
    GPR4..... 12B148A0  GPR5..... 00000000  GPR6..... 00017038  GPR7..... 00021017
    GPR8..... 12B2A135  GPR9..... 12B29136  GPR10..... 12B28137  GPR11..... 12B27138
    GPR12..... 00015910  GPR13..... 00020018  GPR14..... 800180E2  GPR15..... 92B5CF00
  GPREG STORAGE:
    Storage around GPR0 (00000000)
      +0000 00000000  Inaccessible storage.
      +0020 00000020  Inaccessible storage.
      +0040 00000040  Inaccessible storage.
    Storage around GPR1 (000203B4)
      -0020 00020394  12B14934 00021017 12B2A135 12B29136 12B28137 12B27138 00015910 00000000 |.....j...a.....|
      +0000 000203B4  00021097 000210E7 00020FA4 00020FA4 00020E04 00020ED4 00021218 00000000 |...p...X...u...u...M.....|
      +0020 000203D4  00020478 00000000 00000000 00000000 00000001 00000001 12BA0F00 00000000 |.....|
    .
    .
  Additional Language Specific Information:

    errno information :
    Thread Id .... 8000000000000000 Errno ..... 0 Errnojr .... 00000000
```

- We start displaying storage around the registers and then other storage.
- Finally under C/C++ we display the errno and errnojr information (if any)

- We start to display storage....
-
- For C/C++ we show Errno and Errnojr

SYSTEM DUMPS

- Use system dumps (SYSMDUMP/CICS dump) for additional debug
 - Use VERBX CEEERRIP (alias LEDATA) from IPCS - CICS users must specify CAA(xxxxxxxx)
 - ▶ No options (or 'SUMMARY') for general info and runtime options.
 - ▶ 'CEEDUMP' for report like CEEDUMP's traceback information.
 - ▶ 'CM' for condition management
 - Dump title is most often ABENDU4039

TITLE
LIST 00000000 LITERAL LENGTH(54) CHARACTER
00000000 | JOBNAME JMONTI@C STEPNAME GO CLG USER 4039



- ▶ We need to use a system dump when the CEEDUMP is not sufficient. Use the VERBX
- ▶
- ▶ CICS must specify CAA

SYSTEM DUMPs

■ VERBX LEDATA Example - VERBX CEEERRIP

- Command: VERBX CEEERRIP
- Command: VERBX CEEERRIP 'SUMMARY'

```
*****
                        LANGUAGE ENVIRONMENT DATA
*****

TCB: 008BBD90          LE Level: 08          ASID: 00F3

Active Members: C/C++

CEECAA: 00015910
+000000 FLAG0:00      LANGP:08      BOS:00022000      EOS:00042000
+000044 TORC:00000000      TOVF:8000F008      ATTN:12B13B10
+00015C HLLEXIT:00000000      HOOK:50C0D064 05C058C0 C00605CC
+0001A4 DIMA:0000F21E      ALLOC:0700C198      STATE:0700C198
+0001B0 ENTRY:0700C198      EXIT:0700C198      MEXIT:0700C198
+0001BC LABEL:0700C198      BCALL:0700C198      ACALL:0700C198
+0001C8 DO:0700C198      IFTRUE:0700C198      IFFALSE:0700C198
+0001D4 WHEN:0700C198      OTHER:0700C198      CGOTO:0700C198
+0001F4 CRENT:00000000      EDCV:92BEAED8      TCASRV_USERWORD:00000000
+00025C TCASRV_WORKAREA:12B13460      TCASRV_GETMAIN:00000000
+000264 TCASRV_FREEMAIN:00000000      TCASRV_LOAD:8000F748
```

► Show LE level. Not much else is useful

SYSTEM DUMPs

■ VERBX CEEERRIP ...

■ COMMAND: VERBX CEEERRIP continued...

```
CEERCB: 00013918
+000000 EYE:CEERCB          SYSTM:03   HRDWR:03   SBSYS:02   FLAGS:80
+000014 DMEMBR:12B06BB8     ZLOD:12BD2268   ZDEL:12B78BD8
+000020 ZGETST:12BD3E10     ZFREEST:12BD1EE8   VERSION_ID:000000B4

CEEEDB: 000148B0
+000000 EYE:CEEEDB          FLAG1:D0    BIPM:00    BPM:00
+00000B CREATOR_ID:01      MEMBR:000157D0   OPTCB:00014F00
+000014 URC:00000000       RSNCD:00000000   DBGEH:00000000
+000020 BANHP:00014D78     BBEHP:00014DA8   BCELV:00017038
+00002C PCB:00014558       ELIST:00000000   PL_ASTRPTR:00000000
+000038 DEFPLPTR:000149D0   CXIT_PAGE:00000000
+000040 DEBUG_TERMID:00000000 PARENT:00000000 R13_PARENT:00005F88
+000054 LEOV:00000000       ENVAR:12B13708   ENVIRON:00014908
+000064 OTRB:00000000       PSA31:12B18BE4   PSL31:00000000
+000070 PSA24:000166C0     PSL24:00000000   PSRA:12BD1CF8
+00007C CAACHAIN@:00015910     FLAG1A:40    MEMBERCOMPAT1:00
+000090 THREADSACTIVE:00000001 CURMSGFILEDCBPTR:00013B80
+000098 CEEINT_INPUT_R1:00005FD8 LAST_READDR:008BBC30
+0000A0 LAST_RBCNT:00000001
```

► PARENT - points to previous CAA in nested enclaves.

► PARENT in EDB may be used to find other CAAs

SYSTEM DUMPs

■ VERBX CEEERRIP ... (COMMAND: VERBX CEEERRIP)

```
LAST WHERE SET      Override  OPTIONS
*****
INSTALLATION DEFAULT OVR      ABPERC(NONE)
PROGRAM INVOCATION  OVR      ABTERMENC(ABEND)
INSTALLATION DEFAULT OVR      NOAIXBLD
INSTALLATION DEFAULT OVR      ALL31(OFF)
INSTALLATION DEFAULT OVR      ANYHEAP(00016384,00008192,ANY ,FREE)
INSTALLATION DEFAULT OVR      NOAUTOTASK
INSTALLATION DEFAULT OVR      BELOWHEAP(00008192,00004096,FREE)
INSTALLATION DEFAULT OVR      CBLOPTS(ON)
INSTALLATION DEFAULT OVR      CBLPSHPOP(ON)
INSTALLATION DEFAULT OVR      CBLQDA(ON)
INSTALLATION DEFAULT OVR      CHECK(ON)
INSTALLATION DEFAULT OVR      COUNTRY(US)
INSTALLATION DEFAULT OVR      DEBUG
INSTALLATION DEFAULT OVR      DEPTHCONDLMT(00000010)
INSTALLATION DEFAULT OVR      ENVAR(" ")
INSTALLATION DEFAULT OVR      ERRCOUNT(00000020)
INSTALLATION DEFAULT OVR      ERRUNIT(00000006)
INSTALLATION DEFAULT OVR      FILEHIST
DEFAULT SETTING     OVR      NOFLOW
INSTALLATION DEFAULT OVR      HEAP(00032768,00032768,ANY ,
                                KEEP,00008192,00004096)
INSTALLATION DEFAULT OVR      HEAPCHK(OFF,00000001,00000000)
```

- ▶ Runtime options:
- ▶
- ▶ ABPERC(NONE) was an installation default
- ▶ ABTERMENC(ABEND) I specified in my JCL on EXEC card as part of PARM= - Therefore PROGRAM INVOCATION

SYSTEM DUMPs

■ VERBX CEEERRIP ... (COMMAND: VERBX CEEERRIP)

```
INSTALLATION  DEFAULT  OVR      HEAPPOOLS(OFF,
                                00000008,00000010,
                                00000032,00000010,
                                00000128,00000010,
                                00000256,00000010,
                                00001024,00000010,
                                00002048,00000010)
                                OVR
                                INFOMSGFILTER(OFF)
INSTALLATION  DEFAULT  OVR      INQPCOPN
INSTALLATION  DEFAULT  OVR      INTERRUPT(OFF)
INSTALLATION  DEFAULT  OVR      LIBRARY(SYSCEE)
INSTALLATION  DEFAULT  OVR      LIBSTACK(00008192,00008192,FREE)
INSTALLATION  DEFAULT  OVR      MSGFILE(SYSOUT  ,FBA ,00000121,00000000)
INSTALLATION  DEFAULT  OVR      MSGQ(00000015)
INSTALLATION  DEFAULT  OVR      NATLANG(ENU)
INSTALLATION  DEFAULT  OVR      NONONIPSTACK(00004096,00004096,BELOW,KEEP)
INSTALLATION  DEFAULT  OVR      OCSTATUS
INSTALLATION  DEFAULT  OVR      NOPC
INSTALLATION  DEFAULT  OVR      PLITASKCOUNT(00000020)
INSTALLATION  DEFAULT  OVR      POSIX(OFF)
INSTALLATION  DEFAULT  OVR      PROFILE(OFF," ")
INSTALLATION  DEFAULT  OVR      PRTUNIT(00000006)
INSTALLATION  DEFAULT  OVR      PUNUNIT(00000007)
INSTALLATION  DEFAULT  OVR      RDRUNIT(00000005)
```

► Not much here, just the options

SYSTEM DUMP_s

■ VERBX CEEERRIP ... (COMMAND: VERBX CEEERRIP)

```
INSTALLATION DEFAULT OVR RECPAD(OFF)
INSTALLATION DEFAULT OVR RPTOPTS(OFF)
INSTALLATION DEFAULT OVR RPTSTG(OFF)
INSTALLATION DEFAULT OVR NORTEREUS
INSTALLATION DEFAULT OVR RTLS(OFF)
INSTALLATION DEFAULT OVR NOSIMVRD
INSTALLATION DEFAULT OVR STACK(00131072,00131072,BELOW,KEEP)
INSTALLATION DEFAULT OVR STORAGE(NONE,NONE,NONE,00008192)
PROGRAM INVOCATION OVR TERMTHDACT(UADUMP)
INSTALLATION DEFAULT OVR NOTEST(ALL,*,PROMPT,INSPREF)
INSTALLATION DEFAULT OVR THREADHEAP(00004096,00004096,ANY,KEEP)
INSTALLATION DEFAULT OVR TRACE(OFF,00004096,DUMP,LE=00000000)
PROGRAMMER DEFAULT OVR TRAP(ON,SPIE)
INSTALLATION DEFAULT OVR UPSI(00000000)
INSTALLATION DEFAULT OVR NOUSRHDLR( )
INSTALLATION DEFAULT OVR VCTRSVE(OFF)
INSTALLATION DEFAULT OVR VERSION( )
INSTALLATION DEFAULT OVR XUFLOW(AUTO)
*****
```

- ▶ Again I specified TERMTHDACT(UADUMP) in JCL.
- ▶
- ▶ TRAP(ON,SPIE) says Programmer Default, Why?
- ▶ #pragma runopts()

SYSTEM DUMPs

■ VERBX CEEERRIP 'CEEDUMP'

■ COMMAND: VERBX CEEERRIP 'CEEDUMP'

Information for enclave main

Information for thread 8000000000000000

Traceback:

DSA Addr	Program Unit	PU Addr	PU Offset	Entry	E Addr	E Offset	Statement	Load Mod	Service	Status
00022460	CEEHSDMP	12B7F950	-0001FE3A	CEEHSDMP	12B7F950	-0001FE3A				Call
00020018	CEEHDSP	12B27138	+000026A4	CEEHDSP	12B27138	+000026A4				Call
000223C0		12B00B80	+00000062	func_with_error						
					12B00B80	+00000062				
Exception										
00022320		12B00A00	+0000005E	second_func_called						
					12B00A00	+0000005E				Call
00022280		12B00880	+0000005E	first_func_called						Call
					12B00880	+0000005E				Call
000221E0		12B005E8	+00000066	main	12B005E8	+00000066				Call
000220C8		12D5BE0E	+000000B4	EDCZMINV	12D5BE0E	+000000B4				Call
00022018	CEEBBEXT	0000E690	+0000013C	CEEBBEXT	0000E690	+0000013C				Call

- Not really a CEEDUMP.
- Missing some info - we've requested it.

SYSTEM DUMPs

■ VERBX CEEERRIP 'CM'

■ COMMAND: VERBX CEEERRIP 'CM'

— (First CIBH is zeroes and should be ignored)

Condition Management Control Blocks

```
HCOM: 12B13AC8
+000008 EYES:HCOM FLAG:60304000 CIBH:000208C0

CIBH: 000208C0
+000000 EYE:CIBH BACK:12B148A0 FRWD:00000000
+000010 PTR_CIB:00000000 FLAG1:00 ERROR_LOCATION_FLAGS:00
+000018 HDLQ:00000000 STATE:00000000 PRM_DESC:00000000
+000024 PRM_PREFIX:00000000
+000028 PRM_LIST:00000000 00000000 00000000 00000000
+000038 PARM_DESC:00000000 PARM_PREFIX:00000000
+000040 PARM_LIST:00000000 00000000 00000000 00000000 FUN:00000000
+000058 FLG_5:00 FLG_6:00 FLG_7:00 FLG_8:00 FLG_1:00
+00005D FLG_2:00 FLG_3:00 FLG_4:00 ABCD:00000000
+000064 ABRC:00000000 OLD_COND_64:00000000 00000000
+000070 OLD_MIB:00000000 COND_64:00000000 00000000
+00007C MIB:00000000 PL:00000000 SV2:00000000
+000088 SV1:00000000 INT:00000000 MID:00000000
+000094 HDL_SF:00000000 HDL_EPT:00000000 HDL_RST:00000000
```

► Ignore the first one. It is waiting for the next condition

SYSTEM DUMPs

■ VERBX CEEERRIP 'CM'... COMMAND: VERBX CEEERRIP 'CM'

— (Second CIBH contains exception information)

```
CIBH: 12B148A0
+000000 EYE:CIBH BACK:00000000 FRWD:000208C0
+000010 PTR_CIB:00020478 FLAG1:C5 ERROR_LOCATION_FLAGS:1F
+000018 HDLQ:00000000 STATE:00000008 PRM_DESC:00000000
+000024 PRM_PREFIX:00000000
+000028 PRM_LIST:00020490 00020558 00020564 12B14CBC
+000038 PARM_DESC:00000000 PARM_PREFIX:00000000
+000040 PARM_LIST:00020554 00020478 00020564 12B14CBC FUN:00000067
+000058 FLG_5:48 FLG_6:23 FLG_7:00 FLG_8:00 FLG_1:00
+00005D FLG_2:00 FLG_3:00 FLG_4:05 ABCD:940C9000
+000064 ABRC:00000009 OLD_COND_64:00030C89 59C3C5C5
+000070 OLD_MIB:00000001 COND_64:00030C89 59C3C5C5
+00007C MIB:00000001 PL:12B00CC8 SV2:000223C0
+000088 SV1:000223C0 INT:12B00BE2 MID:00000003
+000094 HDL_SF:000161A8 HDL_EPT:12BA0F00 HDL_RST:00000000
+0000A0 RSM_SF:000223C0 RSM_POINT:12B00BE6 RSM_MACHINE:12B14BE8
+0000B0 COND_DEFAULT:00000003 Q_DATA_TOKEN:12B149D8 FDBK:00000000
+0000BC ABNAME:.....
```

■ OLD_COND_64: 00030C89 59C3C5C5 is an LE condition token

● CEE3209S

- ▶ This one is good.
- ▶
- ▶ +5D line has ABCD
- ▶
- ▶ Then OLD_COND_64 - Break down the token.

SYSTEM DUMPs

■ VERBX CEEERRIP 'CM'... COMMAND: VERBX CEEERRIP 'CM'

— (Then machine state information like CEEDUMP)

```
Machine State
+000278 MCH_EYE:ZMCH          MCH_GPR00:00022460      MCH_GPR01:12E77040
+000288 MCH_GPR02:92D5BEC2      MCH_GPR03:92B00BCA
+000290 MCH_GPR04:8000E774      MCH_GPR05:00000000
+000298 MCH_GPR06:00000000      MCH_GPR07:0000000A
+0002A0 MCH_GPR08:00000001      MCH_GPR09:80000000
+0002A8 MCH_GPR10:92D5BE02      MCH_GPR11:8000E690
+0002B0 MCH_GPR12:00015910      MCH_GPR13:000223C0
+0002B8 MCH_GPR14:000223C0      MCH_GPR15:12B00B80
+0002C0 MCH_PSW:078D2400 92B00BE6      MCH_ILC:0004      MCH_IC1:00
+0002CB MCH_IC2:09 MCH_PFT:00000000      MCH_FLT_0:4D000000 0002281D
+0002D8 MCH_FLT_2:00000000 00000000      MCH_FLT_4:00000000 00000000
+0002E8 MCH_FLT_6:00000000 00000000      MCH_EXT:00000000
```

- ▶ Machine State (ZMCH)
- ▶ Floating points too

SYSTEM DUMPs

■ VERBX CEEERRIP 'CM'... COMMAND: VERBX CEEERRIP 'CM'
— (and CIB)

```
CIB: 00020478
+000000 EYE:CIB BACK:00000000 FRWD:00000000
+000010 PLAT_ID:00000000 COND_64:000300C6 59C3C5C5
+000020 MIB:00000000 MACHINE:00020ED4
+000028 OLD_COND_64:00030C89 59C3C5C5 OLD_MIB:00000001
+000034 FLG_1:00 FLG_2:00 FLG_3:00 FLG_4:04 HDL_SF:00022018
+00003C HDL_EPT:12BA0F00 HDL_RST:00000000 RSM_SF:000223C0
+000048 RSM_POINT:12B00BE6 RSM_MACHINE:12B14BE8
+000050 COND_DEFAULT:00000003 VSR:00000000 00000000 VSTOR:00000000
+00009C VRPSA:00000000 MCB:00000000 MRN:00000000 00000000
+0000AC MFLAG:00 FLG_5:48 FLG_6:23 FLG_7:00 FLG_8:00
+0000B4 ABCD:940C9000 ABRC:00000009 ABNAME:00000000 00000000
+0000C4 PL:12B00CC8 SV2:000223C0 SV1:000223C0
+0000D0 INT:12B00BE2 Q_DATA_TOKEN:00000000 FDBK:00000000
+0000DC FUN:00000067 TOKE:00022018 MID:00000003
+0000E8 STATE:00000008 RTCC:FFFFFFFC PPAV:00000002
+0000F4 AB_TERM_EXIT:00000000 00000000 SDWA_PTR:00000000
+000100 SIGNO:00000000 PPSD:12B14CCC
```

- NOTE: There may be additional CIBH/CIB pairs for prior conditions.

- ▶ CIB
- ▶ COND_64 not the CEE3209S what is it?
- ▶ CEE0198S - current condition in CEEDUMP....

SYSTEM DUMPS

- Getting the name of the failing csect (function) (or any LE enabled csect).
 - From option I (Browse) in IPCS
 1. Get address (2nd word) from MCH_PSW
 2. L xxxxxxxx (where xxxxxxxx is the addr).
 3. F CEE prev
 4. Back up 5 bytes to the 47F0xxxx instruction
 - a. This is the beginning of the csect/function
 - b. Repeat Step 3 and 4 as needed.
 5. Add the value at offset x'0C' to the module address
(note: offset may be negative)
 6. Go to that location
 7. First byte is offset (or half offset) to name.
 8. This is a 2byte prefixed string with the function name

► Read through this. Explain that it still may be confusing.

SYSTEM DUMPs

■ Getting the name of the failing csect (function)

■ Example: I. MCH_PSW: 078D2400 92B00BE6

```
12B00B80 47F0F026 01C3C5C5 000000A0 00000148 | .00...CEE..... |
| | | | | 5. Take value at offset x'0C' and add to addr
(12B00B80+148)
| | | 3. F CEE PREV
| 4. Back up 5 bytes this is the beginning of the module/csect/function (12B00B80)
12B00B90 47F0F001 183F58F0 C31C184E 05EF0000 | .00...0C..+.... |
12B00BA0 000047F0 303A90E7 D00C58E0 D04C4100 | ...0...X}.. \<.. |
12B00BB0 E0A05500 C3144720 F0145000 E04C9210 | \...C...0.&.\<k. |
12B00BC0 E00050D0 E00418DE 05304400 C1B04150 | \.&}\.....A.& |
12B00BD0 00005050 D0984400 C1AC4160 000A8E60 | ..&&}q..A..-...- |
12B00BE0 00205D60 D0985070 D09C4400 C1AC58F0 | ..)-}q&..}...A..0 |
| 2. MCH_PSW points here do L 12B00BE6
12B00BF0 D09C47F0 302E0700 4400C1B8 58D0D004 | }..0.....A..}}. |
12B00C00 58E0D00C 9837D020 051E0707 12B00C20 | .\}.q. .... |
12B00C10 12B00CB0 12B00A00 80000001 000006D8 | .....Q |
12B00C20 12B00B80 0000008A 12B00C5C 00000000 | .....*.... |
12B00C30 02E0004A 00000300 004E0000 03200057 | ..\..+..... |
12B00C40 00000360 006B0000 03800078 000003A0 | ...-..... |
12B00C50 00780000 03A0007C 000003A0 0E0E0E0E | .....@..... |
12B00C60 0E0E0000 000F86A4 95836DA6 89A3886D | .....func_with_ |
12B00C70 85999996 99400001 93400001 94400000 | error ..l ..m .. |
12B00C80 00000692 01012000 00000198 D000009C | ...k.....q}... |
12B00C90 0000068E 01012000 00000198 D0000098 | .....q}..q |
12B00CA0 00000698 000006A8 00000000 00000000 | ...q...y..... |
12B00CB0 00000000 0000008A 000006B8 00010000 | ..... |
12B00CC0 00000002 F04A3042 10CEA186 FFFFFB08 | ...0\.....f.... |
| 6. Go to that location (12B00CC8)(x'10' is half offset here)
| 7. Add x'20' to 12B00CC8 to get 12B00CE8
12B00CD0 0000008C 00000000 FFC00000 00000000 | .....{..... |
12B00CE0 90000000 02C00019 000F86A4 95836DA6 | .....{....func_w |
| 8. Add x'20' bytes to get to 2 byte prefixed string with name.
12B00CF0 89A3886D 85999996 99405000 0045FFFF | ith_error &..... |
```

► Go through these steps. They should get it now!

SYSTEM DUMPs

■ Other LEDATA (CEEERRIP) options

- 'SM' - VERBX CEEERRIP 'SM'
 - ▶ All Storage Management control blocks
- 'HEAP' - VERBX CEEERRIP 'HEAP'
 - ▶ Just HEAP Storage Management control blocks
- 'STACK' - VERBX CEEERRIP 'STACK'
 - ▶ Just STACK Storage Management control blocks
- 'ALL' - VERBX CEEERRIP 'ALL'
 - ▶ All control blocks including CRTL and COBOL
- 'CAA(xxxxxxxx)' - Combined with other options
 - ▶ Allows specific CAA to be used as anchor (required for CICS) - VERBX CEEERRIP 'ALL CAA(xxxx)'
- 'DSA(xxxxxxxx)' - Combined with other options
 - ▶ Allows specific DSA to be used for traceback info
- 'TCB(xxxxxxxx)' - Combined with other options
 - ▶ Allows specific TCB to be used as starting point

- ▶ Other options
- ▶ STACK and HEAP in part 2 - It pays to advertise

Summary

- LE is **NOT** 100% compatible with previous runtimes
- CEE and EDC are generally supported by Poughkeepsie.
- IBM, AFH, and IGZ are generally supported by STL.
- CEE32xx messages are really ABEND0Cy.
- Don't SLIP on an LE reissued ABEND!
- Change runtime options to get a U4039 dump
- U4039 and other LE reported errors need further investigation to determine owner of **original** problem.
 - GET PSW and Regs from CEEDUMP and/or CEEERRIP.
- Use **EXCEPTION** line in CEEDUMP for program checks.
- Use VERBX CEEERRIP in SYSTEM DUMPS.

► Go through summary...

Sources of Additional Information

- All LE documentation available on DISK I of OS/390 CD collection and on the LE website
- LE Debug Guide and Runtime Messages
- LE Programming Reference
- LE Programming Guide
- LE Customization
- LE Migration Guide
- LE Writing ILC Applications
- Web site
 - <http://www.ibm.com/s390/le>



► Books