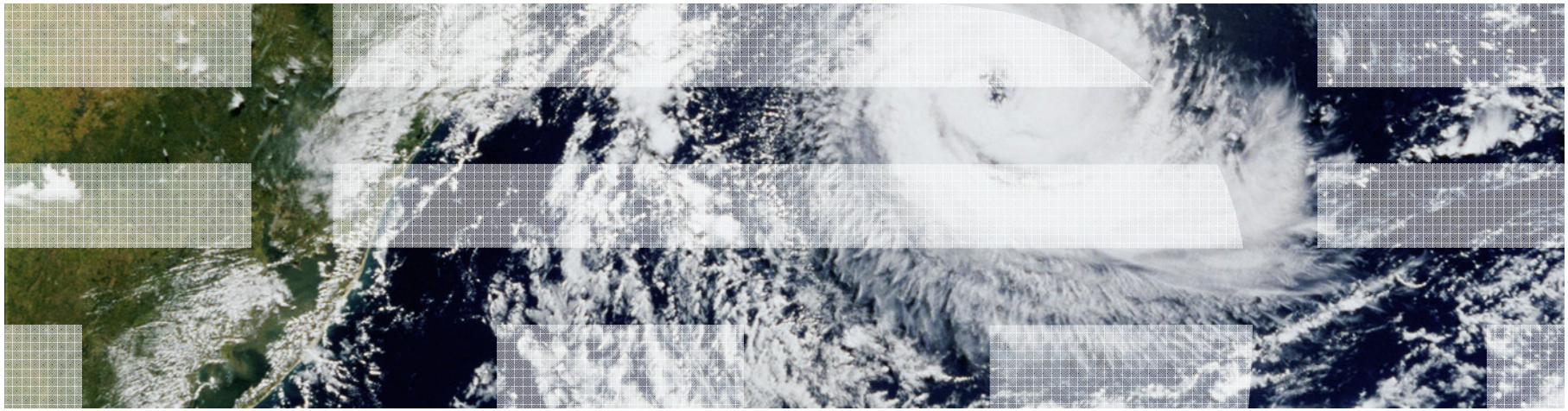
# Bringing You Up to Date with LE for z/VSE

Wolfgang Bosch, IBM Labor Böblingen, email: wbosch@de.ibm.com



http://www.ibm.com/zVSE

http://twitter.com/IBMzVSE

Thursday May 12th, 2011

# Trademarks

**The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.**

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, z/VSE, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

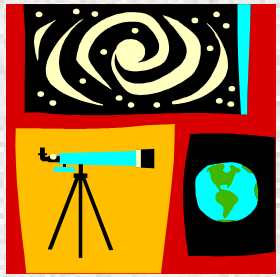
**The following are trademarks or registered trademarks of other companies.**

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

**Notes**:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

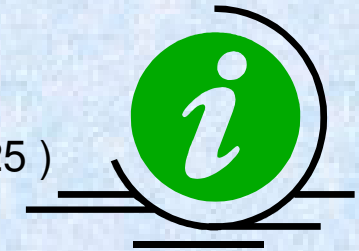
# Overall Contents



- **Module 1:  LE/VSE Overview, Recap and more … ( p.4 ff. )**
- **Module 2: LE/VSE Functional Enhancements with z/VSE 4.3 ( p.26 ff. )**
- **Module 3: Tools Updates, Application Development/Debugging & more ( p.41 ff. )**

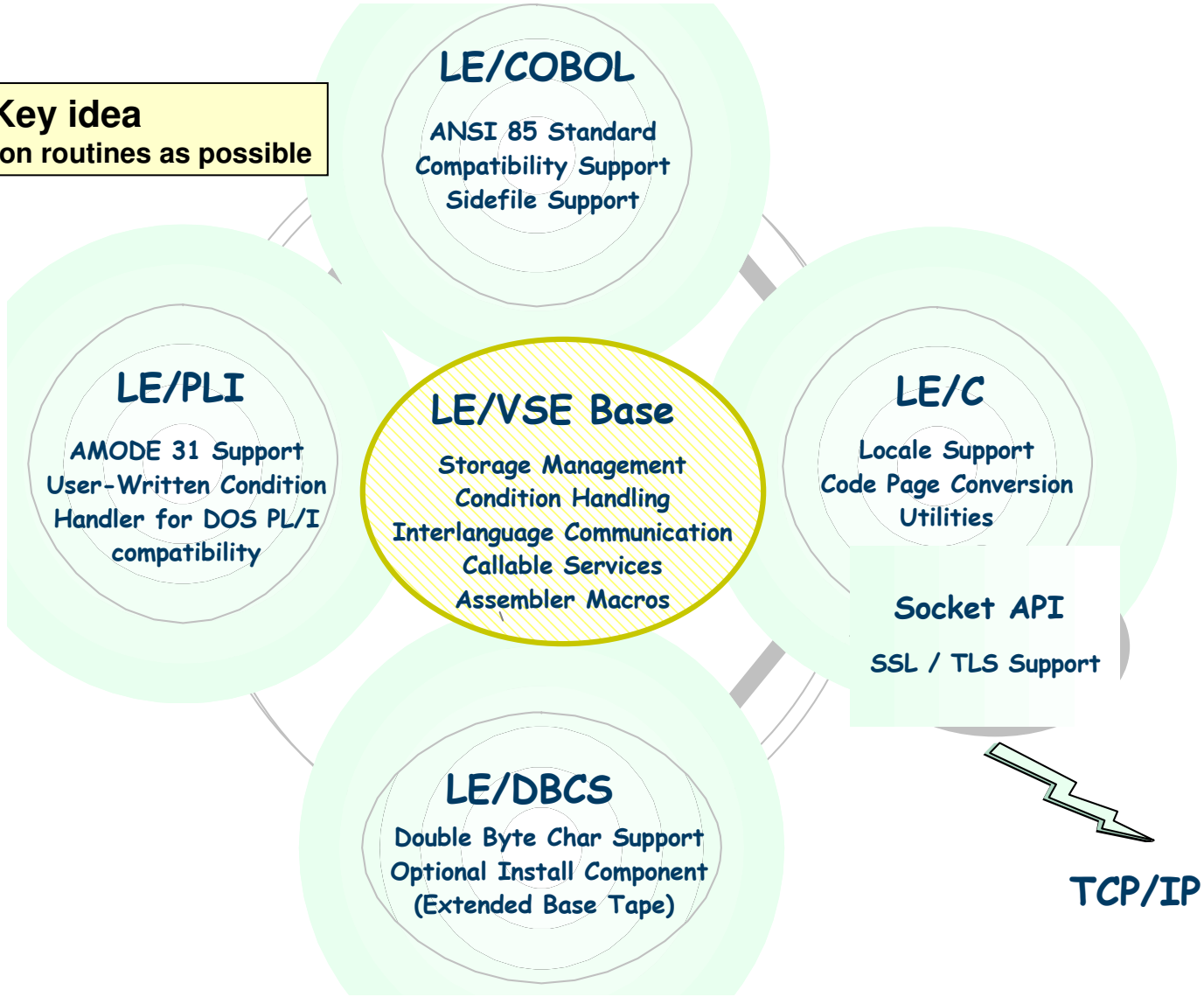
# Module 1

- **LE/VSE Overview, Recap and more …**
  - Product Structure ( p.5 )
  - Supported Languages + Compilers ( p.6 )
  - Languages and Transaction Subsystems, ( p.7 )
  - Callable Services and Common Functions ( p.8 )
  - Interfaces to Other Products ( p.9 )
  - Run-Time Options ( p.10,11 )
  - z/VSE Application Context, Generating Applications ( p.12,13 )
  - How LE/VSE Uses Registers, Support for LE/VSE Conformity ( p.14-16 )
  - Programming Assembler and High Level Languages ( p.17 )
  - TCP/IP Programming ( p.18,19 )
  - Interlanguage Communication ( p.20 )
  - Approaches for Debugging, Sample Views + References ( p.21-25 )



# Product Structure: LE/VSE Consists of 5 Run-Time Components

**Key idea**
**As much common routines as possible**



# Supported z/VSE Languages, Compilers

- **The following High Level Languages (HLLs) are supported in VSE**
  - COBOL, PL/I and C language (spreading: ~80%, ~10-15%, ~5%)
  - Single or mixed language applications can be build
- **LE/VSE-conforming (HLL) compilers are ...**
  - IBM COBOL for VSE/ESA …
  - IBM PLI for VSE/VSE …
  - IBM C for VSE/ESA …
  - Option to involve: High Level Assembler (HLASM) … coding via LE supplied macros
- **Non-LE/VSE Compilers -> typically of "older-age" …**
  - DOS/VS COBOL (EOS: 12/31/1999) ★ there exists LE/VSE Run-Time compatibility support ●
  - VS COBOL II (EOS: 06/30/1998) ★ there exists LE/VSE Run-Time compatibility support ●
  - DOS PL/I ●
  - C/370 ●
  - Fortran (just stand-alone, not part of LE/VSE run-time ⇔ different in LE for z/OS !)
  - RPG (just stand-alone, not part of LE/VSE run-time)

★ Existent legacy COBOL Applications can be LE/VSE-enabled for continued support ...

● Compiler should nomore be used for new development or program maintenance!

# Languages & Transaction Sub-Systems

- **CICS Transaction Server for VSE/ESA supports**
  - All LE/VSE conforming compilers (C/VSE, COBOL/VSE, PLI/VSE)
  - LE/VSE enabled HLASM via CEEENTRY macro and MAIN=NO parameter
  - DOS/VS COBOL and VS COBOL-II (if relinked / using LE/VSE run-time)
  - RPG-II ★(has native run-time support, no LE/VSE support !)
- **CICS TS will not support**
  - DOS PL/1 & C/370 applications (require recompilation with LE-conforming compiler)
- **Supplied LE/CICS Transactions for Run-time Option Control - in flight …**
  - "**CLER**" (tailor LE/VSE CICS-wide default run-time options interactively) http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/fl2ice09/A.4
  - "**ROPC**" (print LE/VSE CICS-wide run-time options to your VSE/ESA console) http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/fl2ice09/A.6
  - "**NEWC**" (activate changed run-time options, while CICS is running) http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/fl2ice09/A.5
- **LE/VSE messages at CICS startup time (CEE3550-3552I)**
  - Confirm language-specific components are operational
  - Implicit check the presence of language-specific event handlers: (03 => C, 05 => PLI, 10 => COBOL)
- **CICS System Definition (CSD) file migration**
  - Shared/non-shared setup (LE/VSE entries must be current/match VSE release)

# LE/VSE Callable Services and Common Functions

- **The following functional areas are covered via Callable Services**
  - Condition Handling, Date and Time
  - Dynamic Storage, General Callable Services
  - Initialization and Termination
  - Locales, Math
  - Message Handling
  - National Language Support
  - Details & Samples (LE/VSE Programming Reference):
    http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/FL2PRE08/3.0

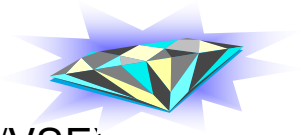
Sample programs (for each service) shipped in LE/VSE product library:

- COBOL/VSE (prefixed: IGZ*.C),
- PLI/VSE (prefixed: IBM*.P)
- C/VSE (prefixed: EDC*.C)

- **Wide range of common functions, utilities, options … available for use**
  - LE-AR commands (status, exit, run-time option control -> D CEE,…; S CEE,…)
  - Dynamic call facilities (involvement of run-time library, assembler call macros …)
  - International Programming Features and Utilities (e.g. code page conversions, locales)
  - Functions pre-LE/VSE run-times have not supported (e.g. device independent coding)
  - Improved Interlanguage Communication (ILC)
  - Common debugging approaches (LE traceback, dump, CEETRACE feature, LSTQ)
  - Compatibility support for legacy applications (under defined conditions, mainly COBOL)
  - User Written Condition Handlers, Inspect Condition Tokens and act upon …
  - … and lot's more

# Interfaces to Other Products

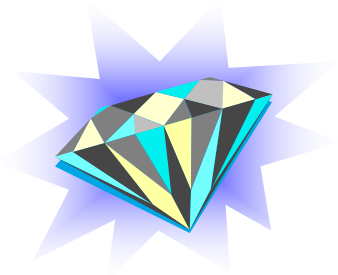
- **There are interfaces / interactions to following VSE Products / Components**
  - **Advanced Functions** (e.g. Attention Routine commands, dynamic DTF build, PRODEXIT)
  - **CICS/VSE/TS** (LE/VSE itself uses EXEC CICS requests)
  - **DB2 for VSE/ESA** (LE conforming application)
  - **Debug Tool for VSE/ESA** (e.g. invoke via LE/VSE, CEETEST callable service API)
  - **DFSORT** product ⇔ LE/VSE capabilities (handle SORT/MERGE verbs in COBOL ... or PLISRT*x* interface in PL/I)
  - **DL/I** (CEETDLI service to call DL/I e.g. add record in DB or set checkpoints)
  - **Some interactions with LE/VSE Conforming Compilers** (e.g. C/VSE, PLI/VSE) … dependent on code compiled
  - **TCP/IP** (LE/VSE C Run-Time Socket API)
  - **Visual Age Generator** (VAGEN), nowadays **Rational COBOL Run-Time** (complementary "run-time" to LE/VSE) to support web-based applications generated via IBM's new business language (EGL = Enterprise generation Language). They are generated and deployed from eclipse based environment (to run in the VSE back-end) …
  - **VSAM** (e.g. extract catalog / file attributes, compare with DLBL's …)



**More about LE/VSE's capabilities …**

# LE/VSE Run-Time Options

- **Set of control options to support "self-management" of LE/VSE run-time**
  - Customizable in various ways (console-, user exit-, PARM overrides … and much more)
  - Either of general nature or language specific (dependent on option)
  - For key areas of LE/VSE operation such as storage, condition management …
  - Reduce dependency on user interaction …
- **Key areas likely to be subject of run-time option customization**
  - Debug assistance (ABEND, CHECK, TERMTHDACT, TEST, TRAP)
  - Language Specific settings/adoptions (CBLOPTS, CHECK, ERRCOUNT)
  - Storage Management (HEAP, STACK, internal areas)



**More about LE/VSE's capabilities …**

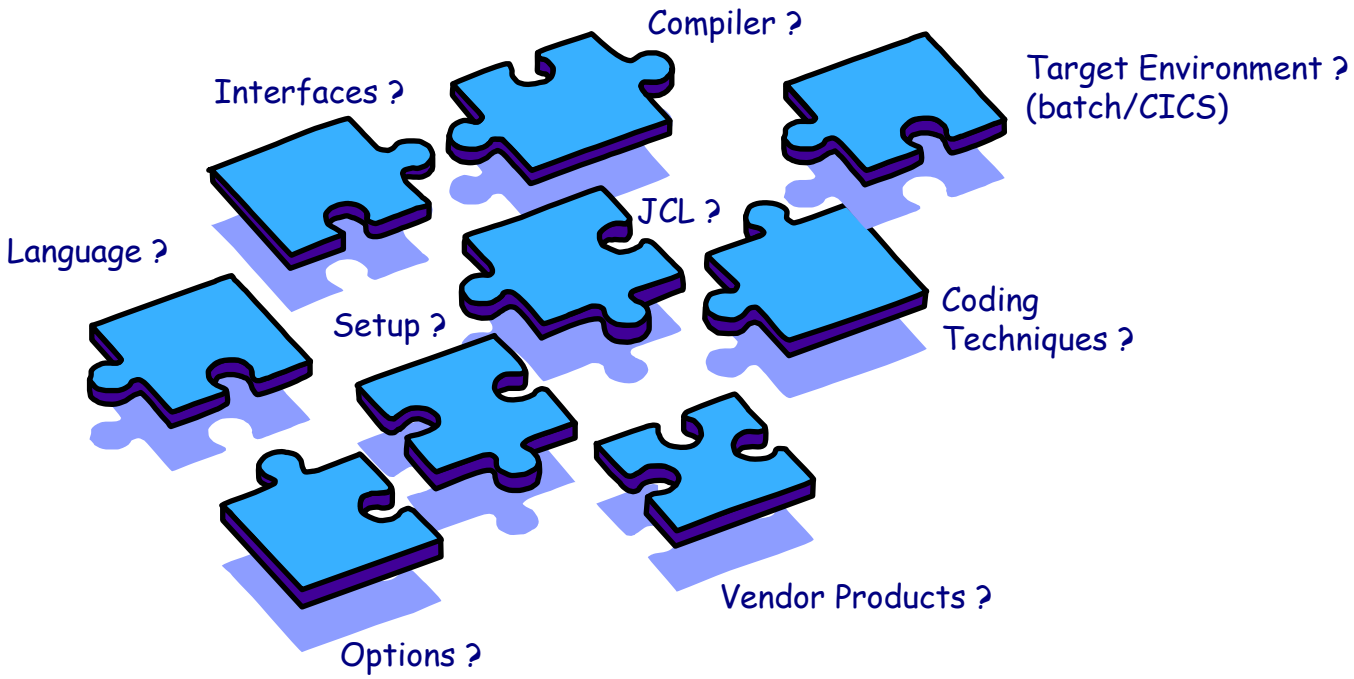
# Recommended LE/VSE Run-Time Options

- **ABTERMENC(ABEND)**
  - For diagnostics
- **ALL31(OFF)**
  - AMODE switch, especially on CICS
  - ALL31(ON), if all migrated 31-bit
- **CBLOPTS(ON)** COBOL
- **CHECK(ON)** COBOL
  - Maps VS COBOL II SSRANGE run-time option
  - Overlay detection (application compiled CBL SSRANGE)
- **ERRCOUNT(20)** COBOL+C
  - PLI/VSE applications: ERRCOUNT(0) PL/I
- **HEAP(ANYWHERE)**
  - Relationship between COBOL compiler options RMODE, DATA(24|31), HEAP run-time option and CICS 24-bit GETVIS limit of 64k
  - Avoid HEAP(BELOW), it ignores DATA !
- **NODEBUG**
  - Better performance
- **NOTEST**
  - TEST, if running Debug Tool for VSE/ESA
- **RTEREUS(OFF)** COBOL
  - (ON) for special environments with repeated batch calls
- **STACK**
  - STACK(BELOW) with ALL31(OFF)
  - STACK(ANY) with ALL31(ON)
- **STORAGE(00,NONE,NONE,32K)** COBOL
  - Maps to VS COBOL II WSCLEAR, migration assist for programs relying on preinitialized storage pattern
- **STORAGE(00,NONE,CLEAR,32K)** PL/I
  - PL/I migrations, automatic variables, performance
- **TERMTHDACT(UADUMP)**
  - For diagnosis in batch
  - Not useful under CICS. Take suboption 'DUMP' here.
- **TRAP(ON,MAX)**
  - For LE/VSE's integrity handling abnormal conditions
- **TRAP(ON,MIN)**
  - Recommended setting in place of TRAP(OFF)
- **In general for PL/I users:** PL/I
  - Take care about DEPTHCONDLMT, ERRCOUNT, USRHDLR, XUFLOW runtime options. There are related notes available in current documentation

http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/FL2ICE09/A.3

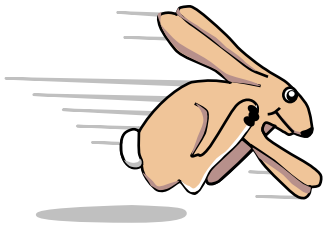
# z/VSE Application Context

- **Typical interdependencies ...** **User Applications**




- VSE code developed in C/VSE, PLI/VSE, COBOL/VSE, Assembler, ... involving LE/VSE run-time and server based products such as ... DB2/VSE, DL/I, CICS, MQ/Series, DFSORT, Debug Tool for VSE/ESA

# Overview - Generating Applications



- **Using Interactive Interface Support**
  - Primary Library, OPTION 8=COMPILE
  - JCL build skeletons for z/VSE languages (and call structures)
  - Batch main/subroutines, CICS or DB2 target environments etc.
  - Arranges for suitable program translate and compile steps
  - Ensures usage of language independent stubs such as CICS DFHELII ...
  - Appropriate compile and pre-processor options are used etc.
- **Program Maintenance from outside-VSE environment**
  - (e.g. VM, FTP etc.) ensure use of current compile skeletons, related options, stubs etc.
- **Modern ways to generate VSE applications (eclipse-based environments)**
  - Rational Developer for System z (COBOL development work frame) ...
  - Rational Business Developer (RBD) and Enterprise Generation Languages (EGL) (Customizable <build Descriptors> ensure deployment & generation)
  - Also see: z/VSE Home -> Solutions -> Multi-Platform Development and VSE http://www-03.ibm.com/servers/eserver/zseries/zvse/solutions/egl.html
- **Security Prerequisites (independent from method chosen)**
  - Every CICS TS transaction need to be "security-enabled" prior to first execution
  - The IUI provides associated support via dialog TAS$SECF, accessible via selection path "Resource Definition" --> "Define Transaction Security"

# How LE/VSE Uses Registers (Conventions)

- **When a routine makes a call, registers 0-15 contain the following values**
  - **R0** is used by COBOL static call
  - **R1** is a pointer to parameter list or 0 if no parameter list passed
  - **R2-R11** is not used by LE/VSE. Caller's values are passed transparently
  - **R12** is the pointer to the Common Anchor Area (CAA) if entry to an external routine
  - **R13** is the pointer to caller's stack frame
  - **R14** is the return address
  - **R15** is the address of the called entry point
- **Stack frames (or Dynamic Storage Areas / DSAs -> R13)**
  - Acquired every time a separately compiler procedure or block is entered
  - There exist standard save are "**back chain**" and "**forward chain**" pointers
  - "Forward chain" pointers might **not** be reliable any time !

**Different register use is not LE/VSE conform !**

# Support for LE/VSE Conformity & Proper Linkage

- **Create LE/VSE environment from assembler and maintain mixed applications**
  - The following LE/VSE macros and services can assist to ensure LE/VSE-conformity
    - **Assembler macros** (CEEENRTY/CEETERM)
    - **Pre-initialization service** (CEEPIPI) ==> "start-facility" from non-LE/VSE main
    - **LE/C-specific macros** (EDCPRLG/EDCEPIL)
- **Use of LE/VSE supplied Assembler macros helps to**
  - Organize complexity with call conventions
  - Preserve program linkage (internal & user-type routines)
  - Honor register conventions (R12: CAA, R13: DSA …)
  - Ensures save areas stay intact (keeps LE/VSE integrity)


**Simplified program linkage view …**

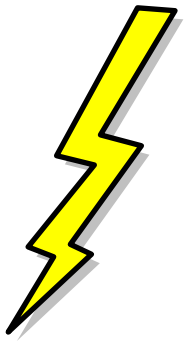
- **Examples with details** are available on LE/VSE Home Page ...
  http://www-03.ibm.com/servers/eserver/zseries/zvse/downloads/samples.html

# Effects with Non-LE/VSE Conforming Applications

- **No preserving of certain registers**
- **Inconsistent use of storage areas and lost linkage** to save areas (R13 corrupted ...)
- **No or limited traceback capabilities**

TRACEBACK:

| DSA ADDR | PROGRAM UNIT | PU ADDR | PU OFFSET | ENTRY | E ADDR | OFFSET | STATEMENT | STATUS |
|---|---|---|---|---|---|---|---|---|
| 007CB018 | CEEHDSP | 0071D920 | +0001DCE | CEEHDSP | 0071D920 | +0001DCE | | CALL |
| 00794528 | CEEHSGLT | 007243A8 | +0000005C | CEEHSGLT | 007243A8 | +0000005C | | EXCEPTION |
| 00794018 | IGZCMSG | 00778E40 | +0000037A | IGZCMSG | 00778E40 | +0000037A | | CALL |
| 00792018 | IGZEQOC | 007C7000 | +00000F8C | IGZEQOC | 007C7000 | +00000F8C | | CALL |
| 00701C | TEST | 00700078 | +00000296 | TEST | 00700078 | +00000296 | 22 | CALL |



- **No robustness / reliability at run-time ...**
- Typical error is the **corrupted back chain** (back-chain linkage problem) --> abend 4083 RC3 ?
- **Stack frame allocation failures** --> abend: 4088 RC99 ?

# Programming Assembler & HLL Routines (Compendium)

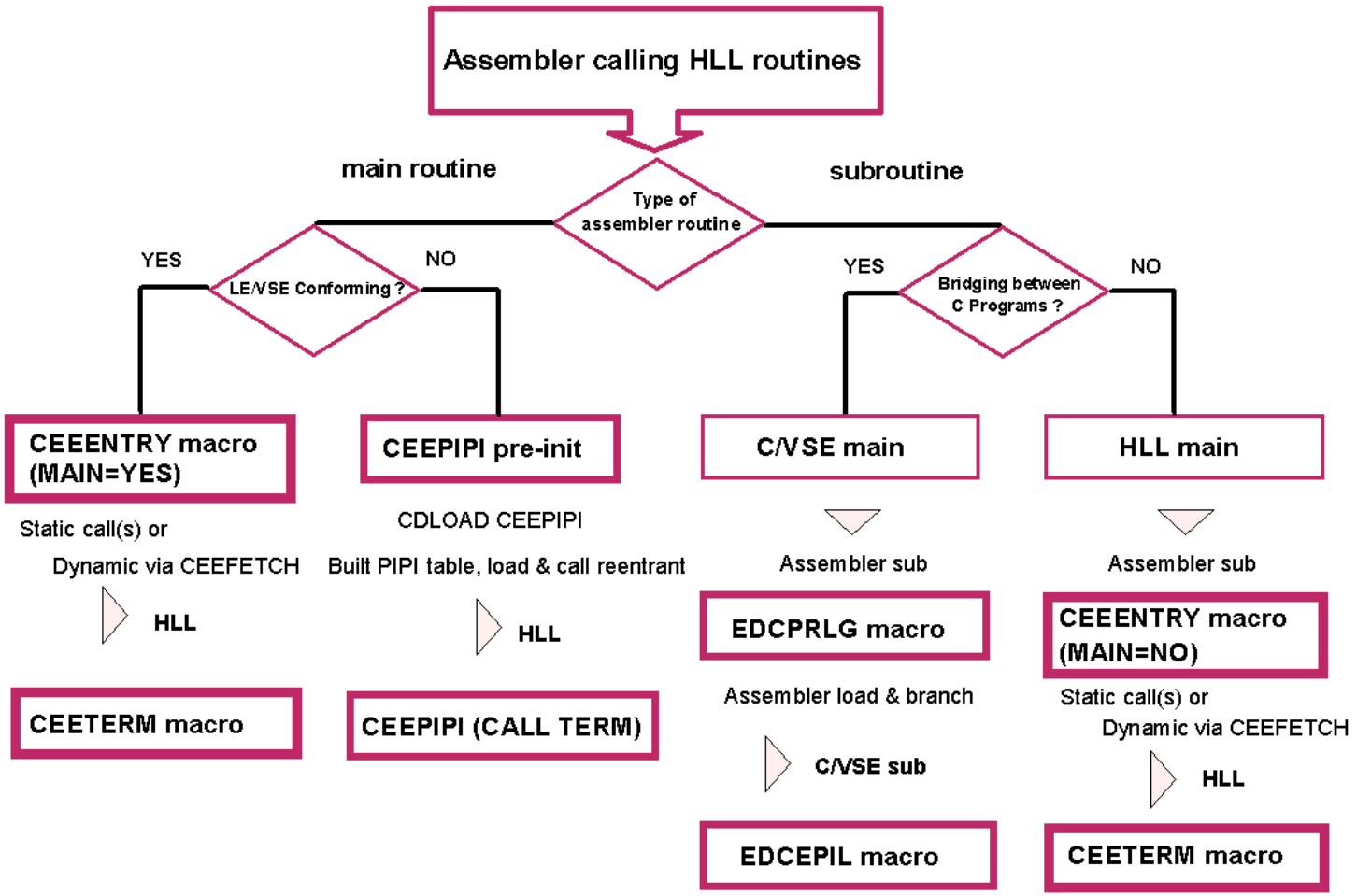



**Fig. Assembler Calling High Level Language (HLL) Routines**

# TCP/IP Programming and Languages

- **Coding options for TCP/IP Socket Programming**
  - EZASMI macro interface (for HLASM programs)
    -> widely OS/390 source compatible
  - EZASOKET (for HLASM, COBOL/VSE and PLI/VSE programs)
    -> widely OS/390 source compatible
  - Use of TCP/IP Callable C-functions

  **Documentation available via:**
  **z/VSE V4R3.0 TCP/IP Support ( SC34-2604-00 ), Chapter 3:**
  http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/iestce40/3.1.6

- **EZASMI and EZASOKET suit for batch and CICS/TS environments**
  - CSD entries (phases EZASOH00, EZASOH01, EZASOH99) are pre-defined
  - 31-bit Addressing Mode is required
  - Interface Module EZASOH00 is SVA-31 eligible
- **Other coding alternatives**
  - CSI TCP/IP SOCKET macro (efficient from Assembler, not portable)
  - CSI EXEC TCP processor (efficient but not portable)

# Secure TCP/IP Socket Programming Model with LE/C

## Client

**gsk_initialize()**

**socket()**

**connect()**

**gsk_secure_soc_init()**
***skwrite() .....**
***skread() .....**
**gsk_secure_soc_write()**
***skwrite() .....**
**gsk_secure_soc_read()**
***skread() .....**
**gsk_secure_soc_close()**
***skwrite() .....**
***skread() .....**

**close()**

## Server

**gsk_initialize()**
**socket()**

**bind()**

**listen()**

**accept()**
**gsk_secure_soc_init()**
***skread()**
***skwrite()**
**gsk_secure_soc_read()**
***skread()**
**gsk_secure_soc_write()**
***skwrite()**
**gsk_secure_soc_close()**
***skread()**
***skwrite()**
**close()**

**close()**

Exploits LE/VSE API from
… function usage point of view

SSL: secure socket layer
TLS: transport layer security

(synonymous use of term)

- Link to SSL & TCPIP Callable functions via: LE/VSE C Run-Time Library Reference SC33-6689

  http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/FL2CLE05/3.0

# Interlanguage Communication

- **Communicating between C and PL/I**
  - Static and dynamic calls are allowed. For details:
    LE/VSE V1R4 Writing Interlanguage Communication Applications SC33- 6686:
    http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/FL2WIE02/3.0
  - For example: PL/I routine may focus on application logic, C routine could cover data transfer issues
- **There exists similar support for C-COBOL and PL/I-COBOL (ILC)**
  - C-COBOL (ILC):
    http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/FL2WIE02/2.0
  - COBOL-PL/I (ILC):
    http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/FL2WIE02/4.0
- **Same for the CICS environment ...**
  - Compile units must be compiled with LE/VSE conforming compilers
  - EXEC CICS XCTL/LINK constructs should be preferred
  - Overview:
    http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/FL2WIE02/7.0

# Approaches for Debugging + Sample References …

- **Identify the Failing LE/VSE Run-Time Component & Language Type …**
  - **Prefix: CEE** -> LE/Base**, IGZ*** -> LE/COBOL, **EDC*** -> LE/C, **IBM*** -> LE/PL/1
- **LE/VSE Diagnostic Messages on VSE Console**
  - **CEE3320C** pointing to abend Program Status Word (PSW)
  - **CEE3321C** VSE cancel code and interruption code reference
  - **CEE3322C** with LE/VSE abend code references ('4xxx' type code)
  - **CEE1000S** referring to accompaining ('4xxx') abend and hex reason code
    -> Abend display via OME-file, string -> '**CEEABENDC**' on console -> PF9
  - Doc link to LE/VSE Debugging & Run-Time Messages Guide SC33-6681
    http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/FL2DRE09/3.0
- **Conditions & Message Relation …**
  - **CEE0374C COND=IGZ3S** ---> IGZ**000**3S (Logic error I/O operation)
    - Mid-padding with zeros is useful for interpretation
    - The string will map existing run-time message number that can be looked up
    - The "COND=...." reference is reliable even if the actual message cannot be issued !
  - **Online applications** may fail in case LE/VSE-CICS transient data queue destinations **CESE** and **CESO** are not defined !

# LE/VSE Traceback Facility - PL/I example

```
TRACEBACK: Plidump called from error ON-unit
     DSA ADDR  PROGRAM UNIT  PU ADDR   PU OFFSET  ENTRY          E ADDR    E  OFFSET   STATEMENT  STATUS
      006A79B0  CEEKKMRA      00227FA0  +000008C4  CEEKKMRA       00227FA0  +000008C4              CALL
      006C77C0  IBMRKDM       00675258  +000000BA  IBMRKDM        00675258  +000000BA              CALL
      006A78B0  EXAMPLE       005000F8  +000001EA  ERR  ON-UNIT   005001F8  +000000EA            6 CALL
      006A76A0  IBMRERPL      0068CF98  +0000063E  IBMRERPL       0068CF98  +0000063E              CALL
      006A75B8  CEEEV010      00658800  +00000126  CEEEV010       00658800  +00000126              CALL
      006A4018  CEEHDSP       03A58D38  +000012FE  CEEHDSP        03A58D38  +000012FE              CALL
      006A7430 IBMRERRI       0068DDD8  +0000036E  IBMRERRI       0068DDD8  +0000036E              EXCEPTION
      006A7360  EXAMPLE       005000F8  +000002E2  LABL1: BEGIN   00500308  +000000D2           11 CALL
      006A7260  EXAMPLE       005000F8  +000000E0  EXAMPLE        00500100  +000000D8            8 CALL
      006A71C0  IBMRPMIA      00692E60  +0000028E  IBMRPMIA       00692E60  +0000028E              CALL
      006A70D8  CEEEV010      00658800  +00000282  CEEEV010       00658800  +00000282              CALL
      006A7018  CEEBBEXT      001FA3E0  +00000132  CEEBBEXT       001FA3E0  +00000132              CALL

   CONDITION INFORMATION FOR ACTIVE ROUTINES
     CONDITION INFORMATION FOR IBMRERRI (DSA ADDRESS 006A7430)
       CIB ADDRESS: 006A4528
       CURRENT CONDITION:
         IBM0281I A PRIOR CONDITION WAS PROMOTED TO THE 'ERROR' CONDITION
       ORIGINAL CONDITION:
         IBM0421I 'ONCODE'=520  'SUBSCRIPTRANGE' CONDITION RAISED
       LOCATION:
         PROGRAM UNIT: IBMRERRI ENTRY: IBMRERRI STATEMENT:  OFFSET: +0000036E
     STORAGE DUMP NEAR CONDITION, BEGINNING AT LOCATION: 0068E136
   +000000 0068E136  5030D080 58A0C2B8 58F0A01C 4110D080  05EF9108 204F4710 B3849104 204F47E0 ....
```


- Associated details via LE/VSE Debugging Guide, SC33-6681 (incl. sample program source)
  http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/fl2dre09/2.3.5.1

# Understanding LE/VSE Dump (Key Section View)

```
CEE5DMP V1 R4.6: Sample LE/VSE dump 1 07/27/01 11:45:54 AM PAGE: 1
CEE5DMP CALLED BY PROGRAM UNIT HANDLER AT OFFSET +0000035C. 2
REGISTERS ON ENTRY TO CEE5DMP: 3
PM....... 0000 GPR0..... 0057476C GPR1..... 00574788 GPR2..... 0051F7EC GPR3..... 00578250  ...
...
INFORMATION FOR ENCLAVE DRVHAND 4
...
INFORMATION FOR THREAD 8000000000000000 5
...
TRACEBACK: 6
DSA ADDR PROGRAM UNIT     PU  ADDR PU  OFFSET  ENTRY    E ADDR    E OFFSET   STATEMENT  STATUS
00574620 HANDLER          00500658 +0000035C   HANDLER  00500658  +0000035C              CALL
0054F018 CEEHDSP          03A58D38 +000017A0   CEEHDSP  03A58D38  +000017A0              CALL
00574038 DRVHAND          00500078 +00000350   DRVHAND  00500078  +00000350        108   EXCEPTION
...
CONDITION INFORMATION FOR ACTIVE ROUTINES 7
CONDITION INFORMATION FOR DRVHAND (DSA ADDRESS 00574038) CIB ADDRESS: 0054F528 CURRENT CONDITION:
  CEE3209S THE SYSTEM DETECTED A FIXED-POINT DIVIDE EXCEPTION.
...
PARAMETERS, REGISTERS, AND VARIABLES FOR ACTIVE ROUTINES: 8
HANDLER (DSA ADDRESS 00574620): SAVED REGISTERS:
...
CONTROL BLOCKS FOR ACTIVE ROUTINES: 9
DSA FOR HANDLER: 00574620
...
STORAGE FOR ACTIVE ROUTINES: 10
...
CONTROL BLOCKS ASSOCIATED WITH THE THREAD: 11
...
ENCLAVE CONTROL BLOCKS: 12
...
ENCLAVE STORAGE: 13
...
PROCESS CONTROL BLOCKS: 14
...
File Status and Attributes: 15
```

- **Detailed Sample Reference:**
  http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/fl2dre09/1.3.3.4

# Language Event Tracing in CICS Sub-Systems

- Failure during LE/VSE ownership processing, COBOL/VSE application (**CICS/VSE**)

**X'D1' trace entries** →

```
ID  REG 14   REQD TASK FIELD A  FIELD B  CHARS    RESOURCE  TRACE TYPE
D1 00000000 3203 00057 00000000 00779D54 ........PCMM0010  LIP ESTABLISH OWNERSHIP TYPE
D1 0075DD36 2203 00057 00000000 00779D54 ........PCMM0010  LIP PROGRAM CHECK RECOVERY
FD 1606139C 0104 00057 004A85B9 004A85B9 ........          ... REPEAT 06139 TIMES
D1 007C4282 2205 00057 00000000 00000008 ........          LIP RETN PROGRAM CHECK RECOVERY
F2 807C4478 6004 00057 C1E2D9C1 00000000 ASRA....          PCP ABEND
F1 4075ACF2 CC04 00057 000000C8 018523A4 ...H....          SCP GETMAIN INITIMG
```

→ **DFHLIP: Language Interface Program**

- Successful execution of C/VSE application (**CICS/VSE**)

```
ID  REG 14   REQD TASK FIELD A  FIELD B  CHARS    RESOURCE   TRACE TYPE
D1 00000000 1403 01325 00000000 004FCB84 ........EDCYCROP   LIP THREAD INITIALIZATION
....
D1 00455EF8 2105 01325 00000000 00000000 ........           LIP RETN RUN UNIT END INVOCATION
D1 00455EF8 1503 01325 00000000 00000000 ........           LIP THREAD TERMINATION
D1 00454BC8 1505 01325 00000000 00000000 ........           LIP  RETN THREAD TERMINATION
F0 40454D0E 8004 01325 00000000 00000000 ........           KCP DETACH
```


CICS transaction dumps contain internal traces

- Same C/VSE application (**CICS/TS**)

**X'AP' trace entries** →

```
01818 1 LD 0002 LDLD  EXIT  ACQUIRE_PROGRAM/OK   82302000,02302000,2E00,0,REUSABLE,ESDSA,OLD_COPY
01818 1 AP 1940 APLI  ENTRY START_PROGRAM        EDCYCROP,CEDF,FULLAPI,EXEC,NO,01C91C60,00000000,...
01818 1 SM 0301 SMGF  ENTRY GETMAIN              A3C4,YES,RUWAPOOL,TASK31
01818 1 SM 0302 SMGF  EXIT  GETMAIN/OK           02003AA8
01818 1 AP 1948 APLI  EVENT CALL-TO-LE/VSE       Thread_Initialization EDCYCROP
01818 1 AP 1949 APLI  EVENT RETURN-FROM-LE/VSE   Thread_Initialization OK EDCYCROP
01818 1 SM 0301 SMGF  ENTRY GETMAIN              250,YES,LE_RUWA,TASK24
01818 1 SM 0302 SMGF  EXIT  GETMAIN/OK           006C0448
01818 1 AP 1948 APLI  EVENT CALL-TO-LE/VSE       Rununit_Initialization EDCYCROP
```

→ **DFHAPLIP: Language Interface Program**

# LE/VSE – Debugging Approaches (Reference)

- **LE/VSE Run-Time Messages and Abend Codes**
  - Check for details:
    http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/fl2dre09/3.0
- **LE/VSE Debugging Approaches (by type)**
  - Traceback-, Dump-, Stack Frame-, Common Anchor Area-, Condition Information
    http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/fl2dre09/1.3.3
- **Assist for Debugging Language Routines**
  - Determine errors in C routines, similar as above:
    http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/fl2dre09/2.1
  - Determine errors in COBOL routines:
    http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/fl2dre09/2.2
  - Determine errors in PL/I routines, generate LE/VSE dump & locate information:
    http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/fl2dre09/2.3
- **Assist for Debugging CICS**
  - LE/VSE + CICS info:
    http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/fl2dre09/2.4
- **Debug Tool for VSE/ESA**
  - Home page + documentation:
    http://www.ibm.com/software/awdtools/debugtoolvse/
    http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/EQAVI000
    http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/EQAVU000

# Module 2

- **LE/VSE Functional Enhancements with z/VSE 4.3**
  - Service for International Currency Symbols ( p.27 )
  - Add-on's to the CEEFETCH Macro ( p.28,29 )
  - Introduce PL/I Multitasking ( p.30,31 )
  - Bad Branch Info to CEE5DMP / LE/VSE traceback+dump ( p.32,33 )
  - LE TCP/IP Multiplexer ( p.34 )
  - Miscellaneous Enhancements ( p.35 )
  - APARs integrated, Available LE/VSE 1.4.7 APARs collection ( p.36,37 )
  - Documentation References ( p.38 )
  - Hints and Tips for z/VSE 4.3 Migration ( p.39,40 )

# LE/VSE Service for International Currency

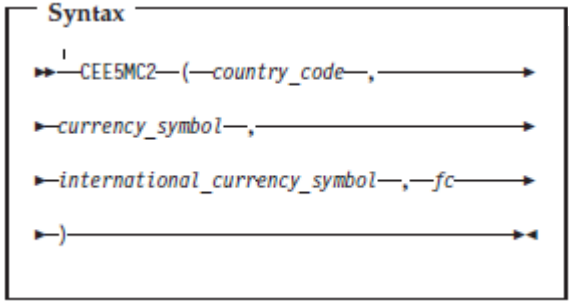
- New Callable Service **CEE5MC2 (Get International Currency as String)**
  - Returns the default currency symbol for specified country code in 2 ways …

| Country | Currency Symbol | Int. Currency Symbol |
|---|---|---|
| Australia | $ X‘5B404040‘ | AUD |
| Finland | € X'5A404040' | EUR |
| Germany | € X'9F404040' | EUR |
| Japan | ¥ X'5B404040' | JPY |
| United Kingdom | £ X'5B404040' | GBP |
| United States | $ X‘5B404040‘ | USD |

Special char,
code page dependent

Special char,
code page dependent

Readable in nearly all code pages !




currency_symbol (output)
A 4-character fixed-length string returned to the calling routine. It contains the default currency symbol for the country specified. The currency symbol is left-justified and padded on the right with blank:

international_currency_symbol (output)
A 3-character alphabetic fixed-length string returned to the calling routine. It contains the international currency symbol for the country specified.

- **CEE5MC2 Service & Default Currency and Picture strings:**
  - LE/VSE Programming Reference Guide, SC33-6685
    - Chapter 3 - LE/VSE Callable Services
    - Appendix A - IBM-Supplied Country/Region Code Defaults

# CEEFETCH Macro Enhancements – Summary –

- **The CEEFETCH macro suits to dynamically load a routine that can be later deleted**
- **New parameters** to support a "wider range" of possible target/candidate routines, respectively enhance the scope and load capabilities
  - **ENTRYPT** (previously loaded module may or may not be LE/VSE conform)
  - **FTCHINFO** (for dynamic calls to all sorts of target routines, main + sub-, in particular, if LE/VSE conformity cannot be assumed)
  - SCOPE=**PROCESS** (in addition to: THREAD and ENCLAVE)
- Similar capabilities are available via CEELOAD macro however with certain restrictions …
  - Target must be an LE/VSE conforming subroutine !
  - In contrary to CEEFETCH there is no corresponding delete service !
  - Limited auto-cleanup capabilities (termination processing), limited scope (enclave/thread)
  - Not supported on target COBOL/VSE programs compiled with SEParate compiler option
  - Cannot be used with target of reentrant C/VSE routines (writable static context)
- Therefore use of CEEFETCH is superior and the recommended choice (over CEELOAD)
- CEELOAD macro is deprecated and only provided for legacy application support
- CDLOAD macro should not be used in an HLL-application/call context, anyhow !
- Please apply PTF for **APAR PM38137** for CEEFETCH corrections (on FTCHINFO results)

# CEEFETCH Macro Enhancements – Keyword Specification –

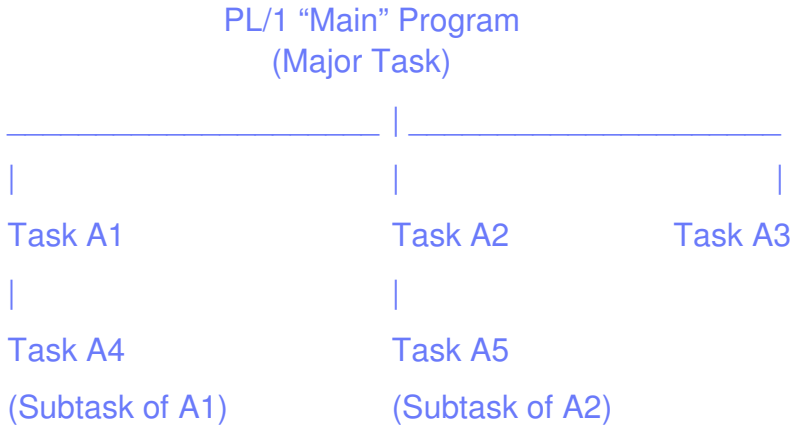
- **ENTRYPT** … relies on FTCHINFO keyword set !
  - If specified, mutually exclusive with NAME, NAMEADDR and SEARCH
- **FTCHINFO** … instructs LE/VSE to attempt load on the target module
  - Keyword must be used in combination with: NAME, NAMEADDR or ENTRYPT
- The **combination** of ENTRYPT and FTCHINFO implies environment preparation (where applicable) and information about the module previously loaded by the user
- If CEEFETCH call target is AMODE24 fetchable C-subroutine, MSG EDC5052I might be issued
  - AMODE of application should match the run-time library !
- Other **CEEFETCH keywords** (short reference):
  - FEEDBACK: symbolic condition returned (transforms to message number)
  - NAME: name of entry point to be loaded by LE/VSE
  - NAMEADDR: address of half-word prefixed name to be loaded by LE/VSE
  - SCOPE: load scope level
  - SEARCH for compatibility with z/OS only. SEARCH=VSE is default
  - TOKEN: variable to contain resulting token (input to CEERELES macro)

# PL/I Multitasking – Overview –

- **Introduces PL/I multi-tasking capability for application programming**
- **Key Concepts** and **Facilities** are**:**
  - Creating of PL/I Tasks (CALL TASK,PRIORITY,EVENT)
  - Coordination and Synchronization of PL/I Tasks (WAIT statement)
  - Termination of PL/I Tasks (RETURN / END statements)
  - STOP and EXIT Processing (entire app <=> subtasks/children)
  - How LE/VSE Options Affect Processing Behavior (inherit caller program options)
  - Debugging of PL/I Tasks (Debug Tool or CEETRACE)
  - Example of PL/I Multitasking (shipped part: IBMIVPMT.P)
- New messages **IBM0565S** and **IBM0566S**
  - Related to ATTACH task request, respectively task creation
- Message **IBM0900S** was complemented to outline associated ONCODE=2050
  - Indicates the WAIT statement not coded properly
- PLI/VSE programs to exploit multi-tasking must be generated with the **TASK** compile option
  - PL/I VSE/ESA compiler PTF UK62805 is required
    - Corresponding APAR: PM17894 (includes documentation)
  - LE/PLI 1.4.7 **PTF UK62594** required (to optimize PL/1 multi-tasking initialization process)
    - Corresponding APAR: PM26532

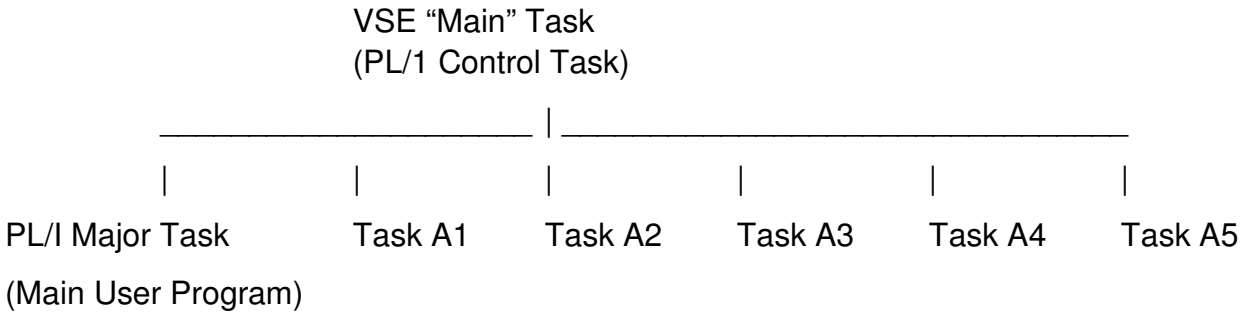
# PL/I Multitasking – Concepts View –

- **PL/I perspective (conceptual hierarchy) … logical dependence**




1. Each PL/I task runs within its own LE enclave, created on its behalf, under supervision of PL/I multitasking control program
2. Task should not terminate itself w/o making sure the context allows it (danger of orphaned subtasks and unpredictable errors)
3. The EVENT variable (in CALL and WAIT statements) is used to coordinate tasks !

- **VSE perspective (actual hierarchy) … tasks run independently**



# Bad Branch Info to CEE5DMP (LE/VSE dump)

- **Prerequisite Info regarding "Break Event Address Register" (BEAR)**
  - BEAR is a 64-bit CPU register, updated with the address of any instruction that causes a break in sequential instruction execution
  - Here, the instruction address in PSW is replaced, not incremented (by instruction length)
  - If feature is present (determined by hardware => z9-109 and above processors), the content of BEAR is stored at real storage location X'110'
  - Overall it serves as hardware provided debugging assist … for "wild" branches
- **LE/VSE relies on BEAR information as defined by the z/Architecture**
  - When available, BEAR info (about bad branch) is included in **CEE5DMP** "condition information section"
  - In case not, the DSA (linkage information) is used in an attempt to produce invalid branch details in CEE5DMP
  - BEAR low core information is likely to be updated frequently in high workload systems !
  - To increase reliability (e.g. in case of page faults or dispatcher interference) we decided to refine current implementation for "Bad Branch Info to CEE5DMP"
  - LE/VSE 1.4.7 **PTF UK62596** required (to exploit new GETFLD FIELD=BEAR capability)
    - Corresponding APAR number: PM26031

# Bad Branch Info to CEE5DMP – Sample View –

- **LE/VSE relies on "BEAR information" in trace and dump facilities …**

**CEE5DMP output:**

```
CEE3201S The system detected an operation exception.
         From compile unit DD:SYSIPT at entry point funca at compile unit offset -00520378 at address 00000000.
         Possible Bad Branch:  Statement: 23  Offset: +0000006E
CEE5DMP V1 R4.7: Condition processing resulted in the unhandled condition.   10/21/10 9:34:49 AM     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  Service  Status
    00652018   CEEHDSP      03B311C0  +00002BD6  CEEHDSP   03B311C0  +00002BD6             LE147GA  Call
    00655288   DD:SYSIPT    00520378  +00000000  funca     00520378  +00000000                      Exception
    006551E8   DD:SYSIPT    005200F8  +00000070  main      005200F8  +00000070         18           Call
    006550D8                03A8425E  +000000B4  @@MNINV   03A8425E  +000000B4                      Call
    00655018   CEEBBEXT     00155370  +00000132  CEEBBEXT   00155370  +00000132             LE147GA  Call

OFFSET OBJECT CODE  LINE#    P S E U D O   A S S E M B L Y   L I S T I N G

000280                        120  funca    DS    0F
000280  47F0  F026            125           B     38(,r15)
00029C  41E0  F03A            126           LA    r14,58(,r15)
...
0002CE                    End of Prolog
0002CE  5840  C1F4            141           L     r4,500(,r12)
0002D2  5010  D088            142           ST    r1,136(,r13)
                    00022 |       *  void funca(int* aa) {
0002D6  4400  C1B0            144           EX    r0,HOOK..PGM-ENTRY
                    00023 |       *    *aa = func_ptr();                  // bad branch at this statement !
0002DA  4400  C1AC            146           EX    r0,HOOK..STMT
0002DE  5870  D088            147           L     r7,136(,r13)
0002E2  5860  7000            148           L     r6,0(,r7)
0002E6  5870  ****            149           L     r7,=Q(func_ptr)
0002EA  58F4  7000            150           L     r15,0(r4,r7)
0002EE  05EF                  151           BALR  r14,r15...                 // ... respectively this instruction !
```

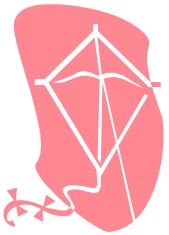
- Offset reference relative to start of failing ** function ** (not module) !

- Calculation:
  **X'280' + X'6E' = X'2EE' (BALR in funca)**
  ➔ caused by call of non-existent routine

# LE/C - TCP/IP Multiplexer capability

- The new **TCP/IP multiplexer configuration** relies on a table that assigns the name of an interface phase to a SYSID value
  - The TCP/IP stack is identified by a unique SYSID
  - Based on SYSID value the TCP/IP multiplexer can determine the TCP/IP interface phase that is to be used
- New messages **EDCT010S** (SYSID value)**, EDCT011S** (Interface LOAD), and **EDCV020W** (warning)
- Messages EDCT001S, EDCT002S, EDCV001I, and EDCV099I have been adjusted in LE/C TCP/IP multiplexer context
- For more details refer to "**Using Fast Path to Linux**" documentation in manual … **z/VSE V4R3.0 TCP/IP Support** ( SC34-2604-00 ), Chapter 4: http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/iestce40/4.1.7.1
- Be aware of **EDCTCPMC** skeleton in ICCF Library 62 (for configuration)

# Miscellaneous Enhancements



- **New messages**
  - **CEE3458E** (related to new CEE5MC2 Callable Service)
  - **CEE3519C** (related to CEEFETCH macro, CEEFTCH control block in use)
  - **CEE3559W** (error recovery with CICS when MSGFILE - LSTQ option is set)
- **User abend U4093**
  - Additional reason code 160 (X'A0') was introduced for attempts to implicitly create another enclave (explanation + specific programmer response)
- **Message U4083 (X'FF3')**
  - Has been refined by adding BEAR related debugging information
- **COBOL Return Codes to CICS**
  - Improved programmer response for return code 51001 (to ease problem follow-up)
- **LE/CICS run-time option defaults** (initial storage values) were changed for …
  - ANYHEAP, BELOWHEAP, HEAP, STACK
  - … to optimize storage use and reduce paging for **higher workloads** (taking into consideration the needs for CICS storage accounting areas (SAA))

# APARs Integrated in LE/VSE 1.4.7

- **APARS included in LE/VSE 1.4.7 base code** (for reference)
  - PK74672 LE/C -> from z/VSE 4.1 - CICS definitions for UCONV programs
  - PK71186 LE/Base -> from z/VSE 4.1 - PLI batch interactive debug gets 4083/RSN3
  - PK81297 LE/PLI -> from z/VSE 4.2 - ABEND 0C4/AKEA IN DT/VSE IVPPLI3
  - PK96905 LE/Base -> from z/VSE 4.1 - Illegal SVC 11 with pre-existing STXITAB
  - PK98714 LE/PLI -> from z/VSE 4.1 - CEE3200S + DTVSE listing variable content
  - PM00829 LE/COBOL -> from z/VSE 4.2 - App exception w/o condition handler
  - PM02293 LE/PLI -> from z/VSE 4.2 - IBM0482I ONCODE=310 fixed overflow cond.
  - PM05257 LE/COBOL -> from z/VSE 4.2 - IGZ0034W File could not be extended
  - PM06694 LE/C -> from z/VSE 4.2 - Prevent overlay using TAPE ENCRYPTION

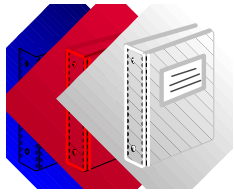
- **APARs included from LE for z/OS** (for reference)
  - PM05199 LE/C - Crash in DSECT utility
  - PQ66639 LE/Base - Abend 0C4 in CEECEOWN + Assembler pgm

# Available LE/VSE 1.4.7 APARs / PTFs

- **On top LE/VSE 1.4.7 APARS / PTFs …** ( for reference / recommended install option )
  - PM25751 / UK62331 LE/COBOL - HANDLING OF EMPTY VSAM KSDS/ESDS FILES NOT IN "LOAD-MODE" BROKEN
  - PM26031 / UK62596 LE/Base - LE/VSE BEAR EXPLOITATION FOR BAD BRANCH INFO TO LE/VSE DUMP
  - PM26532 / UK62594 LE/PLI - OPTIMIZE STORAGE DEMAND TO INITIALIZE PL/1 MULTITASKING ENVIRONMENT
  - PM38137 / UKyyyyy LE/Base - CEEFETCH macro corrections (dynamic call facility)

In contrary to integrated APARs on previous foil, above numbers will be visible in MSHP RETRACE (history file), when installed

# Documentation – Summary Reference –



- **Bird's eye view of new LE/VSE 1.4.7 functions**
  - VSE Release Guide, SC33-8300
- **CEE5MC2 - Callable Service for International Currency**
  - LE/VSE Programming Reference, SC33-6685
- **CEEFETCH macro** and **PLI Multitasking under LE/VSE (new chapter)**
  - LE/VSE Programming Guide, SC33-6684
- **Configuring the LE/C TCPIP Socket API Multiplexer (new section)**
  - LE/C Run-Time Library Reference, SC33-6689
- **LE/VSE Run-Time options changes, APAR integration, SVA configuration etc.**
  - LE/VSE Customization Guide, SC33-6682
- **LE/VSE Messages, Return Codes to CICS etc.**
  - LE/VSE Debugging and Run-Time Messages Guide, SC33-6681
  - Online Message File (OME) … hands-on option via VSE Console:
    - Enter: "CEE*, CEL*, IBM*, IGZ*, EDC*" prefixed message(s) => press PF9
    - For U4xxx abend codes, enter: "CEEABENDC" => press PF9
- **Languages + z/VSE**

Also see: z/VSE Basics, SC24–7436
-> Chapter 8, 9 and Appendix D
http://www.redbooks.ibm.com/abstracts/sg247436.html

# Hints and Tips for z/VSE 4.3 Migration ( 1 )

- **LE/VSE Attention Routine Interface + Commands ...**
  - The AR interface for LE/VSE is shipped pre-customized and activated !
  - Please **keep it enabled**, it suits to display LE/VSE run-time option, exit & status reports !
  - **Ensure** system ASI procedure USERBG is current in this context and contains the following VSE POWER statement:

```
// PWR PRELEASE RDR,CEEWARC LE - AR INTERFACE
```

  - This particularly applies if there is an equivalent or previously tailored version of this procedure in place. For system supplied USERBG.PROC version, please refer to skeleton SKUSERBG in ICCF library 59.
  - Associated sample report:

```
D CEE,CEESTAT
```

```
AR 0015 CEL4014I Current Language Environment for z/VSE Status Information
AR 0015 ------------------------------------------------------------------
AR 0015 CEL4011I Language Environment Version :  01.04.07  LE147GA
AR 0015 CEL4018I Report Produced by CEL4CMDR at  83AD0228  LE147GA
AR 0015
AR 0015 CEL4012I Language  Installed  SVA Resident  SVA Load Addr  APAR
AR 0015 CEL4013I LE/COBOL      YES         NO             -       LE147GA
AR 0015 CEL4013I LE/PLI        YES         NO             -       LE147GA
AR 0015 CEL4013I C/Locale      YES         YES        03B184E8    LE147GA
AR 0015 CEL4013I LE/C          YES         YES        03985400    LE147GA
  ...
```

# Hints and Tips for z/VSE 4.3 Migration ( 2 )

- **LE/VSE Run-time options ...**
  - LE/VSE checks if an old run-time option module (not matching latest release) is loaded !
  - If true, ABEND U4093 RSN42 is issued (for batch), respectively LE/VSE return code 11060 (at CICS initialization time).
  - Previously customized option modules (batch and CICS) therefore had to be reestablished (best via skeletons CEEWCOPT and CEEWDOPT from ICCF library 62).
  - There is **no need for action**, if default LE/VSE run-time options modules are used.

# Module 3

- **Tools Updates (Application Development/Debugging & more …)**
  - CEETRACE tool ( p.42-44 )
  - LEVSE_Control_Center – LECC ( p.45 )

# CEETRACE Facility – General

- **Optional Installable Debugging Facility**
  - "As-Is" supported tool (no optional product, no extra cost)
  - Complements the existing LE/VSE dump (CEE5DMP) capabilities
  - Not intended to replace the LE/VSE dump nor the Debug Tool for VSE/ESA
  - Capable to generate "execution statement history" for failing LE for z/VSE applications
  - Includes COBOL/VSE Source Code Extraction Utility (can produce original source code from load module that was compiled via SEPARATE sub-option of compiler TEST option)
  - Applications that do not abend will **not** produce an execution statement report.
  - The facility is new with z/VSE 4.2 and can be enabled via shipped member CEETRACE.Z
  - Activating the CEETRACE feature will have a negative impact on application performance dependent on options in use (assume ~10% at minimum)
  - For exploitation of the CEETRACE tool, z/VSE 4.2 users should refer to: http://www.ibm.com/servers/eserver/zseries/zvse/downloads/tools.html#ceetrace
- **A report similar to the following COBOL sample will be produced**


```
CEETRACE Program Execution Trace Report Begins
Date Time Program_Name Entry_Name Stmt# Stmt Offs Stmt_Lang Statement Source Code
30/06/2009 10:43:59.62 CELTCIVP Ent/Ext/Par N/A. +000005E8 COBOL External Entry/Exit point, End clause o
30/06/2009 10:43:59.62 CELTCIVP CELTCIVP 677 +000005EC COBOL DISPLAY 'CELTCIVP Begins '.
...
30/06/2009 10:43:59.79 CELTCIVP Ent/Ext/Par N/A. +00000634 COBOL External Entry/Exit point, End clause or Paragraph
30/06/2009 10:43:59.79 CELTCIVP CELTCIVP 683 +00000638 COBOL Display 'Call the CEETRACE feature reporting tool to
30/06/2009 10:43:59.79 CELTCIVP CELTCIVP 687 +0000066E COBOL Call 'CEL4RPRT'.
30/06/2009 10:43:59.88 CELTCIVP Ent/Ext/Par N/A. +000006A4 COBOL External Entry/Exit point, End clause or Paragraph
30/06/2009 10:43:59.88 CELTCIVP Ent/Ext/Par N/A. +000006A8 COBOL External Entry/Exit point, End clause or Paragraph
30/06/2009 10:43:59.88 CELTCIVP CELTCIVP 694 +000006AC COBOL MOVE 8 TO Vstring-length of IN-DATE.
30/06/2009 10:43:59.88 CELTCIVP CELTCIVP 695 +000006B6 COBOL MOVE '19/11/08' TO Vstring-text of IN-DATE 1:8
...
```

# Tools Update – CEETRACE ( V1R1.2 ) –

- Available for download on z/VSE Home (CEETRACE V1R1.2)
  http://www-03.ibm.com/systems/z/os/zvse/downloads/tools.html#ceetrace
- **New configuration option** available in the CEETRACE.INI file:
  – Customer requirement for: "**include_part=**"


- The **"Program Execution" trace report** heading now includes both the
  – CEETRACE and LE for z/VSE version information
- **CEETRACE reporter - CEL4RPRT** has been improved to more effectively detect if …
  – Debug Tool for VSE/VSE is present
  – Then exit immediately (use is mutually exclusive)
- CEETRACE, can be utilized in **Break Event Address Register** and **PL/1 multitasking** environments (with benefit of additional debugging assist information) !
  – BEAR example follows on next page …

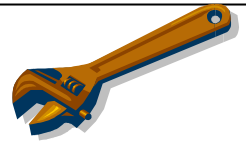
# Tools Update – CEETRACE ( V1R1.2 ) –

- When invoked on BEAR hardware, the CEETRACE report will include the absolute address information associated with a bad branch …

```
CEETRACE V01.01.02 - Program Execution Trace Report Begins.   CEETRACE Using LE z/VSE Version 01 Release 04.07

  Date         Time      Program_Name Entry_Name  Stmt#  Stmt Offs  Stmt_Lang    Statement Source Code
 10/21/2010 11:19:07.33  LE8IGZXR     Ent/Ext/Par  N/A.  +000001C0    COBOL    External Entry/Exit point, End clause or Paragraph
 10/21/2010 11:19:07.33  LE8IGZXR     LE8IGZXR     15    +000001C4    COBOL    Program not compiled with the SEP option.

CEETRACE      Message associated with current condition is :
CEE3201S The system detected an operation exception.

CEETRACE       Breaking Event Address Reg : 00420264
CEETRACE      Program Execution Trace Report Complete

CEE3201S The system detected an operation exception.
         From compile unit LE8IGZXR at entry point LE8IGZXR at compile unit offset -00420078 at address 00000000.
         Possible Bad Branch:  Statement: 15  Offset: +000001EC
CEE5DMP V1 R4.7: Condition processing resulted in the unhandled condition.     10/21/10 11:19:07 AM     Page: 1

Information for enclave LE8IGZXR

  Information for thread 8000000000000000

  Traceback:
    DSA Addr  Program Unit  PU Addr   PU Offset  Entry     E Addr    E  Offset   Statement  Service  Status
    0053D018  CEEHDSP       04E2EEB0  +00002BC8  CEEHDSP   04E2EEB0  +00002BC8              L147GA   Call
    00421920  LE8IGZXR      00420078  +00000000  LE8IGZXR  00420078  +00000000                       Exception
```

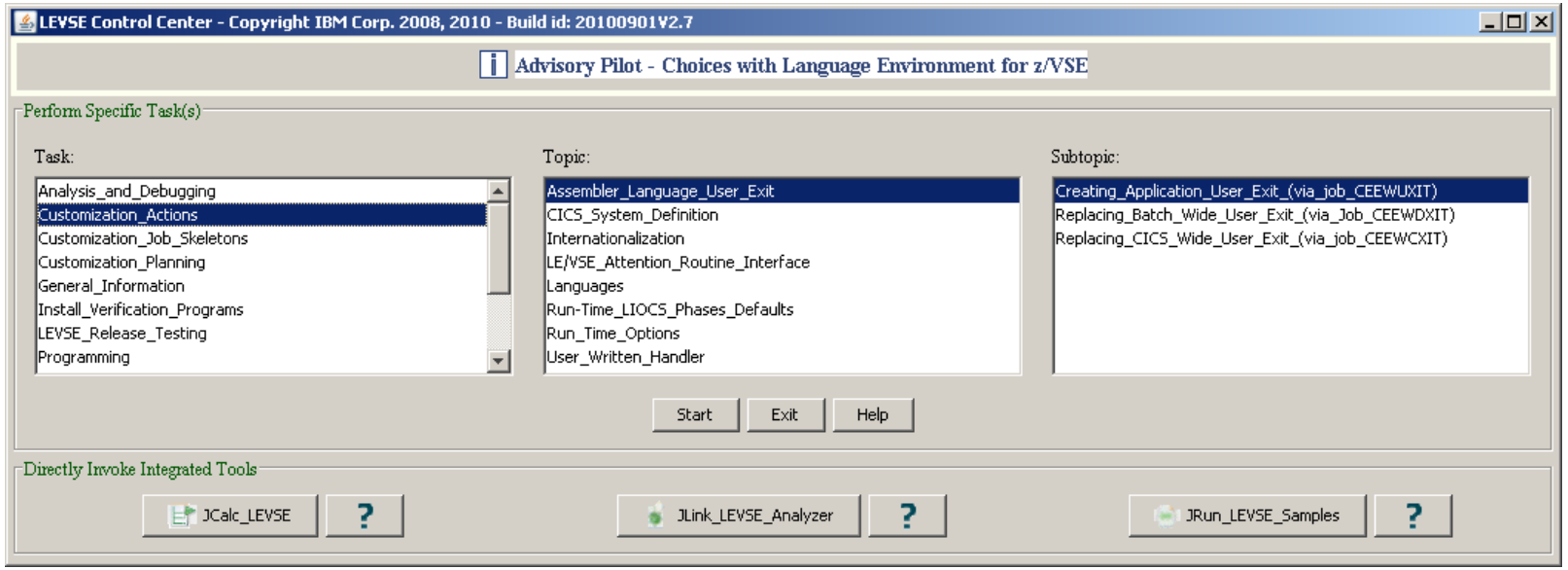
Calculation:
**BEAR X'420264' – X'420078' = X'1EC'**
➔ Bad branch offset in LE8IGZXR

# Tools Update – LEVSE Control Center ( LECC V2.7 ) –



- "As-Is" supported tool for **common z/VSE language tasks,** functional trials, simple AD frame
- **Download option via** z/VSE Home (Build-id: 20100901V2.7) … please check readme + .pdfs
  http://www-03.ibm.com/systems/z/os/zvse/downloads/tools.html#lecc
- **Includes existing LE/VSE web tools** (in latest version):
  - JCalc_LEVSE: calculate and view sets of predefined LE/VSE SVA-eligible routines
  - JLink_LEVSE_Analyzer: check state of VSE apps by inspecting their compile/link lists
  - JRun_LEVSE_Samples: try out shipped LE/VSE Callable Service samples (3 languages)
- Offers the option for **direct, integrated tools invocation** from main window !


- There is a **zJournal Article** providing details on **"LECC" tool capabilities**
  http://www.mainframezone.com/it-management/lecc-a-new-tool-for-managing-languages-and-applications-in-z-vse



# Questions ?

- Any questions ?

# Contact Options

- **Please select dependent on purpose:**
  - Questions specific to this LVC presentation: wbosch@de.ibm.com
  - LE/VSE PMRs or service related issues: vsesupportLE@de.ibm.com
  - General VSE questions and comments: zvse@de.ibm.com
  - Direct submit option for VSE Requirements: https://www-03.ibm.com/systems/z/os/zvse/contact/requirement.html