








Understanding
LE/VSEⁿ Run-Time Options
and
The Impact of Applying Service
on VSEⁿ libraries

Laura Grinham
VSEⁿ Technical Support Senior Software Engineer



Agenda



-  VSEⁿ Language Environment Run-Time Options
-  Determining Current Run-Time Options
-  OLTP and Data Base Considerations
-  Customising Run-Time Options Default Modules
-  Using Run-Time Options with a non-LE Environment
-  Preservation of Customised Members in System Libraries
-  References



VSEⁿ Language Environment Run-Time Options



LE/VSEⁿ:

Single environment

Common functions and callable services

High Level Languages – C, COBOL, PL1 & HLASM

Calling/nesting modules may be written in different language

Manages application initiation/termination

Customised environment with use of run-time options

More than 30 run-time options – see LE/VSEⁿ Programming Reference

COUNTRY: country code with corresponding default values for currency symbol, thousands separator and date/time formats

NATLANG: language used for error messages and month names



VSEⁿ Language Environment Run-Time Options



TEST(<<hook>>,<<sym>>):

establishes conditions used for debugging applications with a debugging aid, options for hook locations, e. g. none, statement, entry points and whether the symbol table is printed in the dump

Many options for Abend Handling:

ABPERC, **ABTERMENC**, **TERMTHDACT**, **TRACE** & **TRAP** controls what information is provided in a dump and under what conditions the dump is triggered.

TERMTHDACT has 3 sub-options for determining the type of data dumped, the destination for the dump output and the amount of storage before/after the general registers



VSEⁿ Language Environment Run-Time Options



Storage Management options:

STORAGE: initial value of acquired storage and a value to be overwritten on storage that is freed.

OLTP considerations for 'Out of Storage' parameter

LIBSTACK, BELOWHEAP: Allocate storage below the 16 Mb line

ANYHEAP, HEAP, STACK:

Amount of storage initially acquired and subsequent increments

Required location for acquired storage – above/below 16Mb or any

RPTSTG: Generates a list of the storage used by an application during execution. Review this list to determine if the storage related run-time options could be tweaked to improve performance.



VSEⁿ Language Environment Run-Time Options (cont)



Language specific options: **RTEREUS** Intended for COBOL, do not use for other high-level languages and is Ignored by OLTP establishes a reusable run time environment

Compiling and Linking options:

TEST run-time option with the hook sub-option set to STMT with a COBOL program compiled with the DATEPROC(FLAG) compiler option displays a message whenever a date sensitive field is accessed

COBOL programs executing with run-time option of **TEST** with a specified HOOK location are not optimized which may override the OPTIMIZE compiler option.



VSEⁿ Language Environment Run-Time Options (cont)



Applications are generated with AMODE/RMODE attributes:

AMODE is the addressing mode (24-bit, 31-bit or any bit-addressing)

RMODE is the residence mode (below 16Mb, any location above/below 16Mb)

ALL31: ON - application runs in 31-bit addressing,
OFF - some part must run in 24-bit addressing.

LE/VSEⁿ's callable services run in 31-bit addressing mode

Automatic mode switching occurs when an application running in 24-bit addressing requests an LE callable service and **ALL31** is set to OFF.

To avoid automatic mode switching and therefore improve performance, re-link the program to run in 31-bit addressing mode and set **ALL31** to ON

Examples of callable services:

CEE5CTY – set default country

CEE5USR – set/query user area



VSEⁿ Language Environment Run-Time Options (cont)



Re-entrancy and AMODE/RMODE

Application programs can be compiled/linked and executed in many combinations of AMODE/RMODE being above/below the line.

COBOL programs can also specify the location of data used for working storage with the compile option DATA(24/31)

This may create conflicts with the **ALL31** run-time option, for example:

COBOL programs compiled with the NORENT compiler option are by default, linked AMODE(ANY) and can execute in either 24-bit or 31-bit addressing mode which conflicts with **ALL31(ON)**.

Resolve conflict by re-linking as AMODE(31)

SVA Considerations: Review AMODE/RMODE attributes of SVA phases with the amount of specified/available SVA space above and below the 16Mb line.



VSEⁿ Language Environment Run-Time Options (cont)



RPTOPTS: Produces list of the run-time options in effect during execution
Check production applications for any run-time options which are intended for use during development and may negatively impact performance
Running with **RPTOPTS(ON)** increases execution time, use in development for analysis, remove before migration to production

LE/VSEⁿ run-time options may be set in:

- Installation defaults modules
- Application default module
- Attention Routine Interface command
- Assembler user exit
- OLTP transaction CLER
- JCL parameters
- Source code statements



VSEⁿ Language Environment Run-Time Options (cont)



Installation defaults:

CEEDOPT for Batch
CEECOPT for OLTP

Application defaults:

Linked into individual programs: CEEUOPT

For performance reasons, only include those values in CEEUOPT which are to be changed from the default values provided by CEEDOPT and CEECOPT

Attention Routine Interface command

Batch processing – Clear/update run-time options with the following:

```
S CEE,OPTS=option1,option2,....
```

```
S CEE,CLEAR
```



VSEⁿ Language Environment Run-Time Options (cont)



Assembler user exit:

CEEBXITA for Batch

CEECXITA for OLTP

Specifies the parameters for the LE enclave before the enclave is established

Written in Assembler as no HLL environment available at the point

During enclave initialization any run-time options in CEEBXITA are processed

Always overrides the values set in CEEUOPT

Only overrides the values set in CEEDOPT & CEECOPT if OVR specified in
default modules CEEDOPT/CEECOPT



VSEⁿ Language Environment Run-Time Options (cont)



OLTP transaction CLER

Displays current run-time options for this instance of OLTP

Values on the OLTP terminal can be over-written and updated (PF5)

```
CLER

                Language Environment for VSEn V1 R5.0
                Dynamic OLTP Run-time Option Modification Facility

ABPERC:      ( NONE )           STORAGE:  ( 00 , NONE , NONE , 0
ABTERMENC:   ( ABEND )         TERMTHD: ( TRACE , MSGFL , 0 )
ALL31:       ( ON )           TRAP:    ( ON , MAX )
CBLPSHPOP:   ( ON )           USRHDLR: ( )
CHECK:       ( OFF )          TERMTHDACT LSTQ OPTIONS :
DEBUG:       ( OFF )          CLASS:  ( L )   DISPOSITION: ( D )
DEPTHCOND:   ( 10 )          NODE-ID: ( )
ERRCOUNT:   ( 20 )          USER-ID: ( )
HEAP:        ( 4080 , 4080 , ANYWHERE , KEEP , 4096 , 4080 )
LIBSTACK:    ( 4096 , 4080 , FREE )
MSGFILE:     ( CESE )         HEAPCHK: ( OFF , 1 , 0 )
RPTOPTS:     ( OFF )
RPTSTG:      ( OFF )
STACK:       ( 4080 , 4080 , ANYWHERE , KEEP )

                Please change the runtime option to the setting required.

<PF3> Exit  <PF4> Restore Defaults  <PF5> Update  <PF12> Exit(No Change)
```



VSEⁿ Language Environment Run-Time Options (cont)



JCL parameters:

Options written on EXEC statement, separated from program parameters by '/':

C for VSEⁿ & PL/I for VSEⁿ, coded as

```
// EXEC.....PARM='run time options/program parameters'
```

The run-time option **EXECOPS/NOEXECOPS** determines whether the run-time options can be passed to C for VSEⁿ or PL/I for VSEⁿ programs as parameters on the EXEC statement. If NOEXECOPS use #Pragma or PLIXOPT

COBOL for VSEⁿ, for compatibility with VS COBOLII programs, order determined by **CBLOPTS** run-time option:

```
CBLOPTS(OFF), same as C & PL/I above
```

```
CBLOPTS(ON) // EXEC.....PARM='program parameters/run-time options'
```



VSEⁿ Language Environment Run-Time Options (cont)



Source Code Statements:

```
C for VSEn,           # pragma runopts (option, option,...),  
PL/I for VSEn,       DCL PLIXOPT char (nn) ... init('option option')
```

Considerations for Dump Analysis:

Carefully review the use of APBERC, ABTERMENC, RPTSTG, TERMTHDACT, TRACE & TRAP options. Use the correct combination of parameters to produce the required dump information, e.g.

TRAP(on,max) must be set to honour the values for ABTERMENC

TERMTHDACT(quiet/msg/trace/dump/uadump) generates different dumps



Determining Current Run-Time Options



To confirm the LE Run-Time options at the time the programs are running:

- Installation-wide defaults – Batch:
enter D CEE,CEEDOPT on the VSEⁿ console
- Installation-wide defaults – OLTP:
enter D CEE,CEECOPT on the VSEⁿ console
USE OLTP transaction code ROPC, report is written to VSEⁿ console
- OLTP session:
USE OLTP transaction code CLER – as discussed previously
- Individual programs:
add the RPTOPTS(ON) run-time parameter on the EXEC JCL statement



OLTP and Data Base Considerations



You cannot specify run-time options for HDB, SDB or OLTP applications as parameters on the JCL EXEC statement, use any of the other methods listed above

To specify run-time options for applications written in C for VSEⁿ with HDB & SDB, code the #pragma statement as **#pragma runopts(plist(os),env(DLI),noexecops)**

The run-time option, **CBLPSHPOP** intended for COBOL for VSEⁿ OLTP applications to allow compatibility with older COBOL programs that contain *some* conditional handling commands. By default, set to OFF for Batch, set to ON for OLTP

- If set to CBLPSHPOP(on), when a COBOL for VSEⁿ program or subroutine is called/terminated, an EXEC CICS PUSH/POP HANDLE is issued

Running with CBLPSHPOP(on), negatively impacts performance, if your OLTP system doesn't have older COBOL programs, override the default value.



OLTP and Data Base Considerations



By default, LE/VSEⁿ run-time messages generated by OLTP are written to the Transient Data Queue CESE

The report output from RPTOPTS(on) may overflow the Transient Data Q.
Alternatively, use transaction ROPC to write to VSEⁿ console

More about report output and destinations..... later.



Customising Run-Time Options Default Modules



Batch Installation Default module

Macro Source Code: CEEDOPT.a

- csect CEEDOPT, assembled with macro CEEXOPT

 - specifies installation-wide defaults for all run-time options except C-specific options which can only be set with #pragma runopts, e.g. argparse

 - OVR/NOOVR parm indicates default value can be overridden

- includes CEELSTQ section, assembled with macro CEELOPT

 - specifies default values for POWER/VSEⁿ LST statement



Customising Run-Time Options Default Modules



OLTP Installation Default module

Macro Source Code: CEECOPT.a

csect CEEDOPT, macro CEEXOPT

same as above, default values optimised for OLTP

Application/User Default module

Macro Source Code: CEEUOPT.a

csect CEEDOPT, macro CEEXOPT

same as CEEDOPT.a above, except no OVR/NOOVR parameter specifies user's application-specific run-time options

takes precedence over installation default options specified as OVR



Customising Run-Time Options Default Modules



Installation Default modules CEEDOPT and CEECOPT include CEELOPT macro default POWER LST Q parms which specify destination for dump output
Class, Disposition, Node and User ID

Back to TERMTHDACT - a parameter determines output destination
set to 'msgfile' (default) or 'LSTQ'

msgfile is also a run-time option - filename for runtime diagnostics
set to 'syslst' (default for batch)
or 'CESE' transient data queue (default for OLTP).

By default, OLTP dump output will probably fill the CESE transient data Q, batch dump output sent to syslst – not very manageable.

A combination of TERMTHDACT → LSTQ, CEELOPT → POWER values in CEEDOPT & CEECOPT, dump output will be sent to specified user.



Customising Run-Time Options Default Modules



The LE/VSEⁿ library PRD2.sceebase contain job control to generate the default object deck/modules.

JCL Code:

CEEWDOPT.z & CEEWCOPT.z contains JCL to:

- Assemble CEECOPT.a & CEEDOPT.a source and catalog object modules
- Linkedit object modules & catalog phases into VSEⁿ library
- Optionally load/reload phases into SVA

CEEWUOPT.z contains JCL to:

- Assemble CEEUOPT.a source and catalog the assembled object module
- object module to be linked into user program phase

CEEXUOPT.z same as CEEWUOPT.z,
contains specific run-time options to generate dumps



Customising Run-Time Options Default Modules (cont)



Link books:

CEE\$COPT.obj to create CEECOPT.phase

CEE\$DOPT.obj to create CEEDOPT.phase

There are corresponding members in ICCF library # 62

The updated installation default phases can be reloaded in SVA, e.g.

```
// libdef phase,search=PRD2.sceebase
set sdl
CEE$COPT,sva
/*
```

To use a newly assembled CEECOPT module without restarting OLTP, use OLTP transaction NEWC. Start the NEWC transaction in each OLTP system.

Note: this is not the same as OLTP transaction CEMT SET PROG CEECOPT NEWC



Using Run-Time Options with a non-LE Environment



Pre-initialization services

- LE applications can be called from a non-LE environment
- To prevent the initiation/termination of the LE-environment each time the application runs/ends, Pre-Initialization services establish the LE-environment once
- This provides better performance when for example, a non-LE application written in Assembler frequently calls an LE language application/subroutine



Using Run-Time Options with a non-LE Environment



The Pre-initialization services use the CEEPIPI interface

- Applications are loaded into the CEEPIPI table by the CEEPIPI(add_entry) request
- Calls to a main application is performed by the CEEPIPI(call_main) request
- Calls to a subroutine is performed by the CEEPIPI(init_sub) request
- Run-Time options are passed in a fixed length string of 255 characters as a parameter on the CEEPIPI(call_main) or CEEPIPI(init_sub) request
- These run-time options always override options specified in EXEC statement

CEEPIPI is only for BATCH, OLTP applications can use
EXEC CICS LINK or EXEC CICS XCTL



Preservation of Customised Members in System Libraries



Please be very aware that the FSU process which applies a Service Refresh to your VSEⁿ system:

- Deletes and recreates the system libraries:
IJSYSRS.SYSLIB
PRD1.BASE, PRD1.MACLIB
PRD2.SCEEBASE, PRD2.TCPIPB, PRDTCPIPC & PRD2.GEN (if Gen. feature)
- Uses the delta libraries - PRD1.BASED, PRD1.MACLIBD & PRD2.SCEEBASD
& PRD2.SAVE

Don't save any phases in PRD2.SAVE as all phases in PRD2.SAVE are copied to IJSYSRS.SYSLIB during the FSU process

The FSU process updates system data but does not delete any user data, hardware configuration tables or installed products.



Preservation of Customised Members in System Libraries (cont)



All **customized** members of 21CS products should be saved in **PRD2.CONFIG** or another **user-defined** VSEⁿ library to ensure they are available after the FSU. This includes VSEⁿ Run-time Options modules, e.g. CEEDOPT, POWER VSEⁿ phases, standard labels and printer control routines (LIOCS). If you need to have customized FCBs in PRD2.save, take backup copy to be restored after FSU.

Also be aware, some skeletons' JCL create a copy of the original member in PRD2.save, e.g. IPL procs and standard labels. Determine if these members need restoring after the FSU and take backups as necessary

After IPLing the refreshed system, any customized phases you restored that were previously in the SVA, can be reloaded in the SVA, e.g.

```
// libdef phase,search=<PRD2.CONFIG> or <usrlib.savelib>  
set sdl  
<phasename>,sva  
/*
```



Preservation of Customised Members in System Libraries (cont)



- ICCF system libraries (1, 2, 50 - 69) should not contain user members.
- Libraries 8, 9 & 10 are by default reserved for the Operator, Programmer & Administrator profiles, respectively.
- User members including modified skeletons can be saved in the (unassigned) User libraries i.e., # 3-7, 11-49, 70-199.
- The (unassigned) User libraries, # 3-7, 11-49, 70-199 are initially empty.
- Any members in the system libraries which are modified e.g. skeletons in library 59, should be saved in user libraries because members in system libraries may be lost/replaced when service is applied.

There is a table of library #s and their use in the Planning manual.

Also see VSEⁿ/ICCF Library Considerations in the Administration manual.



References

LE/VSEⁿ Customization Guide

LE/VSEⁿ Programming Guide

LE/VSEⁿ Programming Reference

LE for VSEⁿ Debugging Guide and Run-Time Messages

VSEⁿ Administration

VSEⁿ Planning

VSEⁿ System Upgrade and Service



Thank You