z/OS Language Environment Setup & Customization

March 2006

Thomas Petrolino
IBM Poughkeepsie
tapetro@us.ibm.com

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

- CICS®
- IMS
- Language Environment®
- OS/390®
- z/OS®

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
UNIX is a registered trademark of The Open Group in the United States and other countries.
SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* 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.

Reflection

- "Language Environment is 100% compatible with previous Run-Time environments" - An unnamed, misinformed marketing rep
- "Read the Language Environment Migration Guide(s) for details of migration problems you **WILL** encounter." - An unnamed Language Environment developer

Agenda

- Introduction
- Which Libraries? When?
- User Exits
- Setting Run-Time Options
- Key Run-Time Options
- Additional Information
- Appendix: AMODE 64 Run-Time Options

Introduction

- Language Environment is made up of several components
  - Common Execution Library (CEL) 568819801
  - COBOL Run-Time Library 568819802
  - PL/I Run-Time Library 568819803
  - FORTRAN Run-Time Library 568819804
  - C/C++ Run-Time Library 568819805
  - Enterprise PL/I (a.k.a. VisualAge PL/I) 568819806

Supported Releases

Supported Release Level	FMID	Support Withdrawn
z/OS V1.4	HLE7707	3/31/2007
z/OS V1.5	HLE7708	3/31/2007
z/OS V1.6	HLE7709	
z/OS V1.7	HLE7720	

Introduction

- Upward Compatibility
  - All releases of Language Environment are upward compatible. Applications compiled/linked with a particular release of Language Environment will run on that Language Environment level or higher (see APAR II11316)

Introduction

- Downward Compatibility
  - Starting with OS/390 2.10 Language Environment, applications can be compiled/linked at OS/390 2.10 Language Environment (or higher) and run on lower levels of Language Environment with downward compatibility toleration APARs.
  - New function is excluded. You can only use function available on the lowest level target system. In many cases CEE3728S message will be received when this is attempted.

Introduction

- Many z/OS elements, features, and functions require Language Environment to be at the level shipped with that release of z/OS...
  - BookManager, C/C++ compiler & class libraries, Communications Server (IP & SNA services), Cryptographic Services, DFSMS (CDRA), DCE Base Services, Distributed File Service, DFSORT Locale Processing, IBM HTTP Server Base, Infoprint Server, Integrated Security Services (DCE & LDAP Server), Library Server, msys for Operations, msys for Setup, NSF, OSA/SF, RMF, z/OS UNIX System Services, etc.
- See z/OS Program Directory section "LNKLSTxx Considerations" under "PARMLIB Member Considerations" section for additional information

Which Libraries? When?

- Customization
  - SCEESAMP
- C/C++ compile steps
  - SCEEH.* in SYSLIB
- C/C++ XPLINK Bind step
  - SCEEBND2 in SYSLIB
  - SCEELIB in SYSLIN

Which Libraries? When?

- Non-XPLINK Prelink/Link step
  - SCEEOBJ (UNIX System Services), SCEECPP (C++) in prelink SYSLIB
  - SCEELKED in SYSLIB
  - SCEELIB in SYSLIN (for z/OS V1.2 C++ Class Library)
- Non-XPLINK Bind step
  - SCEELKEX then SCEELKED in SYSLIB
    - SCEELKEX not needed for Enterprise PL/I
  - SCEELIB in SYSLIN (for z/OS V1.2 C++ Class Library)

Which Libraries? When?

- Run/Go step
  - SCEERUN in STEPLIB/LPA/LNKLST
    - SCEELPA in LPALSTxx
  - SCEERUN2 in STEPLIB/LNKLST for C/C++ XPLINK
    - Cannot be in LPALSTxx since PDSE
    - Can use Dynamic LPA to add these modules
  - Remove all pre-Language Environment runtime libraries from search order

Which Libraries? When?

- Run/Go step *(continued)*
  - CICS TS...
    - SCEECICS (COBOL) then SCEERUN in DFHRPL
    - SCEERUN in STEPLIB (if not in LNKLST/LPALST)
    - CSD must be updated to include the definitions in SCEESAMP (autoinstall will do this)
      - CEECCSD
      - CEECCSDX (for CICS TS 3.1 XPLINK support)
    - Create CESE/CESO/CIGZ transient data queues using CEECDCT in SCEESAMP

User Exits

- Allows customization of Language Environment initialization and termination behavior
- When linked with Language Environment, acts as a system wide exit
- When linked with an application, affects only that application
  - Supersedes exit linked with Language Environment

User Exits

- CEEBXITA – Assembler user exit
- CEEBINT – High-level language (HLL) user exit
- CEEEXTAN – Abnormal termination exit
- CEEBLNUE – Load notification user exit
- CEEBSTX/CEECSTX – Storage tuning user exit

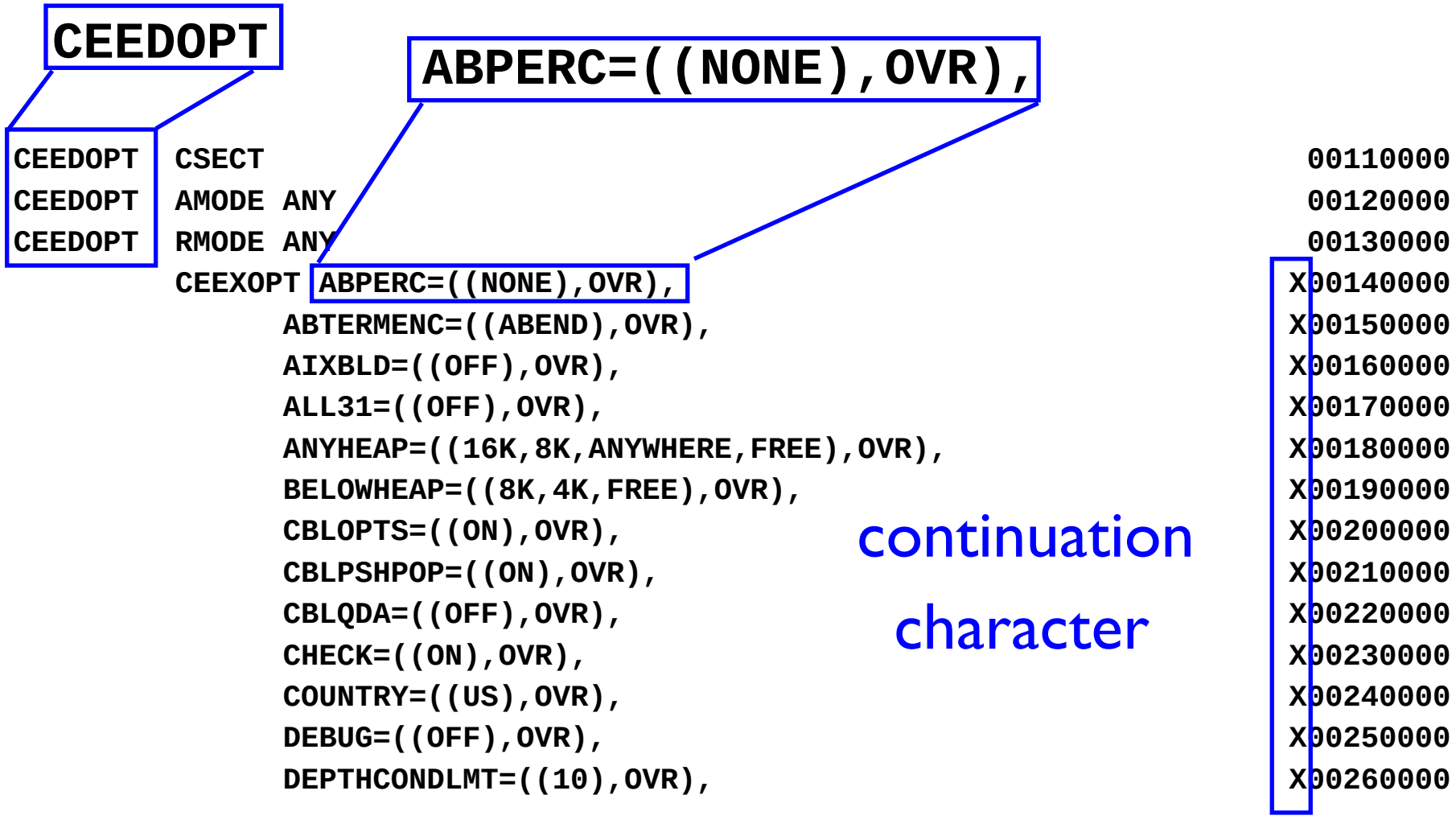
Setting Run-Time Options

- Installation defaults (CEEDOPT/CEECOPT/CELQDOPT)
  - Also referred to as system-wide defaults
  - SMP/E USERMOD to Language Environment modules
    - SCEESAMP(CEEWCOPT) - CICS TS only
    - SCEESAMP(CEEWDOPT) - non-CICS environments
    - SCEESAMP(CEEWQDOP) - AMODE 64
  - All options must be specified

Setting Run-Time Options

- Installation Defaults *(continued)*
  - To change Installation defaults
    1. uninstall USERMOD, if already installed
    2. modify the options
    3. install/reinstall the USERMOD
  - Note: Reinstall may be needed if run-time options are added or changed in the service stream. (++HOLD(ACTION) will be used on these PTFs to identify when this needs to be done).

Setting Run-Time Options



Setting Run-Time Options

- System defaults
  - New for z/OS V1.7
  - Options may be specified in a PARMLIB member
    - CEEPRMxx
  - Options may be specified with an operator command
    - SETCEE
  - Reduces the need to maintain USERMODs for CEEDOPT/CEECOPT/CELQDOPT

Setting Run-Time Options

- System defaults *(continued)*
  - Specifying options in PARMLIB member
    - Member name CEEPRMxx
    - Options specified in groups
      - CEEDOPT
      - CEECOPT
      - CELQDOPT

Setting Run-Time Options

- System defaults *(continued)*
  - Sample CEEPRMxx

```
CEEDOPT(ABPERC(NONE) ALL31(ON)
        rptopts(on) )   /* Options report */

CEECOPT(anyheap(4k,4080,anywhere,free))

CEEDOPT(ALL31(OFF) ) /* Turn off this option */
```

Setting Run-Time Options

- System defaults *(continued)*
  - Using the CEEPRMxx member
    - At IPL
      - Add CEE=xx to IEASYSxx member
      - Use CEE=xx at system parameter prompt
        - R 0,SYSP=AA,CEE=xx
    - After IPL, can select a different CEEPRM member
      - SET CEE=yy command

Setting Run-Time Options

- System defaults *(continued)*
  - Using the SETCEE system command
    - Overrides the current system defaults
    - Usage
      - Specify one group per command
      - Up to 126 characters
      - Example:

```
SETCEE ceedopt,rptstg(on),rptopts(on)
```

Setting Run-Time Options

- System defaults *(continued)*
  - Displaying the system defaults
    - D CEE displays the active members
    - Examples:

```
d cee
CEE3744I 17.56.05 DISPLAY
NO MEMBERS SPECIFIED
```

or

```
d cee
CEE3744I 17.57.31 DISPLAY
CEE=(JM)
```

Setting Run-Time Options

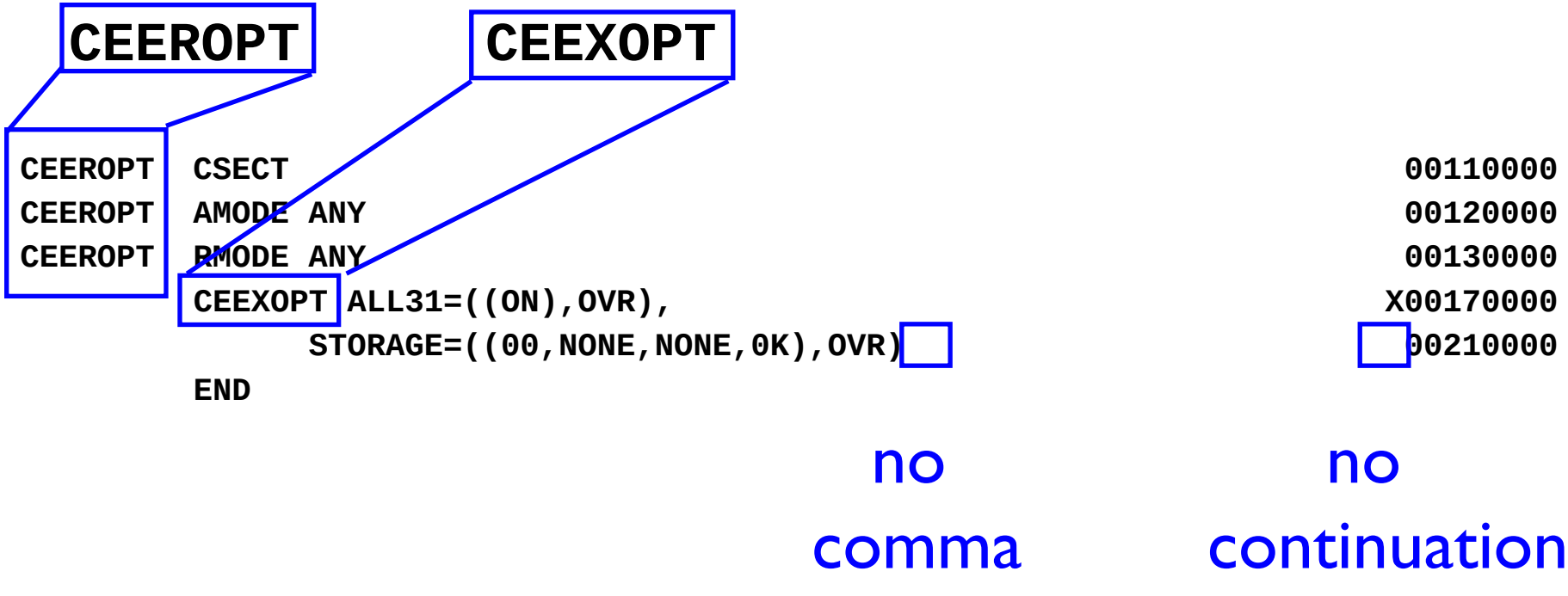
- System defaults *(continued)*
  - Displaying the system defaults
    - D CEE,groupname displays the options for a particular group
    - Example:

```
d cee,ceedopt
 CEE3745I 17.59.44 DISPLAY CEEDOPT
 CEE=(01)
 LAST WHERE SET  OPTION
 ------------------------------------------
 CEEPRM01        ENVAR("testing=roger","verify=1 2 3")
 CEEPRM01        HEAP(4194304,5242880,ANYWHERE,KEEP,
                      16384,16384)
 CEEPRM01        PROFILE(OFF,"XXX")
 CEEPRM01        RPTOPTS(ON)
```

Setting Run-Time Options

- Region Level Overrides (CEEROPT)
  - CICS TS and LRR users (e.g. IMS) only
  - Separate load module dynamically loaded at run-time during region initialization
    - SCEESAMP(CEEWROPT)
    - Must be found in search order, such as STEPLIB for IMS or DFHRPL for CICS TS
  - Specify only those options you wish to change

Setting Run-Time Options



Setting Run-Time Options

- Region Level Overrides (CEEROPT) *(continued)*
  - Certain options can be overridden dynamically in CICS TS region via CLER transaction
    - ALL31
    - CBLPSHPOP
    - RPTOPTS
    - RPTSTG
    - TERMTHDACT
    - TRAP

Setting Run-Time Options

- Application Level Overrides (CEEUOPT/CELQUOPT)
  - CSECT linked with the application
    - SCEESAMP(CEEWUOPT/CEEWQUOP)
  - Specify only those options you wish to change
  - Note: It is recommended to use options at this level when specific options are needed for an application. It is not a good idea to rely on installation defaults. This is especially important for vendors.

Setting Run-Time Options

CEEUOPT

```
CEEUOPT  CSECT                                              00110000
CEEUOPT  AMODE ANY                                          00120000
CEEUOPT  RMODE ANY                                          00130000
         CEEXOPT HEAP=(10M,10M,ANYWHERE,FREE),              X00180000
               STACK=(1M,1M,ANYWHERE,KEEP)                  00250000
         END
```

no "OVR" or outer parens

Setting Run-Time Options

- Programmer Overrides
  - Compiled into program
    - #pragma runopts for C/C++
    - PLIXOPT for PL/I
    - not available for COBOL
  - Internally generates CEEUOPT/CELQUOPT
  - Specify all program requirements, never ever rely on installation/system/region defaults. This is especially important for vendors.

Setting Run-Time Options

- Programmer Overrides *(continued)*
  - For example
    - #pragma runopts(ALL31(ON),ERRCOUNT(0),\
      STORAGE(NONE,NONE,NONE,30K),\
      TRAP(ON,SPIE),LIBSTACK(4K,4K,FREE),\
      STACK(2M,1M,ANYWHERE,KEEP),\
      ANYHEAP(16K,8K,ANYWHERE,FREE),\
      BELOWHEAP(8K,4K,FREE),\
      HEAP(1M,500K,ANYWHERE,KEEP))

Setting Run-Time Options

- Programmer Overrides *(continued)*
  - For example
    - DCL PLIXOPT CHAR(140) VAR
      INIT('ALL31(ON) ERRCOUNT(0)
      STORAGE(NONE,NONE,NONE,30K)
      TRAP(ON,SPIE) LIBSTACK(4K,4K,FREE)
      STACK(2M,1M,ANYWHERE,KEEP)
      ANYHEAP(16K,8K,ANYWHERE,FREE)
      BELOWHEAP(8K,4K,FREE)
      HEAP(1M,500K,ANYWHERE,KEEP)')
      STATIC EXTERNAL;

Setting Run-Time Options

- Program Invocation Overrides
  - In UNIX System Services shell (case sensitive)
    - export _CEE_RUNOPTS='run-time options'
      - No application parameters allowed, thus '/' not needed
        - export _CEE_RUNOPTS='TERM(UADUMP)'
      - Can also contain AMODE 64 run-time options
        - Only applicable options are used, but propagation of all options are done for spawn()ed and exec()ed programs

Setting Run-Time Options

- Program Invocation Overrides *(continued)*
  - In batch, on EXEC card
    - COBOL (with CBLOPTS(ON))
      - PARM='program arguments/run-time options'
    - C/C++, PL/I, FORTRAN, Language Environment-conforming Assembler
      - PARM='run-time options/program arguments'
    - First program must be Language Environment-conforming
    - The slash is required to delineate the run-time options, even when no program arguments.

Setting Run-Time Options

- Program Invocation Overrides *(continued)*
  - In batch, on EXEC card *(continued)*
    - EXEC PGM=MYCOBAPP,PARM='TODAY/TER(UADUMP)'
    - EXEC PGM=MYCOBAPP,PARM='10/27/00'
      - The entire parameter is considered to be application arguments because there are no valid run-time options after the last forward-slash.
    - Note that PARM= is limited to 100 characters

Setting Run-Time Options

- DD:CEEOPTS Overrides
  - New for z/OS V1.7
  - Optional data set in which run-time options may be specified
  - Allows up to 3K characters
  - Allows run-time options to be passed to non-Language Environment conforming main routines

Setting Run-Time Options

- DD:CEEOPTS Overrides *(continued)*
  - Usage: Add or allocate a CEEOPTS DD
    - As in-stream data:

```
//MYAPPL01 EXEC PROG=MYPRG,PARM='RPTOPTS(ON)/'
//CEEOPTS  DD *
* THESE ARE MY OPTIONS:
ALL31(ON), HEAP(64K),
ENVAR("JOHN=MONTI"),
TERMTHDACT(UADUMP)
/*
```

Setting Run-Time Options

- DD:CEEOPTS Overrides *(continued)*
  - Usage: Add or allocate a CEEOPTS DD
    - As a sequential file:

```
//MYAPPL01 EXEC PROG=MYCPROG,PARM='RPTOPTS(ON)/'
//CEEOPTS  DD DSN=JMONTI.MYCPROG.LEOPTS
```

    - As a member of a PDS:

```
//MYAPPL01 EXEC PROG=MYCPROG,PARM='RPTOPTS(ON)/'
//CEEOPTS  DD DSN=JMONTI.MYCPROG(LEOPTS)
```

Setting Run-Time Options

- DD:CEEOPTS Overrides *(continued)*
  - Supported under:
    - Batch
    - TSO
    - Preinit (CEEPIPI) during call_main or init_sub(dp)
    - IMS without LRR
    - DB2
  - Not supported under:
    - CICS
    - IMS/LRR
    - Preinit Compatibility (PICI)
    - System Program C (SPC)
    - Exec()ed programs (BPXBATCH or UNIX shell programs)

Setting Run-Time Options

- Options Merge (priority)
  - Program Invocation Overrides
  - **DD:CEEOPTS Overrides**
  - Programmer Overrides
  - Application Level Overrides
  - Region Level Overrides (where applicable)
  - **System Defaults (CEEPRMxx and SETCEE)**
  - Installation Defaults

Note: Some options can only be specified as Installation Defaults, e.g. CBLOPTS(ON)

Setting Run-Time Options

- Installation Default options must be set based on needs of the applications
  - Language used (PL/I, COBOL, C/C++, FORTRAN) will drive some option settings
    - May require some compromise
  - Subsystem application runs under (CICS TS, IMS, batch, UNIX System Services, etc.) will drive some option settings
  - Specific applications can then override options as needed

Key Run-Time Options

- Tuning
- Diagnostics

Key Run-Time Options Tuning

- ALL31
- ANYHEAP
- BELOWHEAP
- HEAP
- HEAPPOOLS
- STACK
- RPTSTG

Key Run-Time Options

- ALL31(option)
  - ON For AMODE 31 programs
  - OFF For AMODE 24 programs
    (is determined dynamically under CICS TS)
    (also determined non-CICS with z/OS V1.2)

Key Run-Time Options

- ANYHEAP(initial, increment, location, disp)
  - initial — Minimum size of initial heap segment
  - increment — Minimum size of additional segments
  - location — BELOW (<16MB), ANYWHERE
  - disp — KEEP, FREE (action when empty)
  - Notes:
    - Used internally by Language Environment

Key Run-Time Options

- BELOWHEAP(initial, increment, disp)
  - initial Minimum size of initial heap segment
  - increment Minimum size of additional segments
  - disp KEEP, FREE (action when empty)
  - Notes:
    - Used internally by Language Environment

Key Run-Time Options

- HEAP(initial, increment, location, disp, init24, incr24)
  - initial — Minimum size of initial heap segment
  - increment — Minimum size of additional segments
  - location — BELOW, ANYWHERE
  - disp — KEEP, FREE (action when empty)

  - Notes:
    - COBOL working storage (for RENT programs)
    - Dynamic storage (C malloc, C++ new, PL/I ALLOCATE)

Key Run-Time Options

- HEAPPOOLS(ON|OFF, size1, prcnt1, …, size12, prcnt12)
  - size*N* Up to 12 cell pool sizes to be utilized
  - prcnt*N* Percentage of HEAP(initial) to be used for cell pool
  - Notes:
    - C/C++ and Enterprise PL/I only
    - Use RPTSTG(ON) to tune settings for each application

Key Run-Time Options

- STACK(init, incr, location, disp, dsinit, dsincr)
  - init — Actual size of initial stack segment
  - incr — Minimum size of additional segments
  - location — BELOW, ANYWHERE
  - disp — KEEP, FREE (action when empty)
  - dsinit — XPLINK initial stack
  - dsincr — XPLINK increment stack

Key Run-Time Options

- STACK(init, incr, location, disp, dsinit, dsincr) *(continued)*
  - Notes:
    - Dynamic Save Area
    - C/C++ and PL/I local variables
    - Must use STACK(,,BELOW) when running ALL31(OFF)

Key Run-Time Options

- RPTSTG(option)
  - OFF Storage report not requested
  - ON Generates a report of stack/heap usage
    - including recommended settings
  - Caution:
    - Use only for application tuning. Do not make RPTSTG(ON) system wide default due to significant performance impact.

Consider CICS TS dynamic storage tuning as an alternative.

Key Run-Time Options

```
Storage Report for Enclave main 01/23/01 1:18:38 PM
Language Environment V02 R10.00

    STACK statistics:
      Initial size:                                  131072
      Increment size:                                131072
      Maximum used by all concurrent threads:         29024
      Largest used by any thread:                     29024
      Number of segments allocated:                       1
      Number of segments freed:                           0
    THREADSTACK statistics:
      Initial size:                                       0
      Increment size:                                     0
      Maximum used by all concurrent threads:             0
      Largest used by any thread:                         0
      Number of segments allocated:                       0
      Number of segments freed:                           0
         .
         .
```

Key Run-Time Options

```
HEAP statistics:
  Initial size:                                      32768
  Increment size:                                    32768
  Total heap storage used (sugg. initial size):     417624
  Successful Get Heap requests:                         91
  Successful Free Heap requests:                        52
  Number of segments allocated:                          4
  Number of segments freed:                              0
     .
     .
BELOWHEAP statistics:
  Initial size:                                      10240
  Increment size:                                     2048
  Total heap storage used (sugg. initial size):        168
  Successful Get Heap requests:                          1
  Successful Free Heap requests:                         0
  Number of segments allocated:                          1
  Number of segments freed:                              0
```

Key Run-Time Options

```
HeapPools Summary:
  Cell   Extent    Cells Per   Extents     Maximum      Cells In
  Size   Percent   Extent      Allocated   Cells Used   Use
  ------------------------------------------------------------
     8     10          204            0            0           0
    32     10           81            0            0           0
   128     10           24            1            1           1
   256     10           12            0            0           0
  1024     10            3            1            2           2
  2048     10            1            0            0           0
  ------------------------------------------------------------
  Suggested Percentages for current Cell Sizes:
    HEAPP(ON,8,1,32,1,128,1,256,1,1024,7,2048,1)
  Suggested Cell Sizes:
    HEAPP(ON,48,,280,,584,,2048,,0)
```

Key Run-Time Options Diagnostics

- ABTERMENC
- ERRCOUNT
- HEAPCHK
- STORAGE
- TERMTHDACT
- TRAP
- RPTOPTS
- USRHDLR

Key Run-Time Options

- ABTERMENC(option)
  - ABEND Step will be ABENDed (job terminates)
  - RETCODE Step ends with return code (job continues)

Key Run-Time Options

- ERRCOUNT(threshhold)
  - 0 Conditions are not counted
  - >0 The number of conditions allowed before Language Environment terminates the process
  - Notes:
    - ERRCOUNT(0) required by PL/I for ON-unit processing and by C/C++ for signal processing

Key Run-Time Options

- HEAPCHK(ON|OFF, frequency, delay, level, call-depth)
  - OFF — Normal processing
  - ON — Checks HEAP structures on get/free
  - frequency — How often the HEAP is checked
  - delay — Number of get/free before starting
  - level — Number of calls to be displayed in Heap Storage Diagnostic Report
  - call-depth — Number of calls to be displayed for HEAPPOOLS Serviceability

Key Run-Time Options

- HEAPCHK(ON|OFF, frequency, delay, level, call-depth) *(continued)*
  - Caution:
    - Use only for application tuning/diagnostics. Do not make HEAPCHK(ON) system wide default due to serious performance impact.
  - Notes:
    - To generate only Heap Storage Diagnostic Report use, e.g.
      - HEAPCHK(ON,0,0,10,0)
    - To activate only HEAPPOOLS Serviceability use, e.g.
      - HEAPCHK(ON,0,0,0,5)

Key Run-Time Options

```
Heap Storage Diagnostics
  Stg Addr  ID        Length     Entry          E Addr    E Offset    Load Mod

  1FB3C038  00000000  00000C20   CEEV#GTS       0CE7C610  +00000000   CEEPLPKA
                                 CEEVGQP        0CE855E0  +000001FC   CEEPLPKA
                                 setlocale      0CC7BFA0  +000000FC   CEEEV003
                                 tzset          0CBD34E8  +0000059A   CEEEV003
                                 _cinit         0CAC3588  +00002E48   CEEEV003
                                 CEEZINV        0CEB4870  +00000C58   CEEPLPKA
  1FB3CC58  00000000  00000828   CEEV#GH        0CE84690  +00000000   CEEPLPKA
                                 realloc_name_buffer
                                                0CC7BAD8  +00000070   CEEEV003
                                 setlocale      0CC7BFA0  +0000013C   CEEEV003
                                 tzset          0CBD34E8  +0000059A   CEEEV003
                                 _cinit         0CAC3588  +00002E48   CEEEV003
                                 CEEZINV        0CEB4870  +00000C58   CEEPLPKA
      .
      .
      .
```

Key Run-Time Options

- STORAGE(getheap, freeheap, stack, reserve)
  - getheap — One byte value used to initialize every heap allocation
  - freeheap — One byte value used to initialize every heap free
  - stack — One byte value used to initialize every stack allocation
  - reserve — Amount of space to reserve for out of storage condition processing

Key Run-Time Options

- STORAGE(getheap, freeheap, stack, reserve) *(continued)*
  - Notes:
    - STORAGE(AA,EE,,) useful for debugging
      - When HEAPCHK(ON), free elements are checked to ensure they contain the freeheap value
    - STORAGE(00,,,) is equivalent to COBOL WSCLEAR
    - STORAGE(,,00,) is very expensive
      - Especially for C/C++
      - Solution: STORAGE(,,CLEAR)
        - Sets to binary zeros the unused portion of the initial stack segment just prior to the “main” getting control
        - Available via APAR PK02614 (z/OS V1.4 and later)

Key Run-Time Options

- TERMTHDACT(option)
  - QUIET — Messages off, no dump
  - MSG — Messages only, no dump
  - TRACE — CEEDUMP with traceback only
  - DUMP — CEEDUMP
  - UADUMP — CEEDUMP, optional system dump
  - UAONLY — System dump only, no CEEDUMP
  - UATRACE — System dump and traceback
  - UAIMM — System dump of original error

Key Run-Time Options

- TERMTHDACT(option) *(continued)*
  - Notes:
    - UAIMM requires TRAP(ON,NOSPIE)
      - Takes dump in Language Environment ESTAE via SETRP prior to Language Environment condition handling for program checks only
    - SYSMDUMP DD card required for system dump

Key Run-Time Options

- TRAP(option)
  - ON,SPIE — Condition handling enabled
  - ON,NOSPIE — Allows user applications to have their own SPIE routine, Language Environment condition handling will take place via the ESTAE
  - OFF — Condition handling disabled, some functionality not available **(AVOID)**
- Notes:
  - TRAP(ON,SPIE) highly recommended for normal processing

Key Run-Time Options

- RPTOPTS(option)
  - OFF — Options report not requested
  - ON — Generate a report of all current options (upon successful termination)
  - Notes:
    - Automatically included in CEEDUMP

Key Run-Time Options

```
Options Report for Enclave main 07/13/05 12:18:07 PM
Language Environment V01 R07.00

LAST WHERE SET                    OPTION
------------------------------------------------------------------------
Installation default                ABPERC(NONE)
Installation default                ABTERMENC(ABEND)
Installation default              NOAIXBLD
Invocation command                  ALL31(OFF)
Installation default                ANYHEAP(16384,8192,ANYWHERE,FREE)
Installation default              NOAUTOTASK
Programmer default                  BELOWHEAP(10240,2048,KEEP)
Installation default                CBLOPTS(ON)
Installation default                CBLPSHPOP(ON)
Installation default                CBLQDA(OFF)
Installation default                CHECK(ON)
PARMLIB(CEEPRMxx)                   COUNTRY(KY)
Installation default              NODEBUG
DD:CEEOPTS                          DEPTHCONDLMT(15)
Installation default                ENVAR("")
SETCEE command                      ERRCOUNT(0)
```

Key Run-Time Options

- USRHDLR(CEEWUCHA)
  - User condition handler to provide traditional CICS TS ABEND behavior
    - Convert unhandled COBOL detected software errors to U1xxx ABEND issued by COBOL II
      - Allow EXEC CICS HANDLE ABEND LABEL statements to get control
    - Convert PL/I software-raised conditions back to APLS abends

Key Run-Time Options

- USRHDLR(CEEWUCHA) *(continued)*
  - Prevents message IGZ0014W warning of backlevel IGZETUN or IGZEOPT
    - Use CEEUOPT instead
  - Installed as usermod via CEEWWCHA
    - Found in SCEESAMP
  - CEEWUCHA must be defined in PPT and must be available at run-time (e.g., STEPLIB, LPA, etc.)
  - Not recommended for batch

Additional Information

- All Language Environment documentation available:
  - z/OS CD collection
  - Language Environment website:
    - http://www.ibm.com/servers/eserver/zseries/zos/le/
- Language Environment Programming Reference
- Language Environment Programming Guide
- Language Environment Customization
- Language Environment Run-Time Application Migration Guide
  - Run-time Option Summary and Recommendations chapter

Appendix: AMODE 64 Run-Time Options

Existing run-time options supported in AMODE 64:

ARGPARSE/NOARGPARSE
ENVAR
EXECOPTS/NOEXECOPS
FILETAG
HEAPCHK
INFOMSGFILTER
NATLANG
POSIX
PROFILE
REDIR/NOREDIR
RPTOPTS
RPTSTG
STORAGE *
TERMTHDACT *
TEST/NOTEST
TRACE *
TRAP

- All other existing run-time options are not supported
- * indicates some suboptions have changed or are not supported

AMODE 64 Run-Time Options *(continued)*

- HEAPPOOLS64(ON|OFF,cell1 size, cell1 count,...,cell12 size, cell12 count)

- ON|OFF — Are heappools on?
- cellX size — Size of cells with this pool (8 to 64K)
- cellX count — Number of cells in this pool (min 4)

NOTE: Different from AMODE 24/31 (HEAPPOOLS) – was percentage (not changing in AMODE 24/31)

AMODE 64 Run-Time Options *(continued)*

- HEAP64(init64,inc64,disp64,init31, inc31, disp31,
  init24, inc24, disp24)
  - Controls user heap storage
- IOHEAP64(init64,inc64,disp64,init31, inc31, disp31,
  init24, inc24, disp24)
  - Controls I/O storage for the run-time
- LIBHEAP64(init64,inc64,disp64,init31, inc31, disp31,
  init24, inc24, disp24)
  - Controls heap storage usage for the run-time (non I/O)

AMODE 64 Run-Time Options *(continued)*

- AMODE 64 Heap Run-Time Options Parameters

- init64 Initial size of above the bar storage (in MB)
- inc64 Increment size of above the bar storage (in MB)
- disp64 KEEP or FREE (how to treat 64 increments)
- init31 Initial size of above the line storage (in bytes)
- inc31 Increment size of above the line storage (bytes)
- disp31 KEEP or FREE (how to treat 31 increments)
- init24 Initial size of below the line storage (in bytes)
- inc24 Increment size of below the line storage (bytes)
- disp24 KEEP or FREE (how to treat 24 increments)

AMODE 64 Run-Time Options *(continued)*

- STACK64(initial, increment, maximum) - Controls the allocation of user stack
- THREADSTACK64(initial, increment, maximum) - Controls the allocation of user stack

- initial — Size of initial stack (in MB)
- increment — Size of increments of stack (in MB)
- maximum — Maximum stack size (in MB)

- NOTES:
- Stack is always above the bar, one contiguous segment, downward growing (XPLINK)
- Initially we reserve “maximum” space
- Only use initial size and increase in increments until maximum is reached.
- Only what is actually used counts towards MEMLIMIT