What's New in z/OS Language Environment?

Corey Bryant
IBM Poughkeepsie
bryntcor@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.

IEEE is a trademark in the United States and other countries of the Institute of Electrical and Electronics Engineers, Inc.

POSIX® is a registered Trademark of The IEEE.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, or service names may be trademarks or service marks of others.

* 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.

Agenda

- New Function APARs
- What's New in z/OS V1.9?
- What's New in z/OS V1.8?

- Additional information that is not on the agenda:
  - Appendix
    - What's New in z/OS V1.7?
    - Additional Reference Material
    - Sources of Additional Information

NEW FUNCTION APARS

- STORAGE(,,CLEAR)
- XPLINK Performance Improvements
- Enterprise PL/I Under CICS Performance Improvement
- Toleration APAR for CEEUOPT and CELQUOPT
- _EDC_PTHREAD_YIELD
- Locale Updates
- Daylight Savings Time

STORAGE(,,CLEAR)

- **APAR PK02614** (PTFs available 2005-11-23)
  - PL/I migration issue; OS PL/I run-time used to clear the initial stack segment before the “main” received control; Language Environment does not have similar capability; Using STORAGE(,,00) is too expensive.
  - Keyword CLEAR added for 3rd suboption on the STORAGE run-time option
  - Minimum software requirement:
    - z/OS V1.4 Language Environment

XPLINK Performance Improvements

- **APAR PK24077** (PTFs available 2006-08-03)
  - Eliminates a downstack (XPLINK) to upstack (non-XPLINK) transition in the following situations:
    - HEAPPOOLS are being used and a storage element greater than the largest cell size is allocated or freed.
    - HEAPPOOLS are being used and a cell pool is extended.
    - long long division is performed.
  - Minimum software requirement:
    - z/OS V1.8 Language Environment

Enterprise PL/I CICS Performance Improvement

- **APAR PK24874** (PTFs available 2006-11-08)
  - Provides performance improvement for users of Enterprise PL/I in a CICS environment when initializing many run units.
  - Minimum software requirement:
    - z/OS V1.6 Language Environment

Toleration APAR for CEEUOPT and CELQUOPT

- **APAR PK29028** (PTFs available 2007-03-06)
  - This APAR provides CEEUOPT and CELQUOPT downward compatibility toleration.
    - Provides informational run-time messages when an application is compiled and link-edited on z/OS V1.9, and CEEUOPT or CELQUOPT contain run-time options that are not supported when executed on a lower release.
    - This also fixes a potential ABENDS0C4 REASON=11that may result in CELQLIB within CSECT CEEBOPTM when there is more than one error found during the merge of system-wide run-time options or programmer default run-time options for an AMODE 64 application.
  - Minimum software requirement:
    - z/OS V1.6 Language Environment

_EDC_PTHREAD_YIELD

- **APAR PK17514** (PTFs available 2006-07-28)
  - A new environment variable, _EDC_PTHREAD_YIELD, has been added which can be used to control the speed at which pthread_yield() and sched_yield() give up control of a processor so that another thread may have the opportunity to run.
  - Minimum software requirement:
    - z/OS V1.6 Language Environment

- **Possible values for _EDC_PTHREAD_YIELD:**

Value	Meaning
0	Control of the processor is released immediately.
-1	Use an internal timing algorithm to determine if the processor should be released. Default for z/OS V1.6 and V1.7 APAR.
-2	Take the machine speed into account when determining if the processor should be released. Default for z/OS V1.8.
other negative values	Invalid. The default will be used.

Locale Updates

- **APAR PK34038** (PTFs available 2006-12-07)
  - Enhancements are made to the **Slovenian locale** data to make the EURO the default currency symbol.
    - Slovenia will introduce the EURO as legal tender as of January 1, 2007.
  - Minimum software requirement:
    - z/OS V1.4 Language Environment

- **APAR PK38183** (PTFs available 2007-06-27)
  - Euro and preeuro versions of the **locales for Bulgaria and Romania** are provided by this APAR. The default locales for these countries have not changed.
  - Minimum software requirement:
    - z/OS V1.6 Language Environment

Locale Updates

- **APAR PK31540** (PTFs available 2007-07-27)
  - **Yen Sign In IBM-943 Gets Coverted To Backslash**
  - A new Japanese converter has been added, IBM-62383. IBM-62383 adds a converter which will provide a mapping to the Yen symbol. The Yen symbol is not part of the existing IBM-943 converter.
  - A new Japanese converter has been added, IBM-54191. IBM-54191 adds a converter which will provide a mapping to the Yen symbol, as well as modifications to the mapping of the not sign and the circumflex.
  - Minimum software requirement:
    - z/OS V1.6 Language Environment

Daylight Savings Time

- **APAR PK24076** (PTFs available 2006-11-08)
  - Updates the z/OS XL C/C++ Run-Time Library to reflect the new Daylight Savings Time rules.
    - All users of the TZ and _TZ environment variables and all users of the LC_TOD locale category are affected.
    - Language Environment callable services, COBOL and PL/I are not affected.
  - Minimum software requirement:
    - OS/390 V2.10 Language Environment

What's New in z/OS V1.9?

- CEEDUMP and IPCS VERBEXIT LEDATA Enhancements
- Preinitialization (PIPI) Tracing
- 64-Bit Preinitialization (PIPI) Service Routines (LOAD/DELETE)
- XPLINK Support for IMS
- CLER Run-time Option Support
- Callable Services
- DLL Diagnostics
- 2005 GWP Compliance and Turkish Lira Currency Update
- z/OS UNIX CEEBLDTX Support
- NATLANG For All Output
- fdlibm Replacements
- errno2 Enhancements
- C/C++ SUSv3
- Multicast Source Filter APIs
- Heap Pools Performance Improvements
- IEEE Decimal Floating Point
- AMODE 64 CEETBCK and CEEHGOTO
- Metal C Run-time Library
- Daylight Savings Time

CEEDUMP and IPCS VERBX LEDATA Enhancements

- **Problem Statement**
  - Need to improve the serviceability of Language Environment applications.
- **Solution**
  - Improved serviceability of Language Environment applications is provided via:
    - New run-time option to control characteristics of CEEDUMP dataset.
    - Enhanced traceback section of Language Environment dump.
    - Enhanced condition information section of Language Environment dump.
    - Job information added to Language Environment dump page header.
    - New messages identify start and end of Language Environment dump.
    - Language Environment dump processing suppressed when CEEDUMP ddname is defined as dummy.

CEEDUMP and IPCS VERBX LEDATA Enhancements

- A new run-time option will be provided with suboptions for controlling the characteristics of the CEEDUMP data set:

CEEDUMP(pagelen,
SYSOUT=*|SYSOUT=class|SYSOUT=(class,,form),
FREE=END|FREE=CLOSE,
SPIN=UNALLOC|SPIN=NO)

  - **pagelen** - determines number of lines per page dump report
  - **SYSOUT** - controls the SYSOUT class and form attributes for a dynamically allocated CEEDUMP dataset
  - **FREE** - determines when a dynamically allocated CEEDUMP data set is unallocated
  - **SPIN** - determines when a dynamically allocated CEEDUMP data set is available for processing

CEEDUMP and IPCS VERBX LEDATA Enhancements

- **Sample traceback section from a z/OS V1.8 Language Environment dump:**

```
Traceback:
  DSA Addr  Program Unit  PU Addr   PU Offset  Entry        E Addr    E  Offset   Statement  Load Mod  Service  Status
  20A43E80  //'POSIX.CRTL.C(CELDLL)'
                          20DC83B0  +0000012E  dump_n_perc  20DC83B0  +0000012E          34  CELDLL    1.3.B    Call
  20A43DC8                0CD0F628  -00000106  CEEPGTFN      0CD0F4C8  +0000005A              CEEPLPKA           Call
  20A40CA8  CEEHDSP       0CC55518  +000024D0  CEEHDSP       0CC55518  +000024D0              CEEPLPKA  UK20433  Call
  20A40208  //'POSIX.CRTL.C(CELSAMP)'
                          20A26478  +000009BA  main          20A26478  +000009BA         150  *PATHNAM  1.1.D    Exception
  20A400F0                0E284606  +000000C2  EDCZMINV      0E284606  +000000C2              CEEEV003           Call
  20A40030  CEEBBEXT      0CC25418  +000001B6  CEEBBEXT      0CC25418  +000001B6              CEEPLPKA  HLE7730  Call
```

- **The next slide contains a sample traceback section from a z/OS V1.9 Language Environment dump produced by the same program.**

CEEDUMP and IPCS VERBX LEDATA Enhancements

```
Traceback:
  DSA    Entry       E  Offset  Statement   Load Mod            Program Unit                Service  Status
  1      dump_n_perc +0000012E  34          CELDLL              CELDLL                      1.3.B    Call
  2      CEEPGTFN    +0000005A              CEEPLPKA                                        Call
  3      CEEHDSP     +000024BC              CEEPLPKA            CEEHDSP                     D1908    Call
  4      main        +000009BA  150         celsamp.exe         CELSAMP                     1.1.D    Exception
  5      EDCZMINV    +000000C2              CEEEV003                                        Call
  6      CEEBBEXT    +000001B6              CEEPLPKA            CEEBBEXT                    D1908    Call

  DSA    DSA Addr    E  Addr     PU Addr     PU Offset  Comp Date  Compile Attributes
  1      210F6E80    2147B3B0    2147B3B0    +0000012E  20070105   C/C++      POSIX EBCDIC  HFP
  2      210F6DC8    20BA7EB8    20BA8018    -00000106  20061215   LIBRARY    POSIX
  3      210F3CA8    20AEC068    20AEC068    +000024BC  20061215   CEL        POSIX
  4      210F3208    20A26478    20A26478    +000009BA  20070105   C/C++      POSIX EBCDIC  HFP
  5      210F30F0    20F930EE    20F930EE    +000000C2  20061215   LIBRARY    POSIX
  6      210F3030    20ABADB8    20ABADB8    +000001B6  20061215   CEL        POSIX

  Fully Qualified Names
  DSA    Entry       Program Unit                                  Load Module
  1      dump_n_perc //'POSIX.CRTL.C(CELDLL)'                      CELDLL
  4      main        //'POSIX.CRTL.C(CELSAMP)'                     ./celsamp.exe
```

CEEDUMP and IPCS VERBX LEDATA Enhancements

- **Enhanced condition information section of LE dump:**

```
Condition Information for Active Routines
    Condition Information for  (DSA address 20FCB2B0)
      CIB Address: 20FCBC70
      Current Condition:
        CEE0198S The termination of a thread was signaled due to an unhandled condition.
      Original Condition:
        CEE3201S The system detected an operation exception (System Completion Code=0C1).
      Location:
        Program Unit:  Entry: funca Statement:  Offset: -20900978
        Possible Bad Branch:  Statement:   Offset: +0000005A
      Machine State:
        ILC..... 0002    Interruption Code..... 0001
        PSW..... 078D1400 80000002
        GPR0..... 20FCB350  GPR1..... 20FCB2A0  GPR2..... 20FCB2A0  GPR3..... 209009B2
        GPR4..... A09A0BBC  GPR5..... 20912648  GPR6..... 20900AA4  GPR7..... 20900098
        GPR8..... 00000030  GPR9..... 80000000  GPR10.... A0E699E2  GPR11.... A09A0AD8
        GPR12.... 209139B0  GPR13.... 20FCB2B0  GPR14.... A09009D4  GPR15.... 00000000

    Storage dump near condition, beginning at location: 00000000
      +000000 00000000  Inaccessible storage.
    GPREG STORAGE:
       Storage around GPR0 (20FCB350)
          -0020 20FCB330  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  |................................|
          +0000 20FCB350  0808CEE1 20FCB2B0 20FCE470 A09D6B3A  A09EFFD8 20FCB350 20FCB7A8 20912648  |.........U...,....Q...&...y.j..|
            .
            .
```

CEEDUMP and IPCS VERBX LEDATA Enhancements

- **New page headers in the Language Environment dump provide additional information about the job:**
  - **Batch with POSIX(ON):**

```
CEE3DMP V1 R9.0: Sample dump produced by calling CEE3DMP                    01/05/07 11:22:37 PM
     Page:    1
ASID: 003F   Job ID: JOB14342   Job name: CELSAMP    Step name: STEP1      PID: 33620309   Parent PID: 1
     User name: HEALY
```

  - **Batch with POSIX(OFF):**

```
CEE3DMP V1 R9.0: Plidump called from error On-unit                          01/31/07 3:59:39 PM
     Page:    1
ASID: 010E   Job ID: J0009410   Job name: LEDGSMP1   Step name: GO         UserID: BARBARA
```

  - **UNIX shell**

```
CEE3DMP V1 R9.0: Sample dump produced by calling CEE3DMP                    12/22/06 2:25:00 PM
     Page:    1
ASID: 012B   PID: 83952502   Parent PID: 67174622   User name: ALFCAR
```

  - **CICS**

```
CEE3DMP V1 R9.0: Condition processing resulted in the unhandled condition.  01/16/07 6:07:22 PM
     Page:    1
Task Number: 0860   Transaction ID: DIVZ
```

CEEDUMP and IPCS VERBX LEDATA Enhancements

- New messages in the Language Environment dump designate the start and the end of the dump:
  - **CEE3845I CEEDUMP processing started.**
  - **CEE3846I CEEDUMP processing completed.**

- Language Environment dump processing is suppressed when CEEDUMP ddname is defined as dummy:
  - CEE3DMP callable service can return new feedback code CEE317
    - **CEE3111I CEEDUMP was defined as a dummy data set. No Language Environment dump processing was performed.**

Preinitialization (PIPI) Tracing

- **Problem Statement**
  - Language Environment PreInit exploiters have difficulty understanding:
    - what sequence of events occurred
    - what services were used (storage management, etc.)
    - what error indications resulted when the interface was used
- **Solution**
  - Support has been added to improve diagnostics for PreInit applications.
    - Support has been added for PreInit trace.
    - Support has been added for IPCS PreInit trace and control blocks.

Preinitialization (PIPI) Tracing

- **A new keyword (PTBL) has been added to the LEDATA IPCS Verbexit.**

  PTBL (value) - Requests that PreInit information be formatted based on one of the following values:

  - **"CURRENT"**

    If current is specified, the PreInit table associated with the current or specified TCB is displayed.

  - ***address***

    If an address is specified, the PreInit table at that address is displayed.

  - **" * "**

    All active and dormant PreInit tables within the current address space are displayed; note that this option is time-consuming.

  - **"ACTIVE"**

    All TCB's in the address space will have their PreInit tables displayed.

Preinitialization (PIPI) Tracing

- **PreInit Tracing characteristics**
  - Tracing is always active.
    - Tracing begins when the PreInit environment is initialized and ends when the environment is terminated.
  - Trace is kept in an in-storage trace table.
    - Table is a fixed size.
    - Table wraps when the end has been reached.
  - Formatted trace entries are displayed from oldest to newest.
- In addition, the active PreInit routine is now identified in the traceback (previously, CEEPIPI/CELQPIPI was always shown).

Preinitialization (PIPI) Tracing

```
=== > VERBEXIT LEDATA PTBL(CURRENT)'

PreInitialization Programming Interface Trace Data
 CEEPIPI Environment Table Entry and Trace Entry :
  Active CEEPIPI Environment (  Address 25805CB0 )
  Eyecatcher  : CEEXIPTB
  TCB address : 008D1B08

 CEEPIPI Environment :
  Non-XPLINK Environment
  Environment Type : MAIN
  Sequence of Calls not active
  Exits not established
  Signal Interrupt Routines not registered
  Service Routines are not active

  CEEPIPI Environment Enclave Initialized
  Number of CEEPIPI Table Entries = 2
```

Preinitialization (PIPI) Tracing

```
CEEPIPI Table Entry Information :
 CEEPIPI Table Index 0  ( Entry 1   )
  Routine Name  = HLLCRTN
  Routine Type  = C/C++
  Routine Entry Point  = A5810B38
  Routine Function Pointer  = A5810CC0
  Routine Entry is Non-XPLINK
  Routine was loaded by Language Environment
  Routine Address was resolved
  Routine Function Descriptor was valid
  Routine Return Code    = 0
  Routine Reason Code    = 0
```

Preinitialization (PIPI) Tracing

```
Entry of routine in CEEPIPI Table for Index 0 ( 25805DB8 )

 +000000 25805DB8  A5810CC0 25811B30 80000000 00000000
                   00000000 00000000 00000000 00000000
                   |va...a.........................|
 +000020 25805DD8  00000000 00000000 00000000 A5810B38
                   00000003 258117C8 00000003 25810B38
                   |............va.......a.H.....a..|
 +000040 25805DF8  A5810B38 000014C8 C8D3D3C3 D9E3D540
                   00000000 00000000 00000000 00000000
                   |va.....HHLLCRTN ................|

CEEPIPI Table Index 1  ( Entry 2   )  not in use.
```

Preinitialization (PIPI) Tracing

```
CEEPIPI Trace Table Entries :
 Call Type =  INIT_MAIN
  PIPI Driver Address  = A5800A82
  Load Service Return Code   = 0
  Load Service Reason Code   = 0
  Most Recent Return Code    = 0
  Most Recent Reason Code    = 0
  An ABEND will be issued if storage can not be obtained
  PreInit Environment will not allow EXEC CICS commands
  Service RC = 0 :A new environment was initialized
```

Preinitialization (PIPI) Tracing

```
Call Type =  ADD_ENTRY
 Routine Table Index        = 1
 Routine Name  = HLLCOBOL
 Routine Address = A5812E20
 Load Service Return Code   = 0
 Load Service Reason Code   = 3
 Service RC = 0 :The routine was added to the PreInit table.


Call Type =  CALL_MAIN
 Routine Table Index     = 1
 Enclave Return Code     = 0
 Enclave Reason Code     = 0
 Routine Feedback Code   = 0000000000000000
 Service RC = 0 :The environment was activated and the
    routine called.
```

Preinitialization (PIPI) Tracing

```
Call Type =  DELETE_ENTRY
    Routine Table Index     = 1
    Routine Name  = HLLCOBOL
    Routine Address = A5812E20
    Service RC = 0 :The routine was deleted from the
       PreInit table.

   Call Type =  CALL_MAIN
    Routine Table Index     = 0
    Enclave Return Code     = 0
    Enclave Reason Code     = 0
    Routine Feedback Code   = 0000000000000000
    Service RC = 0 :The environment was activated and
       the routine called.
```

64-Bit Preinitialization (PIPI) Service Routines

- **Problem Statement**
  - Some AMODE 64 preinitialization (PreInit) applications need to replace the IBM-supplied LOAD and DELETE functions with customer-written routines (as is possible with AMODE 31).
- **Solution**
  - Support is provided to allow IBM-supplied LOAD and DELETE routines to be replaced (as is the case for AMODE 31).
  - CELQPIPI(init_xxx, ,service_rtns,) – previously-reserved 3rd parameter for CELQPIPI can now optionally point to a service routine vector (similar to AMODE 31 CEEPIPI calls).
  - AMODE 64 PreInit only supports replacing the LOAD and DELETE routines.
    - Whereas AMODE 31 PreInit applications can replace the LOAD, DELETE, GETSTOR, FREESTOR, EXCEPTRTN, and MSGRTN routines.

XPLINK Support for IMS

- **Problem Statement**
  - Support for running AMODE 31 XPLINK applications in an IMS environment with Library Routine Retention (LRR) is required.
- **Solution**
  - Language Environment will support AMODE 31 XPLINK applications running under an IMS environment, both with and without LRR support.
    - See sample XPLINK LRR program code in SCEESAMP(CEELRRXP). This program is the XPLINK version of existing sample program CEELRRIN.
    - This support is invoked via a call to the CEELRR macro from a non-Language Environment-conforming Assembler program. For example:
      - `CEELRR ACTION=INIT,XPLINK=YES`

CLER Run-time Option Support

- **Problem Statement**
  - A request was made for the CHECK and INFOMSGFILTER options, and that RPTOPT and RPTSTG options require confirmation before being turned on.
- **Solution**
  - The CLER CICS transaction has been updated to allow additional run-time options to be changed for a CICS region.
  - The CHECK and INFOMSGFILTER run-time options have been added to the CLER CICS panel.
  - The list of options on the CLER panel is now displayed in alphabetical order.
  - CLER users who use the ON suboption of CBLPSHPOP, RPTSTG, and RPTOPTS, or who use the OFF suboption of ALL31, will now be prompted to confirm these settings.

CLER Run-time Option Support

- **The following example shows the new CLER panel with confirmation message after setting RPTSTG(ON):**

```
CLER                                                          CICS CICS530

               Language Environment Region Level Run-time Options

 Type in your Choices.

 Runtime option     Choice      Possible choices.

 ALL31          ==> ON          ON, OFF
 CBLPSHPOP      ==> ON          ON, OFF
 CHECK          ==> OFF         ON. OFF
 INFOMSGFILTER ==> OFF          ON, OFF - ON equates to INFOMSGFILTER(ON,CICS)
 RPTOPTS        ==> OFF         ON, OFF
 RPTSTG         ==> ON          ON, OFF
 TERMTHDACT     ==> TRACE       QUIET,MSG,TRACE,DUMP,UAONLY,UATRACE,UADUMP,UAIMM
 TRAP           ==> ON          ON, OFF

 When finished, press ENTER.

             W - Options that may adversely affect performance have changed.
             W - Pressing Enter will apply changes.

       PF1=Help    3=Quit    5=Current Settings   9=Error List
```

Callable Services

- **Problem Statement**
  - The VSE operating system provides a callable service CEE5DLY that enables the Language Environment-conforming program to suspend execution for a specified number of seconds.
    - Language Environment has no equivalent functionality on z/OS, causing portability concerns for VSE applications that use CEE5DLY.
      - Current applications use ILBOWAT0 routine, requing an AMODE24 parameter below 16M line, and causing virtual constraint problems.
      - Sites have had to develop and maintain in-house assembler routines.
- **Solution**
  - New callable services are provided for putting the active enclave to sleep for:
    - Seconds (**CEE3DLY**), Milliseconds (**CEEDLYM**)

Callable Services

- **Problem Statement**
  - There was no callable service to return international currency symbol (as a string).
- **Solution**
  - **CEE3MC2** returns the default currency symbol and international currency symbol for the country you specify with *country_code*.
    - For example:
      - The default currency symbol for the US is: $
      - The international currency symbol for the US is: USD

DLL Diagnostics

- **Problem Statement**
  - DLL failures can be difficult to diagnose.
- **Solution**
  - The following RAS improvements are now available for DLL applications:
    - A new C/C++ environment variable (**_EDC_DLL_DIAG**) is being provided to allow users to configure the handling of C/C++ DLL errors.
    - A new Language Environment control block chain (**CEEDLLF**) is being provided that will contain DLL failure diagnostics.

DLL Diagnostics

- **_EDC_DLL_DIAG Value Descriptions**
  - **MSG -** Issue DLL error messages to the Language Environment message file.
  - **TRACE -** Same as MSG, but also calls the ctrace() function to produce a traceback for each error.
  - **SIGNAL -** Same as TRACE, but also signals a condition for each error's feedback code.
  - **QUIET -** Turn off all _EDC_DLL_DIAG error diagnostics. This is the default.

- **_EDC_DLL_DIAG also provides improved error messages:**
  - For example, _EDC_DLL_DIAG=MSG and dllload() is attempted, but the module is not found:
    - The following message is issued by _EDC_DLL_DIAG:
      - **CEE3501S The module *module-name* was not found.**
    - And the following message continues to be issued by perror():
      - **EDC5205S DLL module not found.**

DLL Diagnostics

- **CEEDLLF – DLL Failure Control Block**
  - Contains diagnostics for an implicit or explicit DLL failure. Diagnostics for up to 10 of the most recent DLL failures are available in circular chain of CEEDLLF control blocks.
    - These are formatted in VERBX LEDATA and CEEDUMPs.
  - Following are some of the more valuable CEEDLLF Fields:

CEEDLLF_SERVICE	DLL service that failed (load, query function, free, etc.)
CEEDLLF_REFERENCE_TYPE	DLL reference type (explicit or implicit)
CEEDLLF_LOAD_TYPE	Load type attempted (MVS, Unix File System, or both)
CEEDLLF_PREV	Pointer to previous CEEDLLF in chain
CEEDLLF_NEXT	Pointer to next CEEDLLF in chain
CEEDLLF_FBTOK	Feedback token for this failure
CEEDLLF_DLL_NAME	Pointer to DLL name
CEEDLLF_SYMBOL_NAME	Pointer to DLL function/variable name
CEEDLLF_DLL_NAME_LEN	Length of DLL name
CEEDLLF_SYMBOL_NAME_LEN	Length of DLL function/variable name
CEEDLLF_RETCODE1	Unix File System load return code
CEEDLLF_RSNCODE1	Unix File System load reason code
CEEDLLF_RETCODE2	MVS load return code
CEEDLLF_RSNCODE2	MVS load reason code

2005 GWP Compliance and Turkish Lira Currency Update

- **Problem Statement**
  - New locales are required based on annual updates for new Language-Territory combinations specified in GWP (G11N White Paper).
  - New Turkish Lira currency not supported by current Turkish locales.
- **Solution**
  - New GWP updates have been made. ASCII support has been added to a set of existing Euro and pre-Euro locales.
    - Note: SCEERUN2 is now much larger because of the new ASCII locales.
  - Turkish locales have been updated to use the new Lira currency.
  - See appendix for more details.

z/OS UNIX CEEBLDTX Support

- **Problem Statement**
  - There is no support for the CEEBLDTX utility in the z/OS UNIX environment.
- **Solution**
  - Language Environment wanted this support, and therefore implemented and externalized it.
  - The CEEBLDTX utility is now supported in the z/OS UNIX environment, along with additional new options.

NATLANG For All Output

- **Problem Statement**
  - The NATLANG run-time option setting is not honored for Language Environment dumps, storage reports, and options reports.
- **Solution**
  - Support has been added to eliminate lower case English characters from Language Environment dumps, storage reports, and options reports when NATLANG is UEN or JPN.
  - For Language Environment dumps, storage reports, and option reports, when NATLANG(UEN) or NATLANG(JPN) run-time option is used:
    - Constant data, like titles, headers and key words, will now appear in upper case English.
    - Variable data, like enclave name, entry point name and variable names will be displayed AS IS, unless the environment variable _CEE_UPPERCASE_DATA is set to YES.
  - This will allow Language Environment dumps, storage reports, and options reports to be understandable when printed on Japanese printers.

NATLANG For All Output

- **_CEE_UPPERCASE_DATA**
  - This is a new environment variable that can be used to determine whether or not variable data in the CEEDUMP, options report, and storage report is uppercased when the NATLANG run-time option is UEN (upper case English) or JPN (Japanese).
    - When this environment variable is set to YES, variable data (entry point names, program unit names, variable names, Trace Entry in EBCDIC data, hexadecimal/EBCDIC displays of storage) will be uppercased.
    - When this environment variable is not set or set to a value other than YES, variable data will not be uppercased. Variable data is never uppercased when NATLANG is ENU (mixed case English).

NATLANG For All Output

For comparison, z/OS V1.8 Storage report with NATLANG(UEN):

```
Options Report for Enclave main Thu Mar  1 19:43:08 2007
Language Environment V01 R08.00

LAST WHERE SET                 OPTION
------------------------------------------------------------------------------
Installation default             DYNDUMP(*USERID,NODYNAMIC,TDUMP)
Installation default             ENVAR("")
DD:CEEOPTS                       ENVAR("AaAaAa=BbBbBb","_CEE_UPPERCASE_DATA=YES")
Installation default             FILETAG(NOAUTOCVT,NOAUTOTAG)
Invocation command               HEAPCHK(ON,1,0,0,0)
Installation default             HEAPPOOLS64(OFF,8,4000,32,2000,128,700,256,350,1024,100,2048,50,3072,50,4096,50,8192,
                                   25,16384,10,32768,5,65536,5)
Installation default             HEAP64(1M,1M,KEEP,32768,32768,KEEP,4096,4096,FREE)
Installation default             INFOMSGFILTER(OFF,,,,)
Installation default             IOHEAP64(1M,1M,FREE,12288,8192,FREE,4096,4096,FREE)
Installation default             LIBHEAP64(1M,1M,FREE,16384,8192,FREE,8192,4096,FREE)
DD:CEEOPTS                       NATLANG(UEN)
.
.
```

NATLANG For All Output

z/OS V1.9 Storage report when NATLANG(UEN):

```
OPTIONS REPORT FOR ENCLAVE main THU FEB  8 15:19:31 2007
LANGUAGE ENVIRONMENT V01 R09.00

LAST WHERE SET                OPTION
------------------------------------------------------------------------------
INSTALLATION DEFAULT            CEEDUMP(60,SYSOUT=*,FREE=END,SPIN=UNALLOC)
INSTALLATION DEFAULT            DYNDUMP(*USERID,NODYNAMIC,TDUMP)
INSTALLATION DEFAULT            ENVAR("")
DD:CEEOPTS                      ENVAR("AaAaAa=BbBbBb")
INSTALLATION DEFAULT            FILETAG(NOAUTOCVT,NOAUTOTAG)
INSTALLATION DEFAULT            HEAPCHK(OFF,1,0,0,0)
INSTALLATION DEFAULT            HEAPPOOLS64(OFF,8,4000,32,2000,128,700,256,350,1024,100,2048,50,3072,50,4096,50,8192,
                                  25,16384,10,32768,5,65536,5)
INSTALLATION DEFAULT            HEAP64(1M,1M,KEEP,32768,32768,KEEP,4096,4096,FREE)
INSTALLATION DEFAULT            INFOMSGFILTER(OFF,,,,)
INSTALLATION DEFAULT            IOHEAP64(1M,1M,FREE,12288,8192,FREE,4096,4096,FREE)
INSTALLATION DEFAULT            LIBHEAP64(1M,1M,FREE,16384,8192,FREE,8192,4096,FREE)
DD:CEEOPTS                      NATLANG(UEN)
.
.
```

NATLANG For All Output

z/OS V1.9 Storage report with NATLANG(UEN) and _CEE_UPPERCASE_DATA=YES:

```
OPTIONS REPORT FOR ENCLAVE MAIN THU FEB  8 15:19:32 2007
LANGUAGE ENVIRONMENT V01 R09.00

LAST WHERE SET                OPTION
------------------------------------------------------------------------------
INSTALLATION DEFAULT            CEEDUMP(60,SYSOUT=*,FREE=END,SPIN=UNALLOC)
INSTALLATION DEFAULT            DYNDUMP(*USERID,NODYNAMIC,TDUMP)
INSTALLATION DEFAULT            ENVAR("")
DD:CEEOPTS                      ENVAR("AAAAAA=BBBBBB","_CEE_UPPERCASE_DATA=YES“)
INSTALLATION DEFAULT            FILETAG(NOAUTOCVT,NOAUTOTAG)
INSTALLATION DEFAULT            HEAPCHK(OFF,1,0,0,0)
INSTALLATION DEFAULT            HEAPPOOLS64(OFF,8,4000,32,2000,128,700,256,350,1024,100,2048,50,3072,50,4096,50,8192,
                                  25,16384,10,32768,5,65536,5)
INSTALLATION DEFAULT            HEAP64(1M,1M,KEEP,32768,32768,KEEP,4096,4096,FREE)
INSTALLATION DEFAULT            INFOMSGFILTER(OFF,,,,)
INSTALLATION DEFAULT            IOHEAP64(1M,1M,FREE,12288,8192,FREE,4096,4096,FREE)
INSTALLATION DEFAULT            LIBHEAP64(1M,1M,FREE,16384,8192,FREE,8192,4096,FREE)
DD:CEEOPTS                      NATLANG(UEN)
.
.
```

fdlibm Replacements

- **Problem Statement**
  - In 1999, the fdlibm (Freely Distributed LIBM) library from Sun Microsystems was ported to support the Java language requirement that all platforms must produce the same results as the fdlibm library.
  - Subsequent to the XL C/C++ Run-Time Library release of IEEE754 floating-point math support, Java provided their own support and the requirement on XL C/C++ was no longer needed.
- **Solution**
  - Since the XL C/C++ Run-Time Library no longer needs to honor the bit-wise requirement for Java's math library, the most heavily used math functions will be replaced by functions designed to be better performing and more accurate.

fdlibm Replacements

- **Two methods are provided for users who wish to continue using the original fdlibm versions of the functions:**

1. **#define _IEEEV1_COMPATIBILITY**
   - The use of this feature test macro in an application will cause the original fdlibm functions to be used when compiled FLOAT(IEEE) and _FP_MODE_VARIABLE is not defined. This method requires the application to be recompiled.

2. **_EDC_IEEEV1_COMPATIBILITY_ENV=ON**
   - If the application has not included math.h or uses feature test macro _FP_MODE_VARIABLE, then environment variable _EDC_IEEEV1_COMPATIBILITY_ENV can be set to ON in order to access the original versions of the functions. If the environment variable is unset or set to any value other than ON, the new versions of the functions will be used. This method does not require the application to be recompiled.

errno2 Enhancements

- **Problem Statement**
  - Due to the reuse of the same errno values in multiple failure paths, both within a function and cross functions, it is sometimes difficult to identify the cause of a failure.
- **Solution**
  - perror() should always provide an errno2 value.
  - A unique errno2 value will now be set for each failure path in the XL C/C++ Run-Time Library where errno is also set.
  - The _EDC_ADD_ERRNO2 environment variable behavior is changed such that when unset, the errno2 value is appended to perror() messages.
  - A new edcmtext utility is added that will provide a description and recommended action to resolve the problem for the corresponding XL C/C++ Run-Time Library errno2 value.
  - The bpxmtext utility has been modified to call edcmtext when the errno2 value is in the XL C/C++ Run-Time Library range.

errno2 Enhancements

- **The following are existing resources that can be used to obtain the errno2 value:**
  - The user can obtain the value for errno2 by calling the **perror()** function. By default, **perror()** produces a message with current errno2 value appended to the end of the string.
  - Also, an **__errno2()** function call returns the value of the errno2 variable.
  - Otherwise, the errno2 value can be found on inspection of a CEEDUMP or IPCS verbexit LEDATA output.

- **Note: errno2 values are not programmer interfaces, and can change without notice.**

errno2 Enhancements

- **The following is an example of a perror() call and corresponding output:**
  - Code: `perror("fopen() failed");`
  - Output: `fopen() failed: EDC5121I Invalid argument. (errno2=0xC00B0021)`

- **The errno2 value can then be used as input into edcmtext or bpxmtext to obtain further information about the reason code encountered.**
  - An example of the edcmtext tool: `edcmtext C00B0021`
  - The above example produces the following:
    - JrEdc1opsEinval01: The mode argument passed to fopen() or freopen() did not begin with r, w, or a.
    - Action: Correct the mode argument. The first keyword of the mode argument must be the open mode. Ensure the open mode is specified first and begins with r, w, or a.
    - Source: edc1opst.c

C/C++ SUSv3

- **Problem Statement**
  - z/OS must support the Single UNIX Specification, Version 3 (SUSv3) standard in order to host new customer applications, facilitate porting, and enhance flexibility.
- **Solution**
  - Language Environment now provides a C/C++ run-time library and compilation environment that aligns substantially with the SUSv3 standard.
    - This makes it easier to develop, port, or deploy modern C/C++ applications on z/OS.
    - Though the Language Environment C/C++ run-time library is substantially compliant with SUSv3, it is not fully complaint and will not apply for UNIX03 brand.
  - An SUSv3 compliant threading interface is provided, as well as other header content, missing interfaces, and new SUSv3 behavior.

Multicast Source Filter APIs

- **Problem Statement**
  - In z/OS V1.9, Comm Server is adding support for MLDv2 and IGMPv3.
  - The existing C/C++ IP Multicast APIs do not support source filtering.
- **Solution**
  - The existing C/C++ IP Multicast APIs allow a receiver application to specify the group address (destination) and (optionally) the local interface. These APIs have not changed.
  - New source filter C/C++ APIs are now available and provide the same functionality as the existing APIs, yet they also allow receiver multicast applications to:
    - Specify zero or more unicast (source) address(es) in a source filter.
    - Determine whether the source filter describes an inclusive or exclusive list of sources.
  - 4 new APIs, a set of new socket options, and new structures are now available, supporting both IPv4 and IPv6.

Heap Pools Performance Improvements

- **Problem Statement**
  - The performance of heap pools can be improved by reducing cache contention when they are used.
    - Cache contention occurs when two CPs try to update the same storage address at the same time. This contention may be unavoidable.
    - But cache contention may also occur when two CPs update different storage addresses that are close together at the same time.
- **Solution**
  - New support is available that eliminates some of this type of contention by rearranging how the heap pool structures are laid out in storage.
  - However, doing so uses more virtual storage.

Heap Pools Performance Improvements

- **A new keyword ALIGN is available for the first suboption of the HEAPPOOLS run-time option:**
  - **OFF** – Specifies that Language Environment does not use the Heap Pools Manager.
  - **ON** - Specifies that Language Environment does use the Heap Pool Manager to manage heap storage requests against the initial heap.
  - **ALIGN** - Specifies that Language Environment will structure the storage for cells in a heap pool so that a cell less than or equal to 248 bytes does not cross a cache line. For cells larger than 248 bytes, two cells never share a cache line.

- **A new keyword ALIGN is available for the first suboption of the HEAPPOOLS64 run-time option:**
  - **OFF** – Specifies that Language Environment does not use the Heap Pools Manager.
  - **ON** - Specifies that Language Environment does use the Heap Pool Manager to manage heap storage requests against the initial heap. .
  - **ALIGN** - Specifies that Language Environment will structure the storage for cells in a heap pool so that a cell less than or equal to 240 bytes does not cross a cache line. For cells larger than 240 bytes, two cells never share a cache line.

IEEE Decimal Floating Point

- **Problem Statement**
  - Support is required by the C/C++ RTL to fulfill the overall zSeries and z/OS support of decimal floating point numbers.
- **Solution**
  - Language Environment is providing XL C/C++ Run-Time Library support for decimal floating point.
    - This support, in conjunction with the z/OS V1.9 XL C/C++ compiler, will allow C/C++ applications to exploit the decimal floating point instruction set.
  - The support provided is based on a core subset of the following standards (which are all currently in draft state):
    - **ANSI/IEEE Standard P754/D0.15.3** - IEEE Standard for Floating-Point Arithmetic
    - **ISO/IEC TR24732** - Extensions for the programming language C to support decimal floating point arithmetic.
    - **ISO/IEC TR24733** - Extensions for the programming language C++ to support decimal floating point arithmetic.

AMODE 64 CEETBCK and CEEHGOTO

- **Problem Statement**
  - Support was requested for AMODE 64 services that are equivalent to the CEETBCK and CEEGOTO callable services.

- **Solution**
  - Support for __le_traceback() (CEETBCK) and __far_jump() (CEEHGOTO) was added to the AMODE 64 C/C++ run-time library.
    - The function **__le_traceback()** can be used to generate a traceback instead of calling ctrace() or cdump() when the user desires a different format or different location for the traceback.
    - The function **__far_jump()** can be used in a signal catcher, exception handler or debugger to return control to the application.

Metal C Run-Time Library

- **Problem Statement**
  - Programmers exploiting the Metal C compiler support would find standard C functions useful for:
    - Utility functions for manipulating data
    - Memory management
- **Solution**
  - A simple run-time library has been provided for use with the Metal C compiler support.
  - Allows for improved programmer productivity, reducing the need for programmer to reinvent basic functions.

What's New in z/OS V1.8?

- DYNDUMP - Dynamic SVC Dumps
- Callable Services
- _CEE_ENVFILE_S
- 64-bit CEEDUMP Statement Numbers
- XPLINK Transitions Tracing
- C/C++ I/O Support
- STORAGE(,,CLEAR)
- CEEENTRY MAIN=YES under CICS
- Euro Currency Symbol Update

DYNDUMP- Dynamic SVC Dumps

- **Problem Statement**
  - Valuable debug information can be lost if a job ABENDs and it does not specify a SYSMDUMP DD card (or similar DD card).
- **Solution**
  - DYNDUMP run-time option is being provided. If no dump data sets have been provided via a DD statement, DYNDUMP tells Language Environment to automatically allocate a dump data set and request an IPCS readable transaction dump.
  - The DYNDUMP will be written when:
    - User selectable ABENDs occur.
    - No SYSMDUMP (or similar) DD card has been specified (this can be overridden).
    - The dump data set can be allocated.

DYNDUMP- Dynamic SVC Dumps

- **DYNDUMP**(*hlq*, NODYNAMIC|DYNAMIC|FORCE|BOTH, TDUMP|NOTDUMP)

  - ***hlq*** - High Level Qualifier for the MVS dump data set to be created. This will be concatenated with a timestamp consisting of the julian day and the time of the dump. The Job name or PID will also be part of the name if there is room.
    - Two keywords are allowed for the hlq (in place of a user-specified hlq):
      - ***USERID** – Tells Language Environment to use the userid associated with the job step task as the hlq for the dump data set. If the hlq is blank, it is the same as *USERID.
      - ***TSOPREFIX** – Instructs Language Environment to use the TSO/E prefix. The prefix is only valid in a TSO/E environment. If the prefix is not available, the userid will be used.
        - **Note:** z/OS V1.9 provides support allowing each keyword to be followed by additional characters that will be used to create the data set name. When appended to the userid or the TSO prefix, they would form the hlq used when creating the dump data set.

DYNDUMP- Dynamic SVC Dumps

- **The following keywords are used for 4039 ABENDS only:**
  - **DYNAMIC** – Language Environment will automatically create an IPCS readable dump when the user (application) has not already specified one of the dump ddnames, e.g. SYSUDUMP.
  - **NODYNAMIC** - Language Environment will not create an IPCS readable dump.
  - **FORCE** - Language Environment will always create an IPCS readable dump, even if other dump ddnames have been specified. The SYSMDUMP DD card will be ignored (if it exists).
  - **BOTH** - Similar to FORCE, but there will be two dumps created, the transaction dump and SYSMDUMP (or SYSUDUMP, SYSABEND).
- **The following keywords are used for all other 40xx ABENDS (non-4039):**
  - **TDUMP** – Language Environment will automatically create an IPCS readable dump.
  - **NOTDUMP** – Language Environment will not create an IPCS readable dump.

Callable Services

CEE3INF - Query Enclave Information

- **Problem Statement**
  - Service was requested for determining whether a thread was running in a CICS or Batch environment.
- **Solution**
  - The new CEE3INF callable service queries and returns information about the enclave, such as:
    1. **Operating system and subsystem in use** (CICS, CICS_PIPI, TSO, Batch, USS, z/VSE, z/OS, z/OS.e)
    2. **Environments active in the enclave** (PIPI, PIPI_Main, PIPI_Sub, PIPI_Subdp, PICI, Nested Enclave, LRR, Run-time Reuse, XPLINK(ON), POSIX(ON), at least one pthread created in enclave, currently on IPT, multithreaded fork in effect in current enclave, AMODE Init (24 or 31))
    3. **Member languages currently initialized** (C/C++, COBOL, Fortran, PL/I, Enterprise PL/I)
    4. **Version of Language Environment**

Callable Services

CEE3ABD - Terminate enclave with ABEND

- **Problem Statement**
  - The existing CEE3ABD service generates dumps based on the setting of the TERMTHDACT run-time option, and users are unable to control the type of dump in any other way.
- **Solution**
  - CEE3ABD and __cabend() have been updated to take additional cleanup values that specify the type of dumps to be generated (CEEDUMP, system dump, both or none).

Callable Services

CEE3AB2 - Terminate enclave with an ABEND and reason code

- **Problem Statement**
  - There was no service available for users to request enclave termination with an ABEND and a user defined reason code.
- **Solution**
  - The CEE3AB2 callable service has been added to request that Language Environment terminate the enclave with an ABEND and user defined reason code. The user can specify if the ABEND is to be issued with or without cleanup, and the type of dumps to be generated (CEEDUMP, system dump, both or none).

Callable Services

CEEENV - Process Environment Variables

- **Problem Statement**
  - Needed a language neutral service for manipulating environment variables.
- **Solution**
  - CEEENV is a new callable service for processing environment variables depending on the function code passed in as input. Based on the input to this function, CEEENV can do the following:
    - Obtain the value for an existing environment variable
    - Create a new environment variable with a value
    - Clear all environment variables
    - Delete an existing environment variable
    - Overwrite the value for an existing environment variable

Callable Services

CEE3PR2 - Query Parameter String Long

- **Problem Statement**
  - CEE3PRM queries and returns to the calling routine the parameter string specified at invocation of the program. If the parameter string is longer than 80 characters, it is truncated.
- **Solution**
  - CEE3PR2 has been added. It is similar to CEE3PRM, however it will allow parameter strings of length greater than 80 to be returned to the caller.

_CEE_ENVFILE_S

- **Problem Statement**
  - Existing **_CEE_ENVFILE** environment variable doesn't strip trailing whitespace from each name=value line read from a file. Data sets with fixed-length record format are not recommended because they enable padding with blanks and the blanks are counted when calculating the size of the line.
- **Solution**
  - **_CEE_ENVFILE_S** environment variable enables a list of environment variables to be set from a specified file, stripping trailing whitespace from each NAME=VALUE line read. This environment variable only takes effect when it is set through the run-time option ENVAR on initialization of a parent program.

64-bit CEEDUMP Statement Numbers

- **Problem Statement**
  - Statement numbers are available in 31-bit CEEDUMP tracebacks but not in 64-bit CEEDUMP tracebacks.
- **Solution**
  - A 'Statement' column was added to the 64-bit CEEDUMP traceback to reflect the compile unit statement number.
  - A traceback section named 'Fully Qualified Names' was also added. This will provide the fully qualified data set name or pathname of the source code and load module names if available.
  - The statement number support is made available with the use of the GONUMBER or DEBUG C/C++ compile options.
  - The following examples enable the 64-bit CEEDUMP statement number support (using the c89 compile utility):
    - c89 -Wc,**GONUM**,**LP64** -Wl,**LP64** file.c
    - c89 **–g** -Wc,**LP64** -Wl,**LP64** file.c

XPLINK Transitions Tracing

- **Problem Statement**
  - While using XPLINK can greatly increase the performance of C/C++ applications, transitions between XPLINK and non-XPLINK code can reduce its effectiveness. It is often difficult or impossible to determine where in the application these transitions are occurring.
- **Solution**
  - A new trace level has been added to the existing Language Environment TRACE run-time option to trace these transitions.
  - The following is an example of setting the TRACE run-time option to trace the transitions between XPLINK and non-XPLINK:

```
TRACE=(ON,32K,DUMP,LE=20)
```

C/C++ I/O Support

- **Large Format Sequential Datasets**
  - Full support is provided for large format sequential data sets opened using QSAM (noseek) and partial support for large format data sets opened using BSAM (seek).

- **VSAM Extended Addressability**
  - Support is provided for VSAM data sets defined with the extended addressability attribute. Support applies to key sequenced data sets (KSDS), entry sequenced data sets (ESDS), and relative-record data sets (RRDS). Applications can now read/write/position beyond the 4GB boundary in VSAM data sets.

C/C++ I/O Support

- **flockfile() family of functions**
  - The flockfile() family of functions have been added, allowing multi-threaded applications to delineate a sequence of mutually exclusive I/O statements that are executed as a unit on the same FILE* object.

- **Multi-Volume I/O**
  - The internal implementation of the repositioning functions ftell(), ftello(), fseek(), fseeko(), fgetpos() and fsetpos() have been redesigned to improve repositioning performance on multivolume datasets.
    - For the best bang for your buck, use fgetpos()/fsetpos() rather than fseek()/ftell(), if possible.

Appendix

- What's New in z/OS V1.7?
  - RTO PARMLIB
  - DD:CEEOPTS
  - C99
- Additional Reference Material for z/OS V1.9
  - 2005 GWP Compliance and Turkish Lira Currency Update
- Sources of Additional Information

RTO PARMLIB

- **Problem Statement**
  - Need to set installation-wide default run-time options without having to assemble and link CEEDOPT
  - Usability enhancement
- **Solution**
  - Provide a PARMLIB member to be read at system IPL, or established after IPL, that can contain system-wide default run-time options

Note: The use of CEEPRMxx is optional

RTO PARMLIB

- **Overview**
  - Groups of Language Environment run-time options
    - CEECOPT
    - CEEDOPT
    - CELQDOPT
  - Options in each group delimited using "(" and ")"
  - There can be multiple instances of any group within the member

RTO PARMLIB

- **CEEPRMxx Contents**
  - Acceptable options are those allowed in the assembler parts of the same name
  - Option syntax is invocation command style, e.g. JCL PARM= or TSO command line
  - Blanks or commas can be used as separators between options
  - Comments are allowed, use /* and */, but may not span lines

Note: SCEESAMP(CEEPRM00) contains sample

RTO PARMLIB

```
SYS1.PARMLIB(CEEPRM31)
****** ***************************** Top of Data ************
000001 CEEDOPT( /* Undo Favour 31 */
000002          ALL31(OFF),
000003          STACK(128K,128K,BELOW,KEEP,512K,128K),
000004          THREADSTACK(OFF,4K,4K,BELOW,KEEP,128K,128K),
000005        )
****** ***************************** Bottom of Data **********
```

```
SYS1.PARMLIB(CEEPRMTZ)
****** ***************************** Top of Data ************
000001 CEEDOPT(  ENVAR('TZ=EST5EDT') ) /* Local Timezone */
000002 CEECOPT(  ENVAR('TZ=EST5EDT') )
000003 CELQDOPT( ENVAR('TZ=EST5EDT') )
****** ***************************** Bottom of Data **********
```

RTO PARMLIB

- **During IPL**
  - IEASYSxx
    - The CEEPRMxx member(s) can be specified by adding **CEE=(xx,yy,L)** in the IEASYSxx member used to IPL the system
  - The default IEASYS00 does not specify the CEE parameter
  - System parameters
    - R 0,SYSP=(aa,bb),**CEE=(xx,yy,L)**

RTO PARMLIB

- **During IPL Messages**
  - IEE252I MEMBER CEEPRMxx FOUND IN SYS1.PARMLIB
    - when the member is found at IPL
  - IEA301I CEEPRMxx NOT FOUND IN PARMLIB
    - when the member is not found at IPL
  - CEE3738E A CEE= PARMLIB MEMBER WAS NOT FOUND OR IS IN ERROR.
    - accompanies IEA301I if it was issued
  - CEE3737I CONTENTS OF PARMLIB MEMBER CEEPRMxx
    - when contents are requested through the list option
  - CEE3739I LANGUAGE ENVIRONMENT INITIALIZATION COMPLETE
    - after Language Environment has completed its initialization during IPL (even when no CEEPRMxx member is processed)

RTO PARMLIB

- **SET CEE Command**
  - Once the system is running, the CEEPRMxx member(s) can be specified using the **SET CEE=(xx,yy,L)** console command
  - A successful SET CEE command **replaces** the currently active CEEPRMxx member(s) **AND** SETCEE commands, if any

RTO PARMLIB

- **SET CEE Messages**
  - IEE538I CEEPRMxx MEMBER NOT FOUND IN PARMLIB
    - when the SET CEE command specifies a member that does not exist
  - CEE3737I CONTENTS OF PARMLIB MEMBER CEEPRMxx
    - when the SET CEE command specifies the list option
  - CEE3742I THE SET CEE COMMAND HAS COMPLETED.
    - when the SET CEE command completes successfully

RTO PARMLIB

- **CEEPRMxx Errors**
  - CEE3732I AN END OF COMMENT DELIMITER IS MISSING ON LINE nn OF PARMLIB MEMBER CEEPRMxx
    - Missing comment delimiter, no */ on the line.
  - CEE3733I THE EOF FOR PARMLIB MEMBER CEEPRMxx WAS REACHED BEFORE FINDING A CLOSING OPTION GROUP DELIMITER FOR THE OPTIONS GROUP CExxxxxx
    - Missing group delimiter, no closing parenthesis
  - CEE3734I THE EOF FOR PARMLIB MEMBER CEEPRMxx WAS REACHED BEFORE FINDING A CLOSING QUOTE DELIMITER WHILE PROCESSING THE OPTIONS GROUP CExxxxxx
    - Missing either a " or ' within an option

RTO PARMLIB

- **CEEPRMxx Errors**
  - CEE3735I TEXT ON LINE nn STARTING AT COLUMN mm IS OUTSIDE OF AN OPTIONS GROUP IN PARMLIB MEMBER CEEPRMxx
    - example for CEE3735I:

      ceedopt(all31(off),stack(,,below))

      testing notes:

      ceecopt(posix(off))
    - the line with "testing notes:" will be flagged as an error
  - CEE3736I ERROR CODE ee WHILE READING LINE NUMBER nn OF PARMLIB MEMBER CEEPRMxx
    - should never happen!

RTO PARMLIB

- **CEEPRMxx Errors**
  - CEE3731I THE FOLLOWING MESSAGES PERTAIN TO THE SYSTEM DEFAULT RUN-TIME OPTIONS IN THE OPTIONS GROUP CExxxxxx IN PARMLIB MEMBER CEEPRMxx
    - Issued when there is at least one parsing error
    - Followed by existing options parsing errors
    - Similar to CEE3608I (invocation options) and CEE3627I (programmer options) etc.
  - CEE3741I THE RUN-TIME OPTION option WAS NOT VALID FROM THE SYSTEM DEFAULT OPTIONS.
    - Example is the XPLINK option

RTO PARMLIB

- **SETCEE Command**
  - While the system is running, you can use the **SETCEE** command to update the system-wide run-time options

```
syntax:
SETCEE group,option[[,option]...]

example:
SETCEE ceedopt,rpto(on),rpts(on)
```

RTO PARMLIB

- **D CEE Command**
  - While the system is running, you can use the **D CEE** command to display the system-wide run-time options in effect

```
syntax:
D CEE[,group]

example:
D CEE,ceedopt
```

RTO PARMLIB

- **D CEE Command**

```
d cee
CEE3744I 16.14.06 DISPLAY
NO MEMBERS SPECIFIED

set cee=(31,tz)
CEE3742I THE SET CEE COMMAND HAS COMPLETED.

d cee
CEE3744I 16.14.30 DISPLAY
CEE=(31,TZ)

setcee ceedopt,rptopts(on)
CEE3743I THE SETCEE COMMAND HAS COMPLETED.
```

RTO PARMLIB

- **D CEE Command**

```
d cee,ceedopt
CEE3745I 16.15.03 DISPLAY CEEDOPT
CEE=(31,TZ)
 LAST WHERE SET        OPTION
 ------------------------------------------------------------
 PARMLIB(CEEPRM31)      ALL31(OFF)
 PARMLIB(CEEPRMTZ)      ENVAR("TZ=EST5EDT")
 SETCEE command         RPTOPTS(ON)
 PARMLIB(CEEPRM31)      STACK(131072,131072,BELOW,KEEP,524288,
                          131072)
 PARMLIB(CEEPRM31)      THREADSTACK(OFF,4096,4096,BELOW,KEEP,
                          131072,131072)
```

RTO PARMLIB

- **Order of Precedence**
  - Options in the storage tuning exit (CEECSTX, CEEBSTX)
  - Options in the assembler user exit (CEEBXITA)
  - Invocation (JCL PARM= parameter, TSO command line, _CEE_RUNOPTS)
  - DD:CEEOPTS
  - Programmer default (CEEUOPT, #pragma runopts, PLIXOPT)
  - Region wide CICS or IMS/LRR default (CEEROPT)
  - **System wide (CEEPRMxx and SETCEE commands)**
  - Installation defaults (CEEDOPT/CEECOPT/CELQDOPT)

RTO PARMLIB

- **Dynamic changes to the system-wide run-time options will not be picked up by:**
  - a CICS partition that is already running
  - an IMS/LRR region that is already running
  - a CEEPIPI environment that is already initialized
  - a Language Environment application that is already initialized

- **Non-overridable**
  - During initialization of a Language Environment application, when the run-time options defined at the system level are merged with the installation defaults (CEEDOPT/CEECOPT/CELQDOPT) there may be errors reported if some options in the installation defaults are flagged as non-overridable.

DD:CEEOPTS

- **Problem Statement**
  - JCL PARM= limit of 100 characters not sufficient
  - CEEUOPT not always an option
  - CEEROPT not available for batch programs
  - Can't get options into Language Environment applications called from non-Language Environment-conforming assembler
- **Solution**
  - Look for the CEEOPTS DD card where run-time options can be specified

DD:CEEOPTS

- A CEEOPTS DD card can be used to specify run-time options via:
  - in-stream JCL
  - sequential data set
  - member of partitioned data set
- Merged after programmer default options, but before invocation command options

DD:CEEOPTS

- **Example**

```
****** ***************************** Top of Data ************
000001 * This line is a comment
000002 ALL31(OFF),STACK(,,BELOW)
000003 TRAP(ON,
000004   NOSPIE
000005 * This line is a comment within an option
000006 ),TERMTHDACT(
000007 UAIMM,
000008 ,96)
****** **************************** Bottom of Data **********
```

DD:CEEOPTS

- **Options Report**

```
Options Report for Enclave main 08/11/05 9:37:41 PM
Language Environment V01 R07.00

LAST WHERE SET                  OPTION
------------------------------------------------------------------
Installation default              ABPERC(NONE)
Installation default              ABTERMENC(ABEND)
Installation default              NOAIXBLD
DD:CEEOPTS                        ALL31(OFF)
Installation default              ANYHEAP(16384,8192,ANYWHERE,FREE)
Installation default              NOAUTOTASK
Installation default              BELOWHEAP(8192,4096,FREE)
Installation default              CBLOPTS(ON)
...
```

DD:CEEOPTS

- Order of Precedence
  - Options in the storage tuning exit (CEECSTX, CEEBSTX)
  - Options in the assembler user exit (CEEBXITA)
  - Invocation (JCL PARM= parameter, TSO command line, _CEE_RUNOPTS)
  - **DD:CEEOPTS**
  - Programmer default (CEEUOPT, #pragma runopts, PLIXOPT)
  - Region wide CICS or IMS/LRR default (CEEROPT)
  - System wide (CEEPRMxx and SETCEE commands)
  - Installation defaults (CEEDOPT/CEECOPT/CELQDOPT)

C99

- **Overview**
  - z/OS V1.7 is designed to support the ISO/IEC 9899:1999 standard
    - z/OS V1.7 XL C compiler
    - z/OS V1.7 Language Environment C/C++ Run-time Library

C99

- **Highlights**
  - #define _ISOC99_SOURCE
    - library support without compiler
  - LANGLVL(STDC99)
    - compiler and library support
  - LANGLVL(EXTC99)
    - compiler and library support plus extended language features

C99

- **Highlights**
  - printf & scanf family of functions
    - new length modifiers hh, j, t, z
    - new format specifiers F, a, A
    - new functions
  - numeric conversion functions
    - hexadecimal floating point notation
    - new functions
  - real math
  - complex math
  - type generic math
  - floating point environment

2005 GWP Compliance and Turkish Lira Currency Update

- **The following ASCII locales have been added for 2005 GWP Compliance:**

HFS-Name	Language and Country
sq_AL.UTF-8	Albanian, Albania
bg_BG.UTF-8	Bulgarian, Bulgaria
ca_ES.UTF-8	Catalan, Spain
ca_ES.UTF-8@preeuro	Catalan, Spain
cs_CZ.UTF-8@euro	Czech, Czech Republic
da_DK.UTF-8@euro	Danish, Denmark
nl_BE.UTF-8	Dutch, Belgium
nl_BE.UTF-8@preeuro	Dutch, Belgium
nl_NL.UTF-8@preeuro	Dutch, Netherlands
en_BE.UTF-8	English, Belgium
en_BE.UTF-8@preeuro	English, Belgium
en_IE.UTF-8	English, Ireland

HFS-Name	Language and Country
en_IE.UTF-8@preeuro	English, Ireland
en_GB.UTF-8@euro	English, Great Britain
et_EE.UTF-8	Estonian, Estonia
et_EE.UTF-8@euro	Estonian, Estonia
fi_FI.UTF-8@preeuro	Finnish, Finland
fr_BE.UTF-8@preeuro	French, Belgium
fr_FR.UTF-8@preeuro	French, France
fr_LU.UTF-8	French, Luxembourg
fr_LU.UTF-8@preeuro	French, Luxembourg
de_AT.UTF-8	German, Austria
de_AT.UTF-8@preeuro	German, Austria
de_DE.UTF-8@preeuro	German, Germany

2005 GWP Compliance and Turkish Lira Currency Update

HFS-Name	Language and Country
de_LU.UTF-8	German, Luxembourg
de_LU.UTF-8@preeuro	German, Luxembourg
el_GR.UTF-8@preeuro	Greek, Greece
hu_HU.UTF-8@euro	Hungarian, Hungary
is_IS.UTF-8	Icelandic, Iceland
it_IT.UTF-8@preeuro	Italian, Italy
lv_LV.UTF-8	Latvian, Latvia
lv_LV.UTF8@euro	Latvian, Latvia
lt_LT.UTF-8	Lithuanian, Lithuania
lt_LT.UTF8@euro	Lithuanian, Lithuania
mt_MT.UTF-8	Maltese, Malta
mt_MT.UTF-8@euro	Maltese, Malta
nb_NO.UTF-8	Norwegian Bokmal, Norway
pl_PL.UTF-8@euro	Polish, Poland

HFS-Name	Language and Country
pt_PT.UTF-8@preeuro	Portuguese, Portugal
sr_CS.UTF-8	Serbian (Cyrillic), Serbia and Montenegro
sh_CS.UTF-8	Serbian (Latin), Serbia and Montenegro
sk_SK.UTF-8@euro	Slovak, Slovakia
sl_SI.UTF-8@euro	Slovene, Slovenia
es_ES.UTF-8@preeuro	Spanish, Spain
sv_SE.UTF-8@euro	Swedish, Sweden
cy_GB.UTF-8	Welsh, Great Britain
cy_GB.UTF-8@euro	Welsh, Great Britain
kn_IN.UTF-8	Kannada, India
vi_VN.UTF-8	Vietnamese, Vietnam
zh_SG.UTF-8	Chinese, Singapore
zh_HK.UTF-8	Chinese, Hong Kong

2005 GWP Compliance and Turkish Lira Currency Update

- The new Turkish Lira currency default symbol is: YTL.
- The new Turkish Lira international currency symbol is: TRY.
- The corresponding Turkish locales have been updated to use the new Lira currency:

HFS-Name	TYPE
tr_TR.UTF-8	ASCII
tr_TR.ISO8859-9	ASCII
Tr_TR.IBM-1155	EBCDIC
Tr_TR.IBM-1026	EBCDIC

- Support for the new Turkish Lira has been provided back to z/OS V1.4 and via APAR PK16479.

Sources of 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 Debugging Guide
- Language Environment Run-Time Messages
- Language Environment Programming Reference
- Language Environment Programming Guide
- Language Environment Programming Guide for 64-bit Virtual Addressing Mode
- Language Environment Customization
- Language Environment Run-Time Application Migration Guide
- Language Environment Writing ILC Applications
- Language Environment Vendor Interfaces
- Language Environment Concepts Guide

Sources of Additional Information

- XL C/C++ Compiler and Run-Time Migration Guide for the Application Programmer
- XL C/C++ Run-Time Library Reference
- XL C/C++ Programming Guide
- XL C/C++ User's Guide
- Metal C Programming Guide and Reference
- UNIX System Services Command Reference
- MVS IPCS Commands
- CICS Supplied Transactions
- www.opengroup.org