An Introduction to Language Environment

zSTSU 2004

Mary Astley

IBM Washington Systems Center
Advanced Technical Support

Trademarks

IBM Washington
Systems Center

IBM eserver zSeries

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

AD/Cycle	DB2	S/390
C++/MVS	IBM	VisualAge
C/370	IMS	WebSphere
CICS	Language Environment	z/OS
CICS/ESA	MVS	z/OS.e
COBOL/370	OS/390	zSeries
DFSMS/MVS	OS/390 C/C++	

UNIX is a registered trademark of The Open Group in the United States and other countries.

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.

Agenda

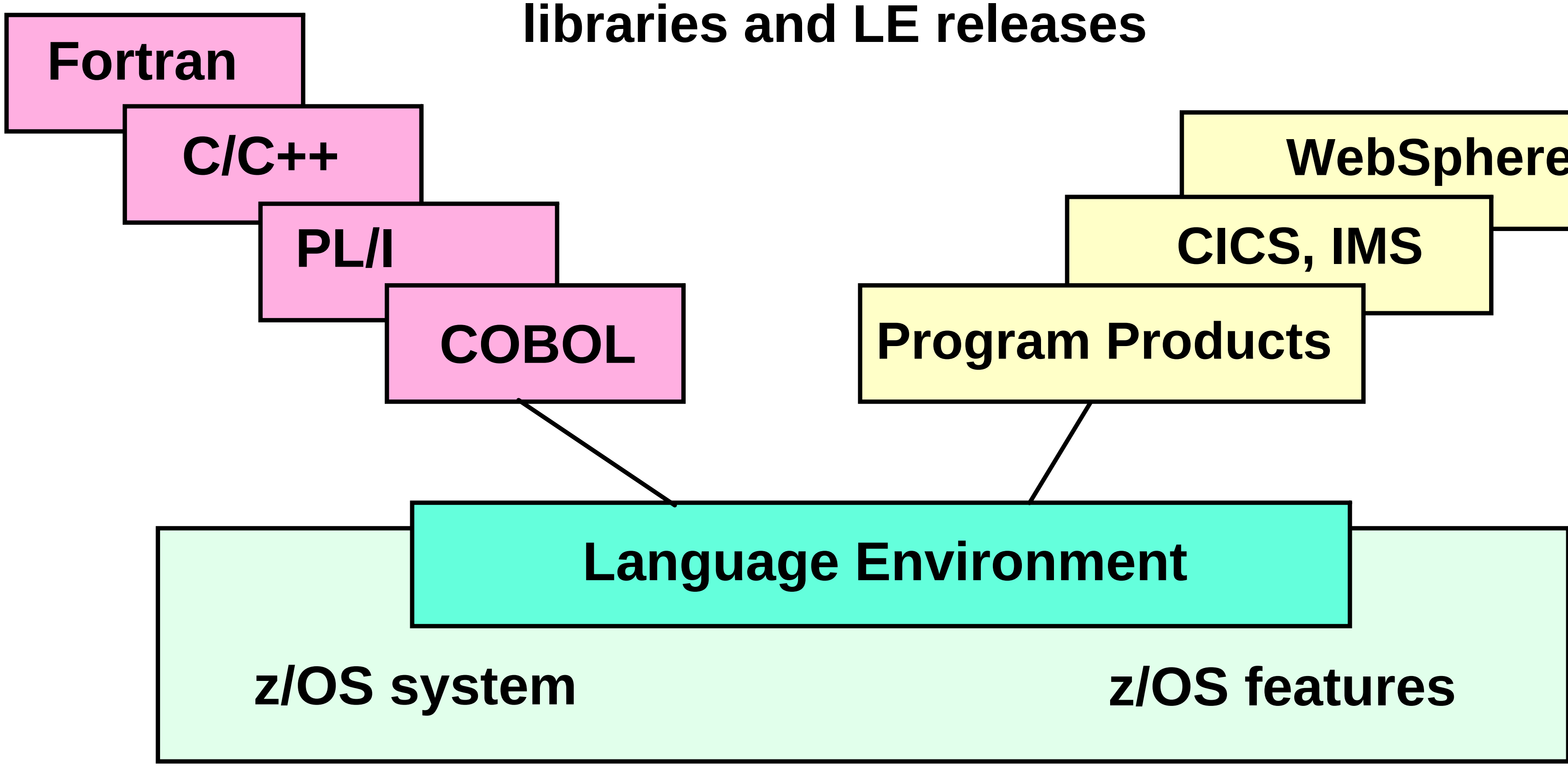
- Language Environment (LE)
- Overview
- Libraries
- Specifying Run-Time Options
- Program Model, Storage Model
- Storage Run-time Options

Why Language Environment?

IBM Washington
Systems Center

LE Provides...

- **A consistent run-time environment**
- **The run-time for today and tomorrow**
- **Consistent condition handling**
- **Compatibility with previous run-time libraries and LE releases**



z/OS LE Compatibility

IBM Washington
Systems Center

- **Upward compatibility**
  - Load modules will run compatibly with an LE level equivalent to or higher than the level used to link them
  - Object modules can be linked with an LE level equivalent to or higher than the level required by the compiler
- **Downward compatibility**
  - Ability to develop and link applications with LE on z/OS and execute them on a lower supported levels of z/OS
  - Applications must only use LE functions available on the z/OS system where they will execute
- **Compatibility with previous run-time libraries**
  - With certain exceptions, LE provides object and load module compatibility for applications generated with pre-LE compilers
  - Load modules generated with these compilers and linked with their run-time libraries will run compatibly with LE
  - Object modules can be linked and run with LE

LE Run-time Environments

With z/OS 1.6 Language Environment run-time has three forms

- **Base form**
  AMODE 31 / AMODE 24, standard linkage
- **XPLINK form**
  AMODE 31, XPLINK linkage
- **64 bit form**
  AMODE 64, XPLINK linkage

More about linkage...

- **Standard linkage**
  "traditional linkage", upward growing stack
- **XPLINK**
  efficient linkage for subroutine calls in C/C++ programs, downward growing stack

Load Modules

IBM Washington
Systems Center

The LE run-time initialized is determined by the entry point and boot strap routines in the main program

- **AMODE 24 or AMODE 31 main program**
  - Entry point is CEESTART
  - CEEROOTA bootstrap routine for non-XPLINK
  - CEEROOTD bootstrap routine for XPLINK
  - CEEBTRM termination stub
  - CEESG003 C signature CSECT
- **AMODE 64 main program**
  - Entry point is CELQSTRT
  - CELQBST bootstrap routine
  - CELQTRM termination stub
  - CELQSG03 C signature CSECT

64 bit Run-time Environment

Run-time environment for AMODE 64 programs

Uses 64 bit addresses

Supports data above the 2 GB bar

Code is executed below the 2 GB bar

For C/C++ and LE-enabled assembler programs

Requires

- z/Architecture hardware
- z/OS 1.6 Language Environment
- z/OS 1.6 C/C++ compiler

Does not support

- High level languages other than C and C++
- CICS, IMS

LE Libraries

LE libraries used during compile or assembly

- Libraries containing C/C++ header files
- SCEEMAC - Macro library for assembler programs using LE services

LE libraries used during link-edit

- Libraries containing static routines which are linked with the application and resolve external references

LE libraries used during program execution

- Libraries containing dynamic routines loaded during program execution

Other LE libraries

- SCEESAMP - sample macros, usermods, user exits
- SCEEPROC - procedures to link-edit and run LE programs

Some LE Link-Edit Libraries

Link-edit libraries for non-XPLINK applications

- **SCEELKED**
  Resident routines, including init/term, callable services
- **SCEELKEX**
  Resident routines with case sensitive names & names longer than eight character names
- **SCEEOBJ**
  Resident definitions for UNIX System Services programs
- **SCEECPP**
  Resident definitions for C++ programs

Link-edit libraries for XPLINK applications

- **SCEEBIND / SCEEBND2**
  Resident routines for XPLINK applications
- **SCEELIB**
  Side-decks for LE provided DLLs to resolve references

LE Run-time Libraries

- **SCEERUN**
  - Contains the LE run-time libraries
- **SCEERUN2**
  - PDSE with run-time routines
  - Member names in SCEERUN2 and SCEERUN are different
- **SCEECICS**
  - Contains COBOL specific CICS run-time programs
  - Included in CICS //DFHRPL DD before SCEERUN
- **SCEELPA**
  - Contains reentrant SCEERUN modules which may be included in LPALST for performance benefits
  - If SCEELPA is in LPALST, then SCEERUN should be in LNKLST

Specifying Run-Time Options

Run-Time Options Overview

IBM Washington Systems Center

- Run-time options provide a method to customize "parameters" used by Language Environment
- LE has three sets of run-time options
  - CEEDOPT - Batch and IMS (AMODE31 / AMODE24)
  - CEECOPT - CICS
  - CELQDOPT - 64 bit run-time options
- These three CSECTs supply the installation defaults for the run-time options
- IBM supplies default values for each run-time option
- An installation may change the values specified for the run-time options in these CSECTs

Where to Specify Options

- **CEEDOPT / CEECOPT**
  Installation default run-time options
- **CEEROPT**
  Region run-time options for CICS, LRR users
- **CEEUOPT**
  User run-time options linked with application program
  - PLIXOPT string in PL/I source code
  - #pragma runopts in C source code
- **PARM parameter in JCL**
  _CEE_RUNOPTS environment variable under USS

Customize Run-Time Options

- **CEEDOPT, CEECOPT, CELQDOPT**
  - **Must specify all options and suboptions**
  - **Specify if option is overrideable or nonoverrideable**
- **CEEUOPT, CELQUOPT**
  - **Only need to specify changed options & suboptions**
- **CEEROPT**
  - **Need to specify changed options with all suboptions**
  - **Specify if option is overrideable or nonoverrideable**
- **Options specified on CEEXOPT macro**
- **Samples provided in SCEESAMP library**

Options CSECT	Sample Job	Options Member
CEEDOPT	CEEWDOPT	CEEDOPT
CEECOPT	CEEWCOPT	CEECOPT
CELQDOPT	CEEWQDOP	CELQDOPT
CEEUOPT	CEEWUOPT	CEEUOPT
CELQUOPT	CEEWQUOP	CELQUOPT
CEEROPT	CEEWROPT	

Examples...


```
CEEDOPT   CSECT
          CEEXOPT ABPERC=((NONE),OVR),                X
           ...
STACK((128K,128K,ANY,KEEP,512K,128K),OVR)             X
           ...
         END
```

```
CEEUOPT   CSECT
         CEEXOPT STACK(64K,64K)
         END
```

```
CEEROPT   CSECT
         CEEXOPT
STACK((64K,64K,ANY,KEEP,512K,128K),OVR)
         END
```

Options Report

IBM Washington
Systems Center

- ➤ Reports run-time options in effect
- ➤ Request with RPTOPTS(ON) option

LAST WHERE SET column = last place option referenced

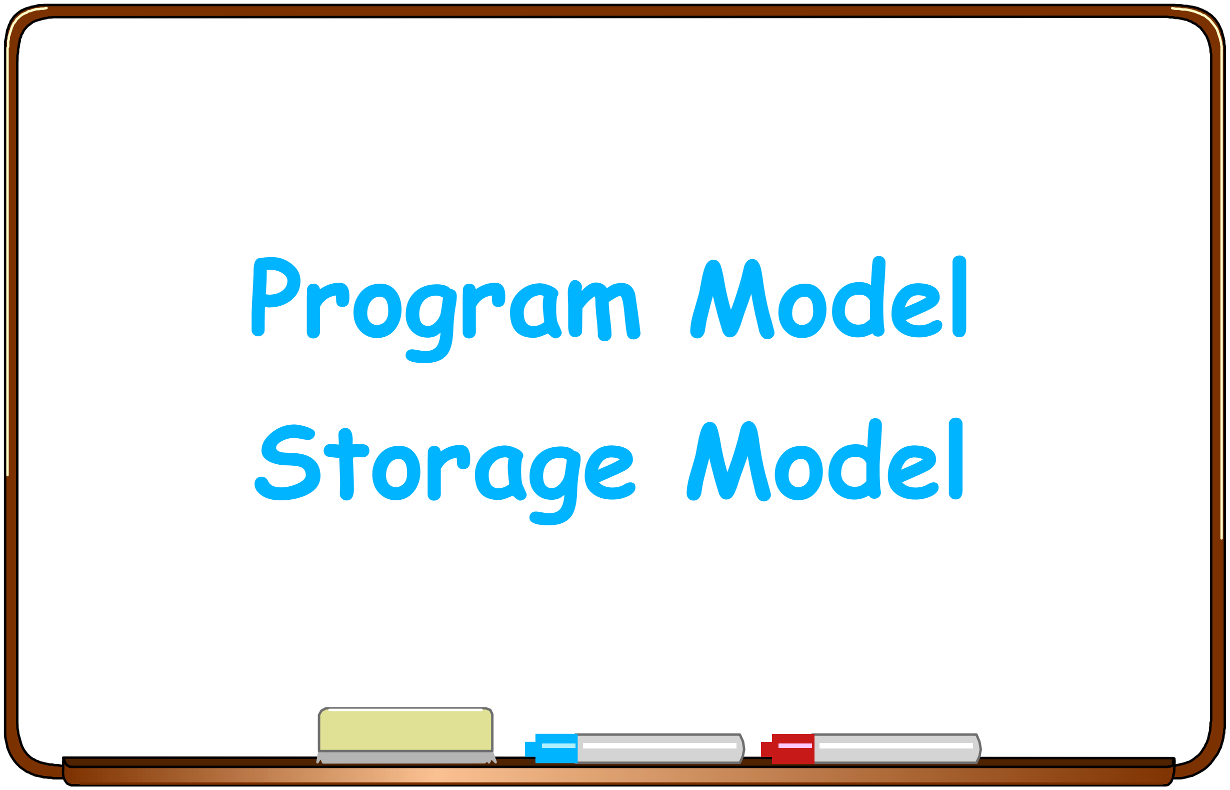
- ► Installation default
  Value from CEEDOPT, CEECOPT, CELQDOPT
- ► Default setting
  Cannot be specified in options module
- ► Programmer default
  From CEEUOPT, CELQUOPT
  #pragma runopts, PLIXOPT
- ► Invocation command
  Specified on PARM= on JCL

Sample Options Report

IBM Washington Systems Center

```
Options Report for Enclave COBTAB 11/08/04 9:49:56 AM
Language Environment V01 R05.00

LAST WHERE SET          OPTION
-------------------------------------------------------------------------------------------------
Installation default    ABPERC(NONE)
Installation default    ABTERMENC(ABEND)
Installation default    NOAIXBLD
Programmer default      ALL31(ON)
...
Default setting         NOFLOW
Invocation command      HEAP(24576,12288,ANYWHERE,KEEP,8192,4096)
...
Invocation command      RPTOPTS(ON)
Installation default    RPTSTG(OFF)
...
Programmer default      STACK(65536,65536,ANYWHERE,KEEP,524288,131072)
Installation default    STORAGE(NONE,NONE,NONE,0)
...
```

IBM Washington
Systems Center



LE Program Model

IBM Washington Systems Center

Process - Collection of resources

Resources Owned
- Enclave
- LE message file
- Shared code & constants

Enclave - Collection of routines main + subroutines

Resources Owned
- Thread
- Heap
- External files
- Static data

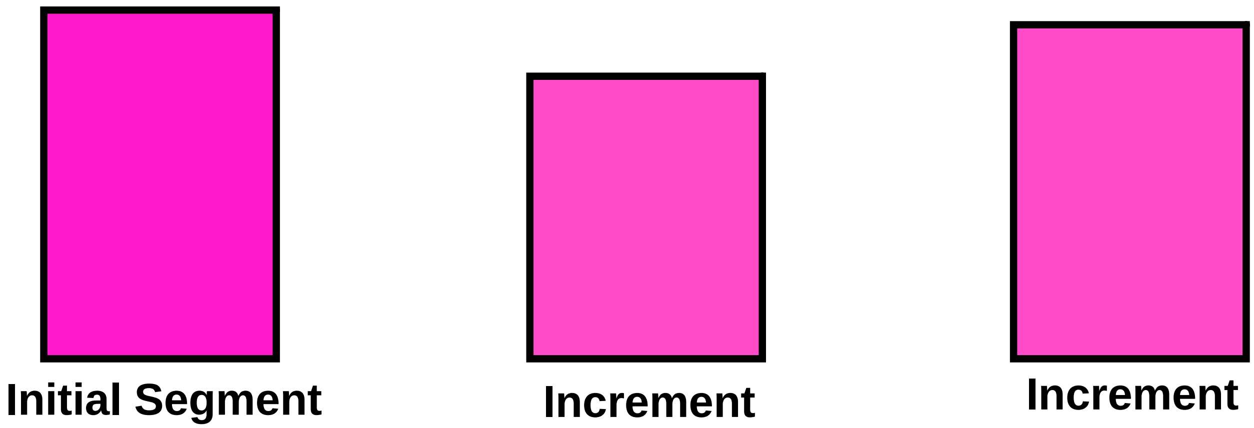
Thread - Line of execution

Resources Owned
- Stack
- Condition manager

Storage Model

IBM Washington
Systems Center

- **Stack Storage**
  - Contains program linkage, condition handling information, C/C++ and PL/I local variables
- **Heap Storage**
  - Allocated for user dynamically allocated variables
  - C malloc, PL/I ALLOCATE, COBOL working storage
  - May be allocated and freed using LE services
- **Storage is divided into segments**
  - Initial Segment + Increments



Storage Run-Time Options

ALL31 Run-Time Option

ALL31(OFF)

- Application has AMODE(24) programs
- AMODE switching is performed for calls to LE routines and callable services
- Storage use below the 16 MB line is affected, as AMODE(24) programs can only access this storage

ALL31(ON)

- All application programs are AMODE(31)
- AMODE switching minimized for calls to LE routines
  No AMODE switching for calls to LE callable services

Heap Storage Run-Time Options

HEAP run-time options specified in CEEDOPT, CEECOPT, CEEROPT, CEEUOPT

HEAP(init,incr,ANY|BELOW,FREE|KEEP,init24,incr24)
- Heap storage used by program

THREADHEAP(init,incr,ANY|BELOW,FREE|KEEP)
- Thread level heap storage

ANYHEAP(init,incr,ANY|BELOW,FREE|KEEP)
- Library heap storage not restricted to below 16 MB

BELOWHEAP(init,incr,FREE|KEEP)
- Library heap storage which must be below 16 MB

HEAPPOOLS(ON|OFF,cellsz1,percnt1,...,cellsz12,percnt12)
- Heap pools is a storage management algorithm to improve the performance of multithreaded C/C++ applications

Heap Storage Run-Time Options

HEAP run-time options specified in CELQDOPT, CELQUOPT

**HEAP64(init64,incr64,<u>KEEP</u>|FREE,init31,incr31,<u>KEEP</u>|FREE,
init24,incr24,KEEP|<u>FREE</u>)**

Heap storage in 64 bit run-time

**IOHEAP64(init64,incr64,KEEP|<u>FREE</u>,init31,incr31,KEEP|<u>FREE</u>,
init24,incr24,KEEP|<u>FREE</u>)**

Storage used by run-time when performing I/O for applications

**LIBHEAP64(init64,incr64,KEEP|<u>FREE</u>,init31,incr31,KEEP|<u>FREE</u>,
init24,incr24,KEEP|<u>FREE</u>)**

Heap storage used by LE

HEAPPOOLS64(ON|<u>OFF</u>,cellsz1,cellcnt1,...,cellsz12,cellcnt12)

Heap pool storage for C/C++ applications

Stack Storage Run-Time Options

STACK run-time options specified in CEEDOPT, CEECOPT, CEEROPT, CEEUOPT

STACK(init,incr,ANY|BELOW,FREE|KEEP,dsinit,dsincr)

- Allocation of stack storage
- If ALL31(OFF) must specify BELOW

**THREADSTACK=((ON|OFF,init,incr,ANY|BELOW,
FREE|KEEP,dsinit,dsincr)**

- Allocation of thread stack storage after initial thread

LIBSTACK(init,incr,FREE|KEEP)

- Library stack storage below 16 MB

Stack Storage Run-Time Options

IBM Washington Systems Center

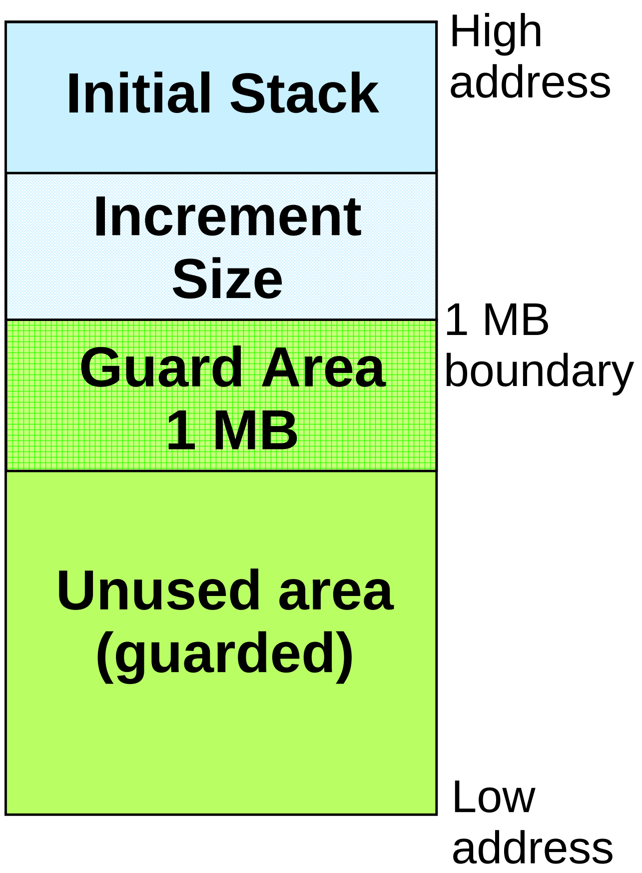
STACK run-time options specified in CELQDOPT, CELQUOPT

STACK64(init,incr,max)

- Stack storage in 64 bit run-time
- Allocated above the 2 GB bar
- max - specifies maximum stack size

THREADSTACK64(OFF|ON,init,incr,max)

- Allocation of thread stack storage after initial thread



Storage Report

Storage
Report

- ➤Provides report of LE storage usage
- ➤Summarizes stack and heap activity
- ➤Provides recommended values

Set run-time option RPTSTG(ON) to request report

```
//RUN  EXEC PGM=COBTAB,PARM='/RPTSTG(ON)'
```

Improve performance by reducing number of operating system calls to allocate and free storage

Storage Report - One Heap Request

IBM Washington
Systems Center

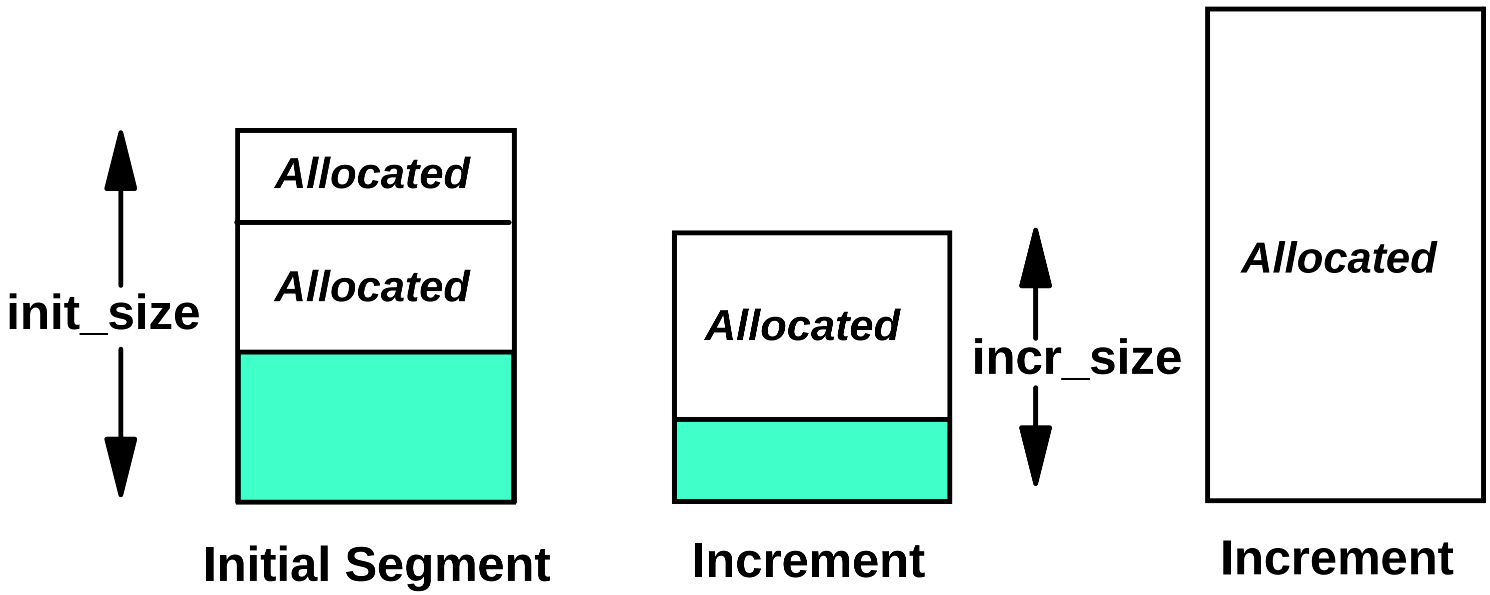
```
Storage Report for Enclave COBTAB 11/08/04 10:00:02 AM
Language Environment V01 R05.00

  STACK statistics:
    Initial size:                                          131072
    Increment size:                                        131072
    Maximum used by all concurrent threads:                  6192
    Largest used by any thread:                              6192
    Number of segments allocated:                               1
    Number of segments freed:                                   0
 ...

 HEAP statistics:
   Initial size:                                         32768
   Increment size:                                       32768
   Total heap storage used (sugg. initial size):       1280224
   Successful Get Heap requests:                             1
   Successful Free Heap requests:                            0
   Number of segments allocated:                             2
   Number of segments freed:                                 0
```

Heap Storage

IBM Washington Systems Center

HEAP(init_size,incr_size,...)

init_size and incr_size specify the minimum segment size



Storage Report - Heap(16K,8K)

IBM Washington
Systems Center

```
Storage Report for Enclave COBHEAP 11/08/04 10:09:15 AM
Language Environment V01 R05.00

  STACK statistics:
    Initial size:                                        131072
    Increment size:                                      131072
    Maximum used by all concurrent threads:                6176
    Largest used by any thread:                            6176
    Number of segments allocated:                             1
    Number of segments freed:                                 0

...

  HEAP statistics:
    Initial size:                                        16384
    Increment size:                                       8192
    Total heap storage used (sugg. initial size):       163864
    Successful Get Heap requests:                           25
    Successful Free Heap requests:                           0
    Number of segments allocated:                           14
    Number of segments freed:                                0

...
```

Storage Report - Heap(160K,8K)


```
Storage Report for Enclave COBHEAP 11/08/04 10:12:27 AM
Language Environment V01 R05.00

  STACK statistics:
    Initial size:                                      131072
    Increment size:                                    131072
    Maximum used by all concurrent threads:              6176
    Largest used by any thread:                          6176
    Number of segments allocated:                           1
    Number of segments freed:                               0

 ...

  HEAP statistics:
    Initial size:                                      163840
    Increment size:                                      8192
    Total heap storage used (sugg. initial size):      163448
    Successful Get Heap requests:                          25
    Successful Free Heap requests:                          0
    Number of segments allocated:                           1
    Number of segments freed:                               0
 ...
```

Storage Run-Time Option

**STORAGE(heap_alloc_value,heap_free_value,
dsa_alloc_value,reserve_size)**

Storage option controls initial content of storage when it is allocated and freed

If NONE is specified, storage is not initialized

- **heap_alloc_value**
  Allocated heap storage initialized to this value
- **heap_free_value**
  Freed heap storage initialized to this value
- **dsa_alloc_value**
  Stack storage initialized to this value
- **reserve_size**
  Storage reserved for out of storage condition

Condition Handling
Run-time Options

Condition Severity Levels

- **LE defines a condition as:**

  Any event requiring the attention of a running application or HLL routine supporting the application

  Condition is also known as an exception, interrupt, signal, and error

- **Severity levels and default LE responses:**

  0 - Informative message

  1 - Warning message

  2 - Error program terminated

  3 - Severe error program terminated

  4 - Critical error program terminated

Condition Handling Model

IBM Washington
Systems Center

Condition signaled as a result of:

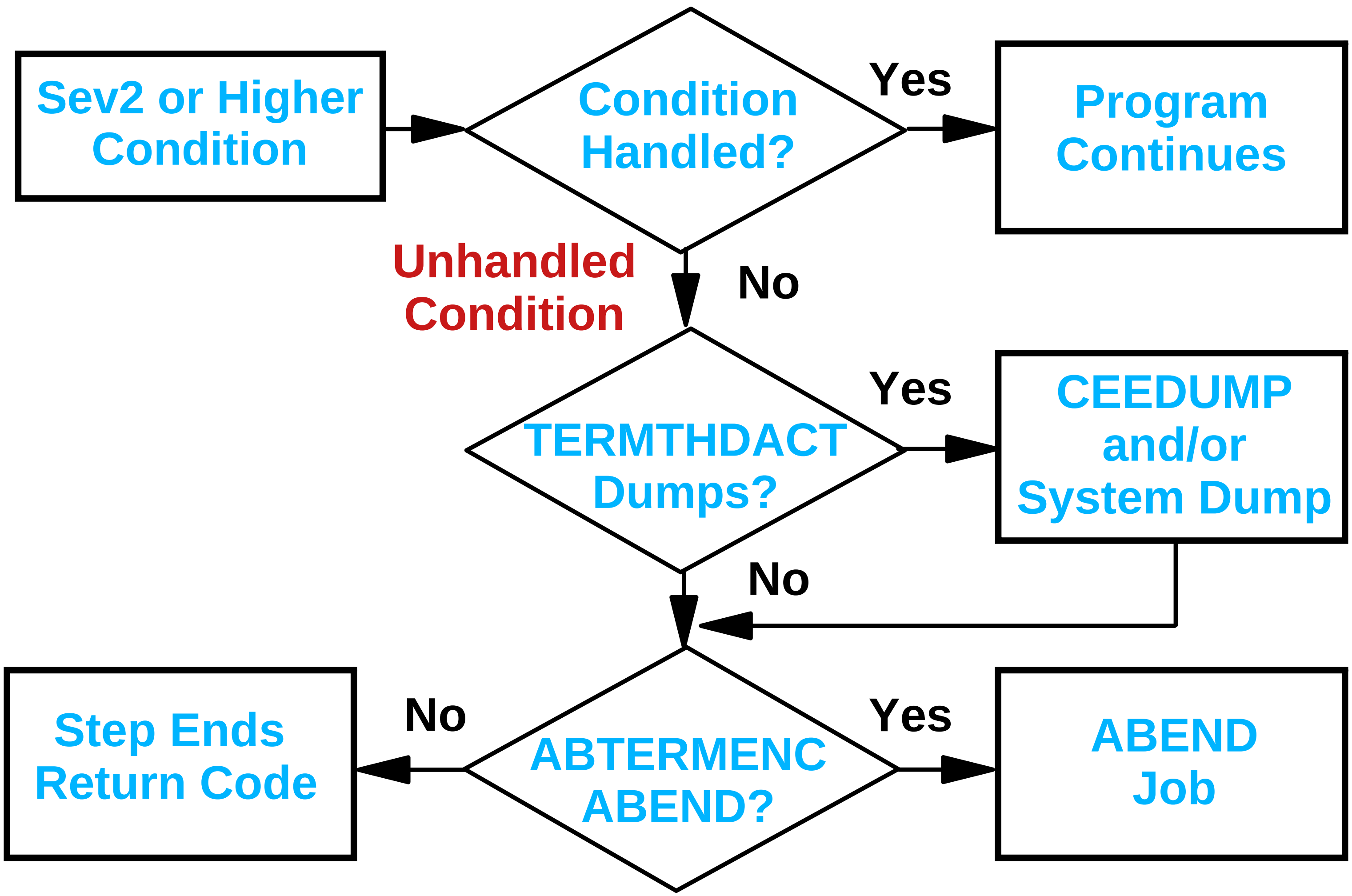
- Hardware detected interrupt, program check
- Operating system abends
- Software generated signals

Condition handling model

- Consistent and predictable condition handling
- Honors each language's error handling semantics
- Flexibility to respond directly to conditions by providing callable services to signal conditions
- Tailor condition handling with a specific routine

Condition Handling Flow

IBM Washington
Systems Center



Condition Handling Options

IBM Washington Systems Center

ABTERMENC(RETCODE | <u>ABEND</u>)

Termination behavior with an unhandled condition

- **RETCODE - to terminate with return code and reason code**
- **ABEND - to terminate with an abend**
- **Requires TRAP(ON)**

TRAP (<u>ON</u> | OFF,NOSPIE | <u>SPIE</u>)

How abends and program interrupts are handled

- **ON - enables Language Environment condition handlers**
- **OFF - condition handlers are not notified**
- **SPIE - use ESPIE macro to handle program interrupts**
- **NOSPIE - use ESTAE macro to handle program interrupts**

TERMTHDACT Options

**TERMTHDACT (MSG|QUIET|<u>TRACE|</u>DUMP|UADUMP|
UAONLY|UATRACE|UAIMM,,nnn)**

Sets level of information provided for an error condition

- **MSG** - **generates a message indicating cause of termination**
- **QUIET** - **suppresses the message**
- **TRACE** - **provides message and trace of active routines**
- **DUMP** - **provides message, trace, and LE dump**
- **UADUMP** - **provides message, trace, LE dump, and system dump to the SYSUDUMP, SYSMDUMP, or SYSABEND DD**
- **UAONLY** - **provides system dump**
- **UATRACE** - **provides message, trace, and system dump**
- **UAIMM** - **provides system dump of the original abend or program interruption & requires TRAP(ON,NOSPIE)**
- **nnn** - **amount of storage dumped around each register in CEEDUMP**

Summary

z/OS Language Environment element

Specifying run-time options

Storage model and run-time options

Condition handling model and
run-time options

Publications

IBM Washington Systems Center

LE Product Information - z/OS

- z/OS LE Concepts Guide, SA22-7567
- z/OS LE Customization, SA22-7564
- z/OS LE Programming Guide, SA22-7561
- z/OS LE Programming Reference, SA22-7562
- z/OS LE Vendor Interfaces, SA22-7568
- z/OS LE Run-Time Migration Guide, GA22-7565
- z/OS LE Debugging Guide, GA22-7560
- z/OS LE Programming Guide for 64-bit Addressing Mode, SA22-7569

LE Website

www.ibm.com/servers/eserver/zseries/zos/le/