z/OS Version 1 Release 2

z/OS 64-bit Virtual Support

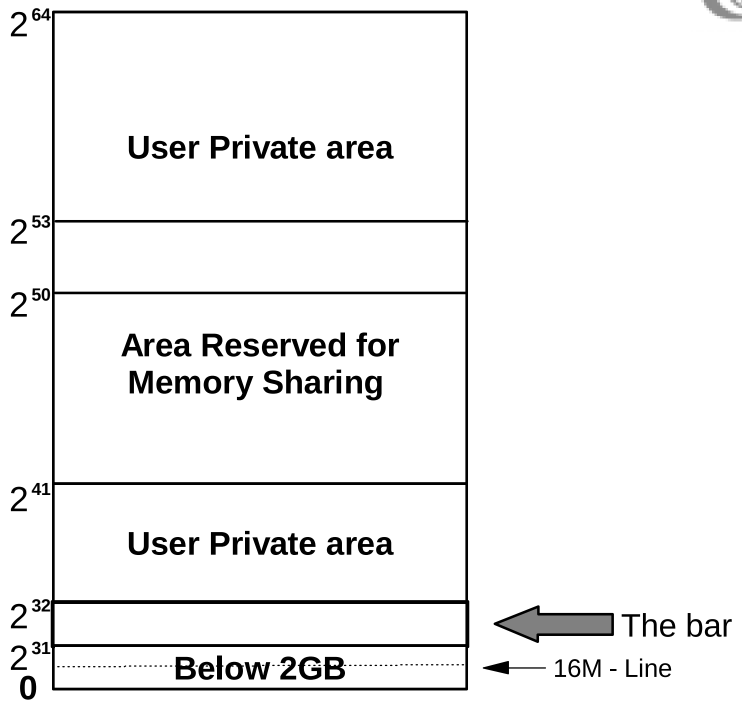
64-bit Virtual Support

- z/OS V1.2 delivers the initial basic 64-bit virtual storage management support
  - 64-bit data addressability within address spaces
  - Ability to store and manipulate data above 2 GB
- 64-bit virtual support
  - New instructions
  - New macros to obtain storage above 2 GB

64-bit Virtual Support

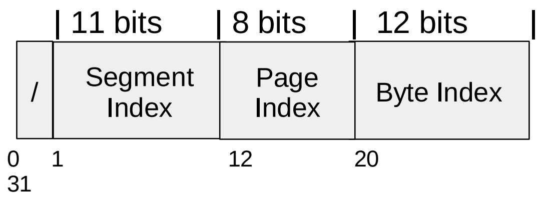
- Support for the e-business growth
  - Large number of users
    - Tremendous capacity demand for subsystem and sophisticated application servers
  - Applications and middleware lean toward a simple vertical growth programming model
    - C/C++ and JAVA programming languages
    - Larger address spaces
  - Great demand on data caching for performance enhancement

64-bit Virtual address Space

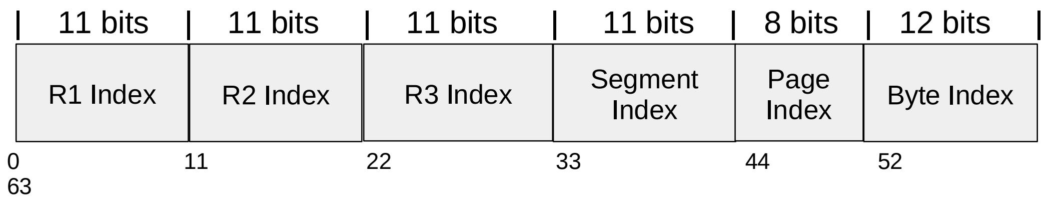


Virtual Address Formats

31-bit Virtual Address



64-bit Virtual Address



64-bit Virtual Support

- First Step z/OS Version 1 Release 2
  - z/OS assembler with support for 64 bit addressing
  - z/OS system support for 64-bit data addressability within a single address space
  - z/OS assembler system service to manage virtual storage above the bar within a single address space

64-bit HLL Support

- First Step C/C++ program execution environment
  - C/C++ Compiler
  - LE C/C++ run time library
  - UNIX System Services(syscall layer, file system, shell/utilities, shmat, mmap)
  - TCP/IP and file systems support for 64-bit data
  - dbx debugger
  - A selected small number of z/OS system services 64-bit APIs

Size and Number Notation

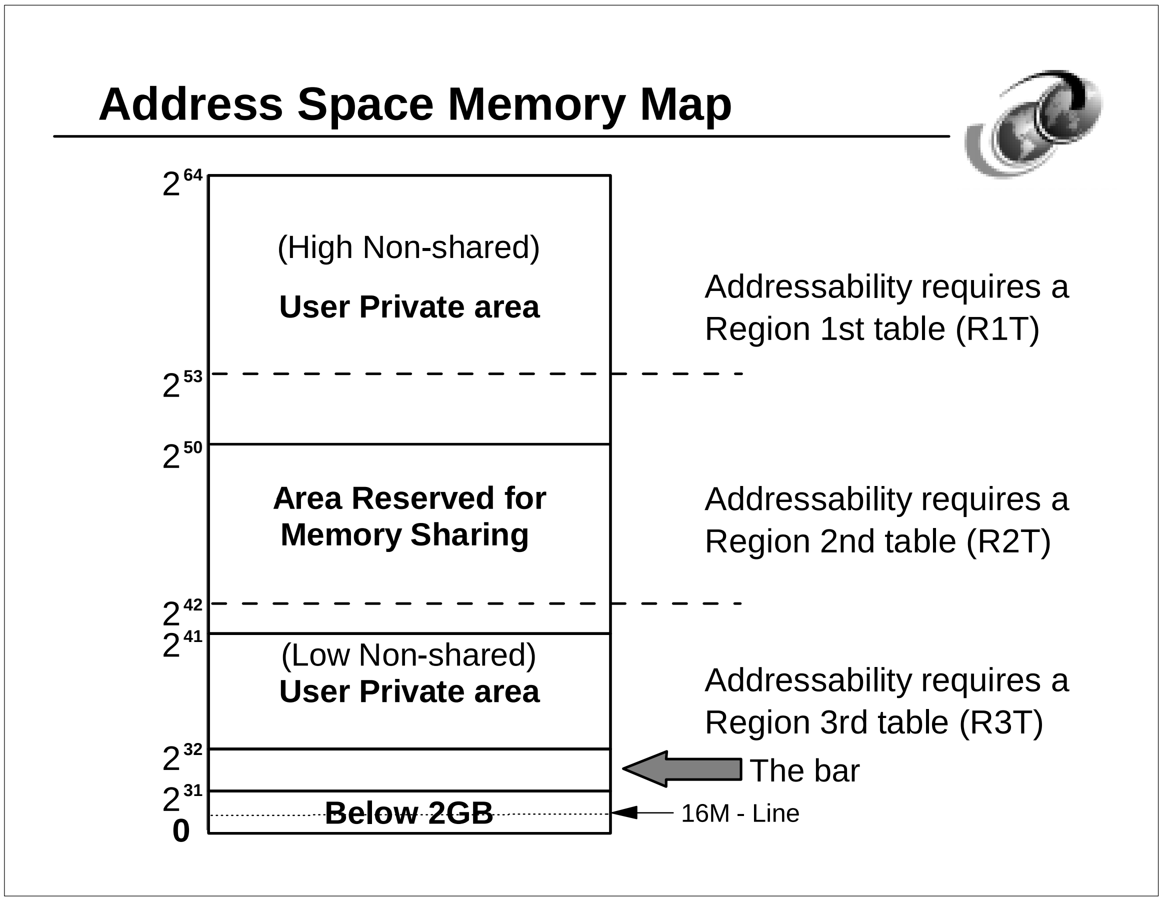
Symbol	Decimal value	Power of 2
K (kilo)	1,024	2**10
M (mega)	1,048,576	2**20
G (giga)	1,073,741,824	2**30
T (tera)	1,099,511,627,776	2**40
P (peta)	1,125,899,906,842,624	2**50
E (exa)	1,152,921,504,606,846,976	2**60

Examples

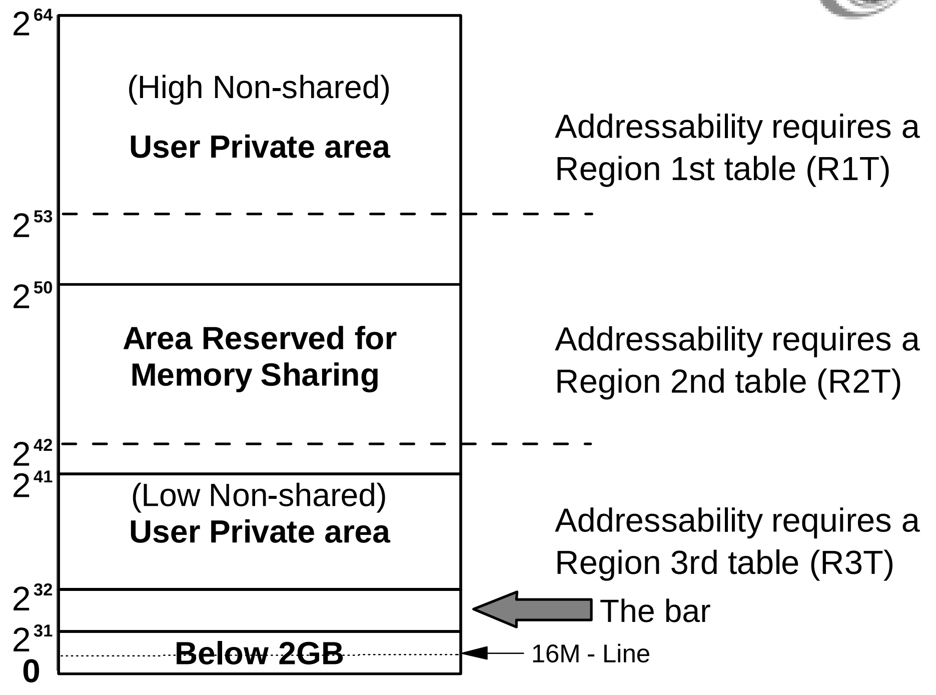
2,048	can be expressed as 2K.
4,096	can be expressed as 4K.
65,536	can be expressed as 64K.
2**24	can be expressed as 16M.
2**31	can be expressed as 2G.
2**43	can be expressed as 8T.
2**64	can be expressed as 16E.

64-bit Address Space

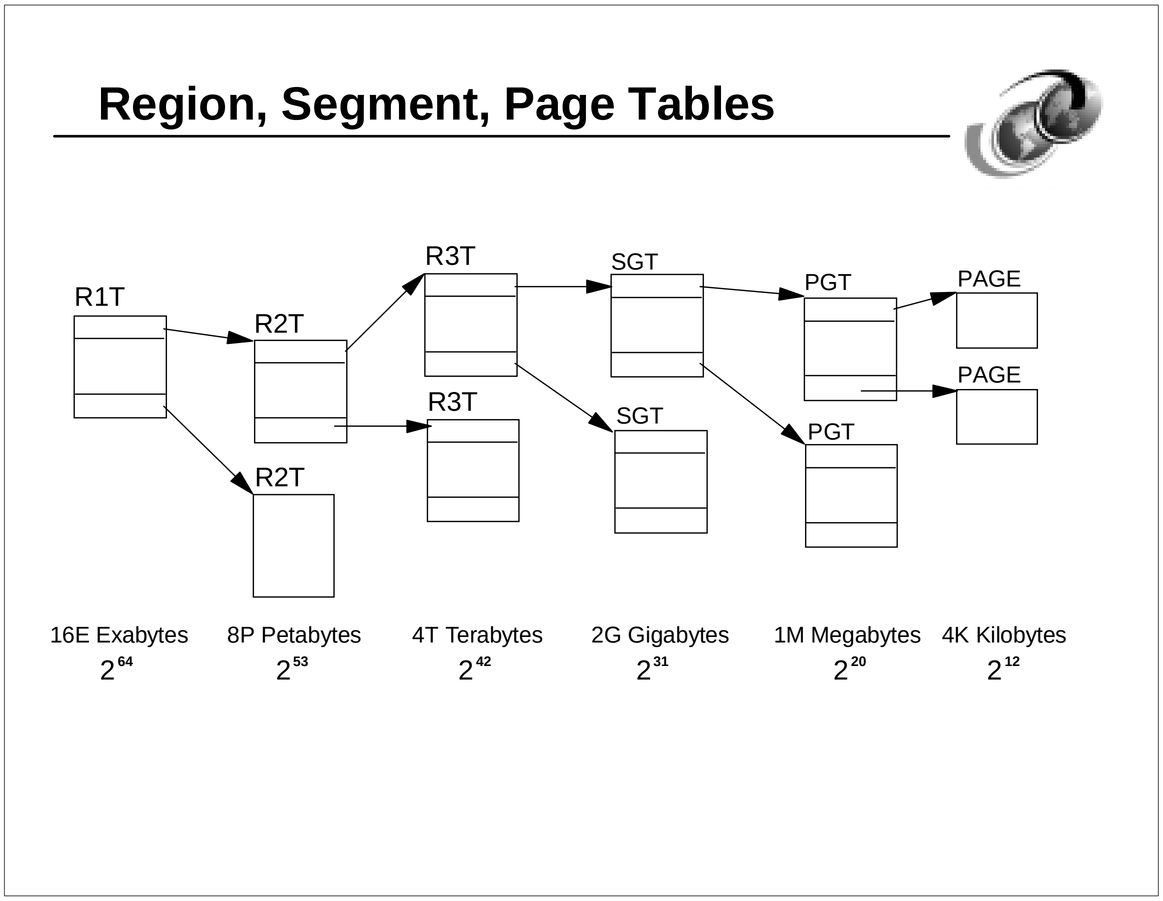
- Each address space is logically 16 exabytes
  - 2^{64} in size
- The area below 2 GB is mapped as before
  - Totally compatible with previous releases
- The area above 2 GB is for application data
  - No common areas, system areas, or programs
- An area is reserved for memory sharing
  - Available in a future release



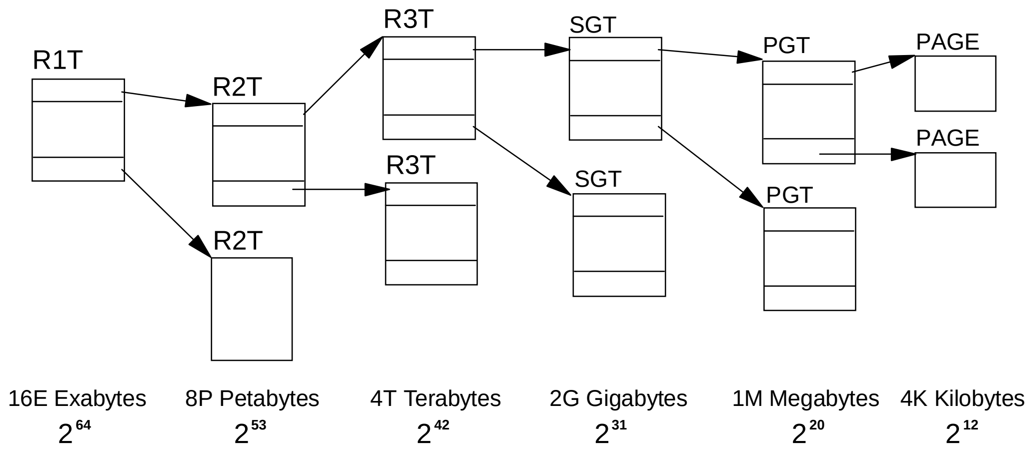
Address Space Memory Map



2^{64}
(High Non-shared)
User Private area
Addressability requires a Region 1st table (R1T)
2^{53}
2^{50}
Area Reserved for Memory Sharing
Addressability requires a Region 2nd table (R2T)
2^{42}
2^{41}
(Low Non-shared)
User Private area
Addressability requires a Region 3rd table (R3T)
2^{32}
The bar
2^{31}
Below 2GB
16M - Line
0



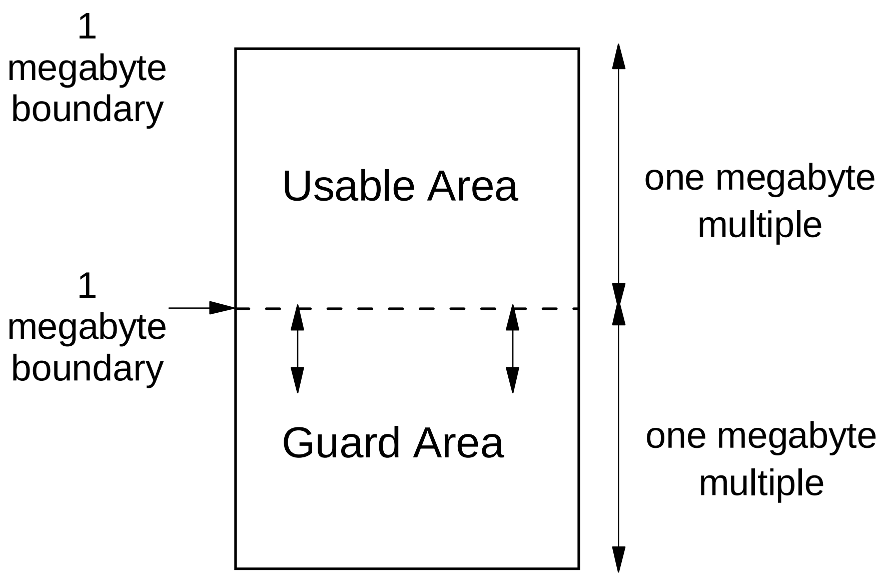
Region, Segment, Page Tables



Memory Objects

- z/OS virtual memory above 2GB is organized as
  - Memory objects
- Memory objects are a contiguous range of virtual addresses created by a program
  - Allocated as a number of 1 MB chunks of storage starting on a 1MB boundary
  - Some of the memory is usable virtual storage.
  - Remainder is not valid and is called the guard area (can be zero)
  - the extent of the usable virtual can be changed, with a compensitory change in the extent of the guard area.

Memory Objects



Using Virtual above 2 GB with V1.2

- z/OS 1.2 sets a new bit in the CVT
  - CVTV64 - when on, indicates 64-bit virtual support is present
- New z/OS High Level Assembler
  - New z/Architecture instructions for manipulating 64-bit registers and addresses
- New Assembler macro instructions to allocate and manipulate virtual storage above 2 GB
- To reference storage above 2G, a program must switch into 64-bit addressing mode (AMODE 64)

Virtual Storage Support Plan

- First Step z/OS V1.R2
  - z/OS assembler with support for 64 bit addressing
  - z/OS system support for 64-bit data addressability within a single address space
  - z/OS assembler system service to manage virtual storage above the bar within a single address space
- Next Step AMODE(64)
  - binder, loader and content supervisor
  - AMODE 64 program execution below 2GB
- Next Step Shared Support
  - z/OS system support for 64-bit data addressability between multiple address spaces
  - z/OS assembler system service to manage virtual storage above the bar between multiple address spaces

Controlling Virtual usage

- An installation wants to limit the maximum physical memory resources (real and auxilliary) that can be committed by a job
- For virtual below 2GB, a limit on virtual storage usage provides (indirectly) a way to limit real and auxilliary storage use by a job
  - The REGION= keyword on JCL and can be overridden by the IEFUSI installation exit

Virtual above the Bar

- No practical limit to the amount of virtual address range that an address space can request
- Provide a limit on the amount of usable virtual storage above 2GB that an address space can use at any one time
- The limit is 0 unless specified through either:
  - The new SMF MEMLIMIT parameter, or
  - The new MEMLIMIT keyword on JCL, and
  - Can be overridden by an IEFUSI exit

Using Virtual above 2GB

```
*  CHANGE TO AMODE 64
         SAM64
*  GET VIRTUAL STORAGE ABOVE THE BAR
         IARV64 REQUEST=GETSTOR,                                       C
               SEGMENTS=MO_SIZE,                                       C
               USERTKN=U_TOKEN,                                        C
               ORIGIN=V64_ADDR
         LTGR  15,15                  GOT MEMORY OBJECT ?
         BC    8,WG                   - YES, OK
         DC    H'0'                   - NO, INVESTIGATE
*  START WORK WITH DATA IN STORAGE ABOVE THE BAR
WG       WTO   'GOT V64',ROUTCDE=11
         LG    4,V64_ADDR             GET ADDRESS OF MEMORY OBJECT
         LHI   2,256*4                LOOP COUNTER, TOUCH ALL PAGES
TOUCH    MVC   0(L'DATA,4),DATA       MOVE IN SOME DATA
         AHI   4,4096                 TO NEXT PAGE
         BRCT  2,TOUCH                LOOP BACK AND TOUCH NEXT PAGE
*  DETACH VIRTUAL STORAGE ABOVE THE BAR
         IARV64 REQUEST=DETACH,                                        C
               MATCH=USERTOKEN,                                        C
               USERTKN=U_TOKEN,                                        C
               COND=YES
         LTGR  15,15                  FREED MEMORY OBJECT ?
         BC    8,WD                   - YES, OK
         DC    H'0'                   - NO, INVESTIGATE
WD       WTO   'DETACHED V64',ROUTCDE=11
```

Data Area for Obtaining Storage

```
*  END EXIT LINKAGE
@DATA    DS    0D
MO_SIZE  DC    FD'4'                  MEMORY OBJECT IS 4 MB
U_TOKEN  DC    FD'1'
DATA     DC    C'DATA ABOVE THE BAR'
```

Addressing Mode Switching

- There are 3 new instructions which change addressing mode without branching:
  - Set Addressing Mode to 24-bit (SAM24)
  - Set Addressing Mode to 31-bit (SAM31)
  - Set Addressing Mode to 64-bit (SAM64)
- There are 2 instructions which change addressing mode and branch:
  - Branch and Save and Set Mode (BASSM)
  - Branch and Set Mode (BSM)

Controlling storage - MEMLIMIT

- Through JCL on the specific job with the new
  - MEMLIMIT JCL keyword
- MEMLIMIT specified on a JOB statement
  - //TC1 JOB MEMLIMIT=50G,REGION=0M
  - //TC2 JOB MEMLIMIT=125M,TIME=NOLIMIT
  - //TC3 JOB MEMLIMIT= 9T,MSGLEVEL=1
  - //TC4 JOB REGION=3M,MEMLIMIT=16384P
  - //TC5 JOB REGION=125M,MSGLEVEL=(1,1),
  - MEMLIMIT=NOLIMIT,MSGCLASS=A
- MEMLIMIT specified on an EXEC statement
  - //STEP1 EXEC PGM=TST6,MEMLIMIT=6400M
  - //STEP2 EXEC PGM=TST7,MEMLIMIT=3P...
  - //STEP3 EXEC MYPROC,MEMLIMIT=NOLIMIT...

Controlling storage - SMFPRMxx

```
ACTIVE                       /*ACTIVE SMF RECORDING*/
DSNAME ( SYS1.MANA,SYS1.MANB,SYS1.MANC) /* NEW D.S. ADDED 11/88 */
PROMPT(LIST)                 /*PROMPT THE OPERATOR FOR OPTIONS*/
REC(PERM)                    /*TYPE 17 PERM RECORDS ONLY*/
BUFNUM(4,9)                  /* 4 - 4096 BUFFERS ALWAYS AND
                                ALLOW UP TO 9 BEFORE SUSPENDING
                                A USER FOR BUFFER SHORTAGE*/
MAXDORM(3000)               /* WRITE AN IDLE BUFFER AFTER 30 MIN*/
MEMLIMIT(24G)
STATUS(010000)              /* WRITE SMF STATS AFTER 1 HOUR*/
JWT(1439)                   /* NO 522 ABENDS*/
SID(168A)                   /* SYSTEM ID IS 168 A*/
LISTDSN                     /* LIST DATA SET STATUS AT IPL*/
SYS(TYPE(0:255),EXITS(IEFACTRT,IEFUJV,IEFUSI,IEFU83,
                      IEFUJI,IEFUTL,IEFU29),NOINTERVAL,NODETAIL)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
MEMLIMIT(16384P)           /* This is the same as NOLIMIT */
MEMLIMIT(125T)

MEMLIMIT(4000P)

MEMLIMIT(0M)                 /* Disallow  storage >2G */

MEMLIMIT(00000M)         /*   DEFAULT   */
```

MEMLIMIT During the IPL

```
    SYS(TYPE(0:255)) -- DEFAULT
          LISTDSN -- DEFAULT
          SID(4381) -- DEFAULT
          STATUS(010000) -- DEFAULT
          MAXDORM(3000) -- DEFAULT
          DDCONS(YES) -- DEFAULT
          LASTDS(MSG) -- DEFAULT
          NOBUFFS(MSG) -- DEFAULT
          SYNCVAL(00) -- DEFAULT
          INTVAL(30) -- DEFAULT
          DUMPABND(RETRY) -- DEFAULT
          REC(PERM) -- DEFAULT
          DSNAME(SYS1.MANY) -- DEFAULT
          DSNAME(SYS1.MANX) -- DEFAULT
          MEMLIMIT(NOLIMIT) -- PARMLIB
          JWT(1439) -- PARMLIB
          PROMPT(ALL) -- PARMLIB
          NOACTIVE -- PARMLIB
 *01 IEE357A REPLY WITH SMF VALUES OR U
00- r 1,MEMLIMIT(2G)
    IEE600I REPLY TO 01 IS;MEMLIMIT(2G)
   *02 IEE357A REPLY WITH SMF VALUES OR U
```

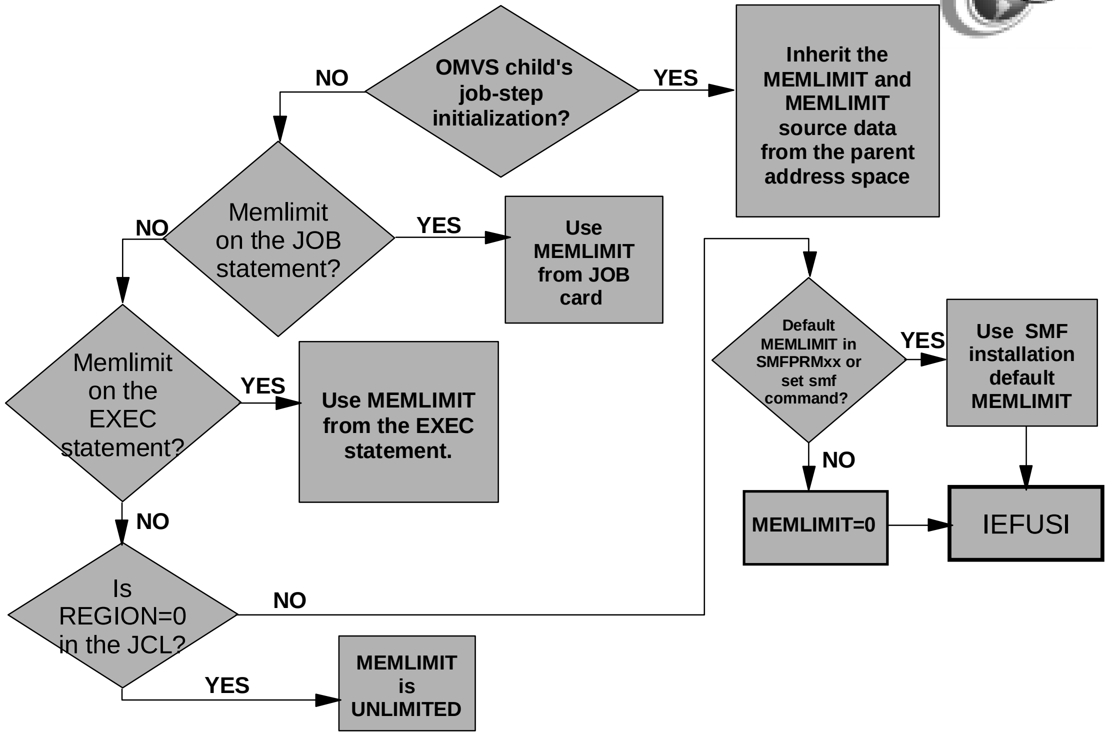
Reset the SMF Parameters

```
SET SMF=M4
IEE252I MEMBER SMFPRMM4 FOUND IN
RSMID.PARMLIB
IEE536I SMF      VALUE M4 NOW IN EFFECT
D SMF,O
IEE967I 00.56.34 SMF PARAMETERS 379
MEMBER = SMFPRMM4
DSNAME(SYS1.MANY) -- DEFAULT
DSNAME(SYS1.MANX) -- DEFAULT
ACTIVE -- DEFAULT
MEMLIMIT(00003G) -- PARMLIB
JWT(2400) -- PARMLIB
PROMPT(ALL) -- PARMLIB
```

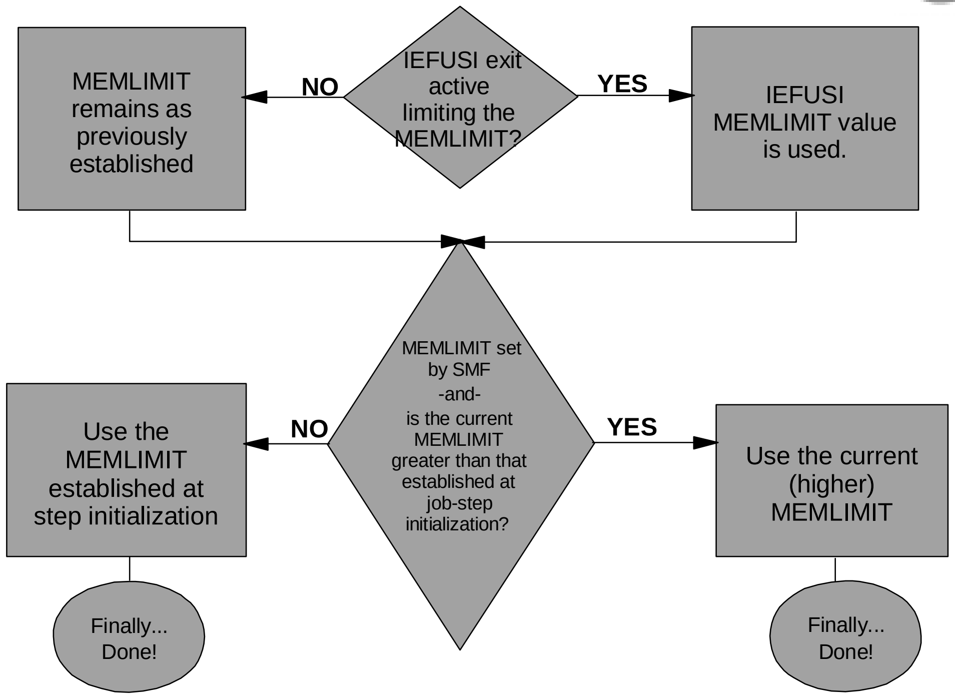
Change MEMLIMIT Value

```
 setsmf memlimit(120t)
  IEE712I SETSMF PROCESSING COMPLETE
 d smf,o
IEE967I 01.29.56 SMF PARAMETERS
       MEMBER = SMFPRMBR
       MEMLIMIT(00120T) -- REPLY
       PROMPT(ALL) -- PARMLIB
       DDCONS(YES) -- DEFAULT
       LASTDS(MSG) -- DEFAULT
       NOBUFFS(MSG) -- DEFAULT
       SYNCVAL(00) -- DEFAULT
       INTVAL(30) -- DEFAULT
       DUMPABND(RETRY) -- DEFAULT
```

Determine MEMLIMIT for Jobstep


OMVS child's job-step initialization?
YES
Inherit the MEMLIMIT and MEMLIMIT source data from the parent address space
NO
Memlimit on the JOB statement?
YES
Use MEMLIMIT from JOB card
NO
Memlimit on the EXEC statement?
YES
Use MEMLIMIT from the EXEC statement.
NO
Is REGION=0 in the JCL?
YES
MEMLIMIT is UNLIMITED
NO
Default MEMLIMIT in SMFPRMxx or set smf command?
YES
Use SMF installation default MEMLIMIT
NO
MEMLIMIT=0
IEFUSI

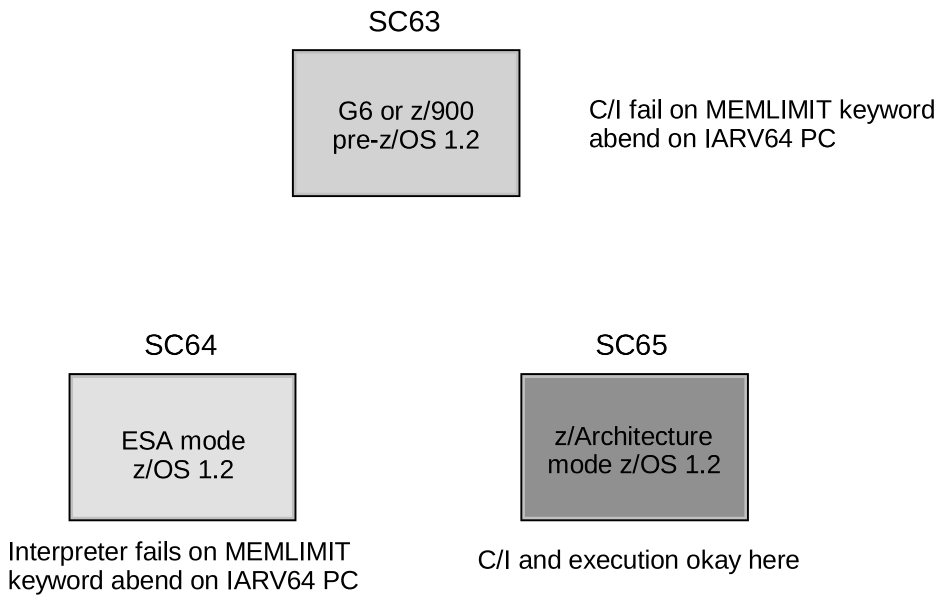
IEFUSI Exit


IEFUSI exit active limiting the MEMLIMIT?
NO
MEMLIMIT remains as previously established
YES
IEFUSI MEMLIMIT value is used.
MEMLIMIT set by SMF -and- is the current MEMLIMIT greater than that established at job-step initialization?
NO
Use the MEMLIMIT established at step initialization
YES
Use the current (higher) MEMLIMIT
Finally... Done!
Finally... Done!

MEMLIMIT - SMF Type 30

- MEMLIMIT and source of MEMLIMIT are recorded in the SMF type 30 record in the Storage and Paging section
  - Bit 2 of SMF30SFL is set when IEFUSI changes the MEMLIMIT value
  - New doubleword field SMF30MEM at offset X'A8' is the MEMLIMIT value used
  - New byte field SMF30MLS at offset X'B0' indicates the MEMLIMIT source

JES2 Sysplex Considerations

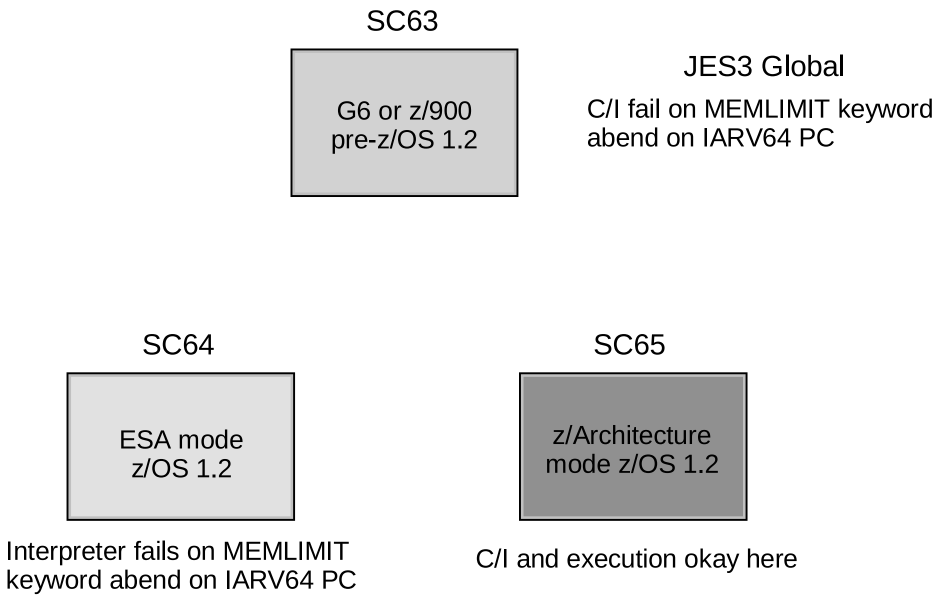

SC63
G6 or z/900 pre-z/OS 1.2
C/I fail on MEMLIMIT keyword abend on IARV64 PC
SC64
ESA mode z/OS 1.2
Interpreter fails on MEMLIMIT keyword abend on IARV64 PC
SC65
z/Architecture mode z/OS 1.2
C/I and execution okay here

MEMLIMIT: Sysplex Considerations

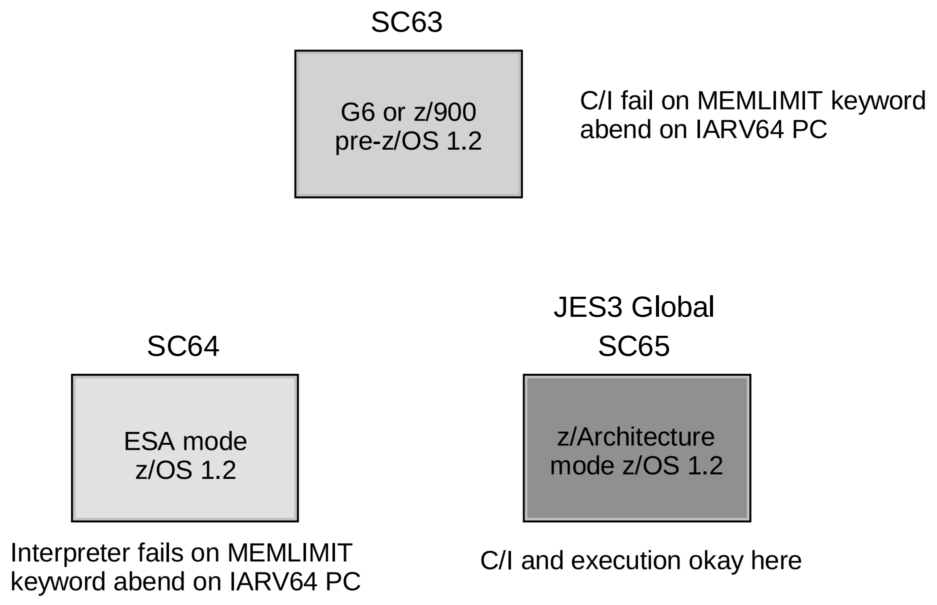
JES2 SYSTEMS

Converted on	**Interpreted and executed on**	**Results**	**System Programmer Response**
SC63	never gets this far	IEFC630I - fails in Converter	Code SYSAFF=SC65
SC64 or SC65	SC63	IEF630I - fails in interpreter	Code SYSAFF=SC65 or SCHENV=
SC64 or SC65	SC64	IEF897I - fails in interpreter	Code SYSAFF=SC65 or SCHENV=
SC64 or SC65	SC65	job runs ok	-

JES3 Sysplex Considerations


SC63
G6 or z/900 pre-z/OS 1.2
JES3 Global
C/I fail on MEMLIMIT keyword abend on IARV64 PC
SC64
ESA mode z/OS 1.2
Interpreter fails on MEMLIMIT keyword abend on IARV64 PC
SC65
z/Architecture mode z/OS 1.2
C/I and execution okay here

JES3 Sysplex Considerations

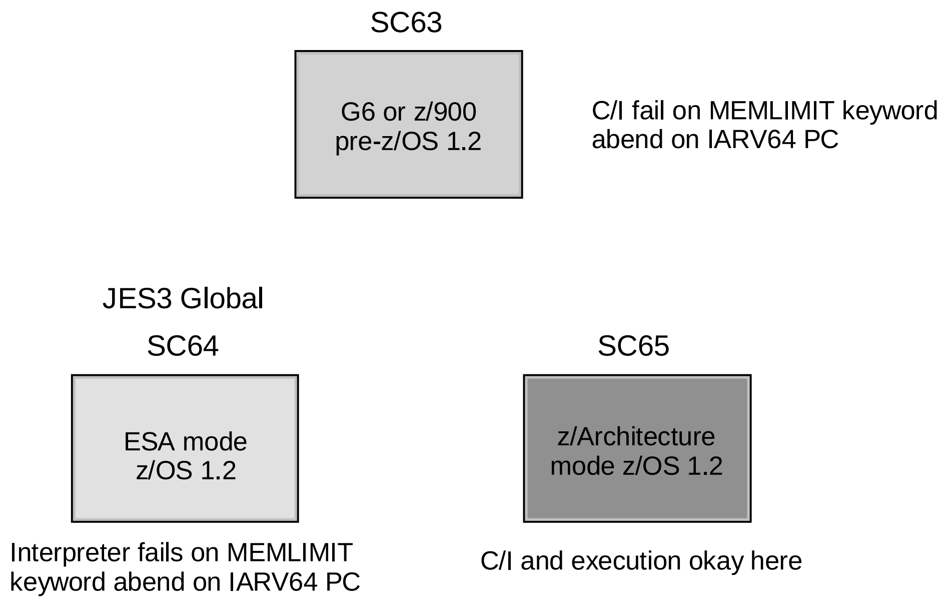


C/I fail on MEMLIMIT keyword abend on IARV64 PC

Interpreter fails on MEMLIMIT keyword abend on IARV64 PC

C/I and execution okay here

JES3 Sysplex Considerations



C/I fail on MEMLIMIT keyword abend on IARV64 PC

Interpreter fails on MEMLIMIT keyword abend on IARV64 PC

C/I and execution okay here

MEMLIMIT: Sysplex Considerations

JES3 SYSTEMS

Converted and Interpreted on	Executed on	Results	System Programmer Response
SC63 ------------------------- SC64	never gets this far	IEFC630I - fails in Converter --------------------------- IEF897I - fails in Interpreter	code an FSSDEF,TYPE =CI,..SYSTEM= SC65... statement in the JES3 INIT DECK
SC65	SC63, SC64	Job runs - application will abend - 0D6 on IARV64 PC **	//*MAIN SYSTEM = SC65 Assign job to a job class enabled on only SC65 //JOB...SCHENV = PROD01... defined in WLM policy)
SC65	SC65	job runs ok	--

Dumping Virtual above 2 GB

- SVC Dump service has been enhanced for 64-bit support (SDUMPX macro)
- In the future, binary dumps taken to SYSMDUMP data sets will be enhanced for 64-bit support
- The dumps taken to SYSABEND and SYSUDUMP data sets have not been enhanced for 64-bit support

Using SDUMPX for Virtual above 2 GB

- The SDUMPX macro can be invoked in 64-bit addressing mode.
- The data in memory objects which were created with SVCDUMPRGN=YES is included in the dump when SDATA=RGN is requested
- You can specify a list of 64-bit address ranges to be included in the dump. The list itself, however, must reside below 2GB. [LIST64=]

Functions not supported - 64-bit Virtual

- Dataspaces
- Hiperspaces
- Subspace capability mutually exclusive with high virtual capability in an address space
- DIV
- IARVSERV (Copy-on-Write, ChangeAccess)
- Change Key
- Checkpoint Restart

Middleware Support - Later Release

- Exploitation of 64 Bit Virtual by the Middleware
  - DB2 - enhance database buffer management support to provide continued growth in the large system transaction environment
  - Websphere
    - 64 bit server regions
    - Support very large objects and a very large numbers of objects
    - Provide relief for 2GB address space limit
- Support of 64 Bit Virtual Application by the Middleware
  - DB2
    - Support of both 31 and 64 bit applications