

Unit 11d – Prepare your Operating System for IMS

IMS runs exclusively in a z/OS and requires services in order to execute. There are a number of activities that need to be performed in order to permit IMS to run in a z/OS LPAR. Some of these activities only need to be performed once for a z/OS system; other of these activities might need to be performed each time a new version of IMS is installed. This chapter covers a few information about IMS IVP example steps (that demonstrate key z/OS-related activities), describes alternative procedures for defining IMS SVCs to MVS, identifies which IMS modules need to be in LPALIB, Which steps are involved for specifying IMS-VTAM definitions .

Some steps related to preparing z/OS for IMS

Help

Execution (LST Mode) - DBT

Row 14 to 28 of 261

Action Codes: Brm Doc Edm eNt eXe Ftl spR

JOB/Task	Step	Title.....
IV3D001T	D0	NOTE - Introduction - z/OS and VTAM Interface
IV3D101T	D1	XMPL - Allocate Interface Data Sets
IV3D200T	D2	XMPL - Update JESx Procedure
IV3D201T	D2	XMPL - Update IEAAPFxx or PROGxx - Authorized DSN
IV3D202T	D2	XMPL - Update IEALPAXx - MLPA Modules
IV3D206T	D2	XMPL - Update IEFSSNxx - RLM Subsystem Names
IV3D207T	D2	XMPL - Update IEASVCxx - SVC Numbers
IV3D209T	D2	XMPL - Install TYPE 2 SVC
IV3D210T	D2	XMPL - Link-edit TYPE 4 SVC
IV3D212T	D2	XMPL - Link-edit Abend Formatting Module
IV3D214T	D2	XMPL - Add DFSAFMD0 to IEAVADFM CSECT of IGC0805A
IV3D215T	D2	XMPL - Update BLSCECTX IPCS Exits
IV3D216T	D2	XMPL - IPCS ISPF Dataset Concatenation
IV3D217T	D2	XMPL - Define z/OS Dump Options
IV3D218T	D2	XMPL - Define RACF Security Profile

Command ==>

Scroll ==> CSR

F1=Help

F3=Exit

F7=Backward

F8=Forward

F10=Actions



KC110 unit 11d page 2

The IVP D Series steps (IV3D101 through IV3D401) are where the z/OS interface activities are illustrated. The z/OS interfaces are illustrated in the IV3D2nnT steps; the VTAM interfaces steps are shown in the IV3D3nnT steps. We will discuss only a few of these steps.

JOB/Task	Step	Title.....
IV3D301T	D3	XMPL - Define VTAM Application Nodes
IV3D302T	D3	XMPL - Define VTAM Network Nodes
IV3D303T	D3	XMPL - Define VTAM Logon Mode Tables
IV3D304T	D3	XMPL - Define VTAM Interpret Tables
IV3D305T	D3	XMPL - Define VTAM USS Definition Tables
IV3D306T	D3	XMPL - Define VTAM Configuration List (ATCCONxx)
IV3D307T	D3	XMPL - Define VTAM Start Option List (ATCSTRxx)
IV3D308T	D3	XMPL - Copy VTAM Procedure to SYS1.PROCLIB
IV3D401T	D4	TASK - IPL z/OS with MLPA or CLPA Option

IV3D201T: PROCLIB definition in SYS1.PROCLIB member JES2

```
//JES2      PROC M=JES2PARM,MC=JES2COMM,MN=JESWMNEW,
//          PDSN='SYS1.TSO.PROCLIB'
//IEFPROC EXEC PGM=HASJES20,DPRTY=(15,15),TIME=1440,PERFORM=9
//*
//* JES2 JCL FOLLOWS
//*
//PROC00     DD DSN=&PDSN,DISP=SHR
//           DD DSN=SYS1.PROCLIB,DISP=SHR
//           DD DSN=CICS.PROCLIB,DISP=SHR
//           DD DSN=DB2.PROCLIB,DISP=SHR
//           DD DSN=AS.PROCLIB,DISP=SHR
//           DD DSN=MQS.PROCLIB,DISP=SHR
//           DD DSN=IMS.STC.PROCLIB,DISP=SHR          IMS or common STARTED TASK LIBRARY
//           DD DSN=SYS1.JES.PROCLIB,DISP=SHR DD
//           DSN=SYS1.WEBSRV.PROCLIB,DISP=SHR DD
//PROC01     DD DSN=&PDSN,DISP=SHR
//           DD DSN=SYS1.PROCLIB,DISP=SHR
//PROC02     DD DSN=IMS.DBC1.PROCLIB,DISP=SHR
//           DD DSN=SYS1.PROCLIB,DISP=SHR
//PROC03     DD DSN=IMS.DBC.PROCLIB,DISP=SHR
//           DD DSN=SYS1.PROCLIB,DISP=SHR
//PROC04     DD DSN=IMS.PROCLIB,DISP=SHR
//           DD DSN=SYS1.PROCLIB,DISP=SHR
//PROC05     DD DSN=IMS.DBT1.PROCLIB,DISP=SHR
```

Here is an activity that only needs to be performed once. At least some IMS procedures must be in the JES proc concatenation so that IMS can be started using the z/OS START (for example, S IMS.xx) command; at least the IMS, DLISAS, DBRC and IMSRDR procedure should be in the JES concatenation. If security considerations permit, it is more convenient for us to have the entire contents of IMS.PROCLIB as we are demonstrating here through the use of the IMS.STC.PROCLIB data set.

However, in general most reference to the IMS.PROCLIB can be done in jobs with the JCL statement as shown below:

```
//IMSPROCS      JCLLIB      ORDER=(IMS.PROCLIB,.....)
```

IV3D203T: Update PROGxx – Authorized data sets

Contents of member PROGxx in SYS1.PARMLIB

```
.  
.  
APF FORMAT (DYNAMIC)  
APF ADD DSNAME (SYS1.PPLINK) VOLUME (PROD00)  
.  
.  
.  
.  
.  
APF ADD DSNAME \(IMS.SDFSRESL\) VOLUME \(DBDC02\)  
APF ADD DSNAME \(IMS.SDXRRESL\) VOLUME \(DBDC02\)  
APF ADD DSNAME \(IMS.MODBLKSA\) VOLUME \(DBDC02\)  
APF ADD DSNAME \(IMS.MODBLKSB\) VOLUME \(DBDC02\)  
.
```

This activity might need to be performed repeatedly. Note that IMS requires certain authorized libraries .

These include libraries that contain IMS executable load modules (for example, IMS.SDFSRESL above) plus other libraries, such as MODBLKS (A&B) and DYNLIBs / DYNALLOCs.

IV3D207T: Update IEASVCxx – SVC numbers

```
EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE
EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE
EXAMPLE--
EXAMPLE--          NOT INTENDED FOR USER EXECUTION          --EXAMPLE
EXAMPLE--
EXAMPLE--          --EXAMPLE
EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE
EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE--EXAMPLE
/*
.
.
.
SVC Parm 254 REPLACE,          /* IMS TYPE 2 SVC */
      TYPE(2)
SVC Parm 255 REPLACE,          /* IMS TYPE 4 SVC FOR DBRC */
      TYPE(4)
.
.
.
```

Here we have the SYS1.PARMLIB member that assigns SVCs. IMS uses two (downward-compatible) SVC's: one Type2 and one Type4 SVC (DBRC) . Depending on how your installation manages SVC numbers, you could be required to have this PARMLIB member updated each time you upgrade to a new IMS version.

There is an SVC update utility (since IMS V12) – in case you want to use same SVC numbers and not waiting for an IPL.
(SVCs are downwards compatible)

IV3D208T: Update SCHEDxx – PPT entries

```

      .
      .
      .
PPT PGMNAME(DXRRLM00) /* IRLM - RESOURCE LOCK MANAGER */
                        /* BITS WERE '6870FFFF00000000' */
      CANCEL           /* PROGRAM CAN BE CANCELLED (DEFAULT) */
      KEY(7)           /* PROTECT KEY ASSIGNED IS 7 */
      NOSWAP           /* PROGRAM IS NOT-SWAPPABLE */
      NOPRIV           /* PROGRAM NOT PRIVILEGED (DEFAULT) */
      DSI              /* DOES NOT REQUIRE DATA SET INTEGRITY (DFLT) */
      SYST             /* PROGRAM IS A SYSTEM TASK */
      NOPASS           /* CAN BYPASS PASSWORD PROTECTION */
      AFF(NONE)        /* NO CPU AFFINITY (DEFAULT) */
      NOPREF           /* NO PREFERRED STORAGE FRAMES (NODEFAULT) */
      .
      .
      .
PPT PGMNAME(DFSMVRC0) /* IMS ONLINE CONTROL REGION */
                        /* BITS WERE '6870FFFF00000000' */
      CANCEL           /* PROGRAM CAN BE CANCELLED (DEFAULT) */
      KEY(7)           /* PROTECT KEY ASSIGNED IS 7 */
      NOSWAP           /* PROGRAM IS NOT-SWAPPABLE */
      NOPRIV           /* PROGRAM NOT PRIVILEGED (DEFAULT) */
      DSI              /* DOES NOT REQUIRE DATA SET INTEGRITY (DFLT) */
      SYST             /* PROGRAM IS A SYSTEM TASK */
      NOPASS           /* CAN BYPASS PASSWORD PROTECTION */
      AFF(NONE)        /* NO CPU AFFINITY (DEFAULT) */
      NOPREF           /* NO PREFERRED STORAGE FRAMES (NODEFAULT) */
      .
      .
      .
```

Notes:

This step might not be necessary if already included in MVS - z/OS as part of preconditioning. This is an activity that will only need to be performed once. Another PPT entry you will need for OTMA .

```
//STEP02      EXEC  PGM=IEWL,REGION=512K,
//              PARM='LIST,XREF,NCAL,SCTR,LET,SIZE=(1024K,128K)'
//SYSPRINT DD  SYSOUT=*
//RESLIB      DD  DISP=SHR,DSN=IMS.SDFSRESL
//SYSLMOD      DD  DSN=SYS1.NUCLEUS,DISP=SHR,UNIT=3390,VOL=SER=MRSALL
//SYSUT1       DD  UNIT=VIO,SPACE=(CYL,(3,1))
//SYSLIN       DD  *

                .
                .

OUTPUT DATA SET SYS1.NUCLEUS IS ON VOLUME MRSALL
+
+ INSERT/ORDER STATEMENTS FROM PREVIOUS z/OS NUCLEUS LINK
+
IEW0000          INSERT    IEAVNIPO
IEW0000          INSERT    IEAVFX00
IEW0000          ORDER MPL,IEAVEIO,IEAVESVC,IEAVEEXP,IEAVESLK
IEW0000          ORDER    IEAVBK00
IEW0000          ORDER    IEAVESVT
IEW0000          INCLUDE RESLIB(IGC254)                DFSVC200 (IMS TYPE 2 SVC)
IEW0000          INCLUDE SYSLMOD(IEANUC01)            OLD Z/OS NUCLEUS
IEW0000          NAME IEANUC02(R)                    NEW Z/OS NUCLEUS WITH IMS SVC
IEW0461          IECDDEDC
IEW0461          IOSRMIH3
** IEANUC02 REPLACED AND HAS AMODE 31

                .
                .
                .
```

The update to the IMS Type 2 SVC typically happens once per IMS version upgrade; it is downward compatible for use by older IMS versions running on the same LPAR. Illustrated here is one way of including the Type 2 SVC into z/OS. Once everything executes properly, IEANUC02 can be renamed IEANUC01.

Also, this IMS DFSUSVC0 Utility allows the user to replace the SVC without a re-IPL of z/OS by Loading the SVC into global storage. No IMS that uses the particular SVC might be active while utility is executed. We will discuss this utility later in this topic.

Your z/OS systems programmers (or possibly you if the use of DFSUSVC0 is permitted) will manage the implementation of the IMS SVCs based on the security/change management rules of your company.

IV3D210T: Link Edit (Bind) TYPE 4 SVC into LPALIB

```
//LINKSVC4 EXEC PGM=IEWL,  
//          PARM='LIST,LET,XREF,RENT,REFR'  
//SYSPRINT DD SYSOUT=*  
//RESLIB   DD DISP=SHR,DSN=IMS.SDFSRESL  
//SYSLMOD  DD DISP=SHR,DSN=SYS1.LPALIB.IMS      <<<<< YOUR MLPA LIBRARY  
//SYSUT1   DD UNIT=3390,SPACE=(CYL,(3,1))  
//SYSLIN   DD *
```

```
JOB IV2D210T  STEP LINKSVC4  
INVOCATION PARAMETERS - LIST,LET,XREF,RENT,REFR  
ACTUAL SIZE=(317440,86016)  
OUTPUT DATA SET SYS1.LPALIB.IMS IS ON VOLUME XASYST  
IEW0000      INCLUDE RESLIB\(IGC0025E\)          DSP00MVS DBRC TYPE 4 SVC - 255  
IEW0000      NAME IGC0025E\(R\)
```

CROSS REFERENCE TABLE

CONTROL SECTION			ENTRY
NAME	ORIGIN	LENGTH	NAME LOCATION NAME LOCATION
		.	
		.	
		.	

Notes:

Alternatively, the SVC can be implemented using the IMS DFSUSVC0 Utility if permitted. As is the case for Type 2 SVCs, the update to the IMS Type 4 SVCs typically happens once per IMS version upgrade.

IV3D212T: Link Edit (Bind)

ABEND Dump Format Module into LPALIB

```
//LINKAFMD EXEC PGM=IEWL,
//          PARM='LIST,LET,XREF,RENT,REFR'
//SYSPRINT DD SYSOUT=*
//RESLIB   DD DISP=SHR,DSN=IMS.SDFSRESL
//SYSLMOD  DD DISP=SHR,DSN=SYS1.LPALIB.IMS          <<<<< YOUR MLPA LIBRARY
//SYSUT1   DD UNIT=3390,SPACE=(CYL,(3,1))
//SYSLIN   DD *
```

```
JOB IV2D212T    STEP LINKAFMD
INVOCATION PARAMETERS - LIST,LET,XREF,RENT,REFR
ACTUAL SIZE=(317440,86016)
OUTPUT DATA SET SYS1.LPALIB.IMS IS ON VOLUME XASYST
```

```
IEW0000      INCLUDE RESLIB (DFAFMDO)          DFAFMDO
```

```
IEW0000      SETCODE AC(1)
```

```
IEW0000      NAME DFAFMDO(R)
```

ABEND DUMP FORMATTING ROUTINE

0

0

CROSS REFERENCE TABLE

0 CONTROL SECTION

ENTRY

0 NAME ORIGIN LENGTH

NAME LOCATION

NAME LOCATION

.

.

.



KISSLICH CONSULTING

KC110 unit 11d page 9

Notes:

The update to the IMS Dump Formatting Module is required once per IMS version upgrade; it is downward-compatible with older IMS versions. The update of this module (and the need to perform an IPL) can be deferred as it is only required for ONLINE Dump formatting. When the IMS FMTO= Option is set to FMTO=D (the recommended value), dump formatting can be performed later by IPCS and IPCS will not require a new copy of this module in LPALIB.

IV3D401T: IPL z/OS with MLPA or CLPA option

You might need

IPL

NUC=0x

CLPA

- If SYS1.LPALIB
or user.LPALIB included in LPALSTxx

MLPA

- To temporarily load modules included
in IEALPAxx into a LPA extension.

Notes:

Eventually an IPL will be required. Depending on how the changes we have described were performed, we might be able to defer this IPL and still use the changes we have described earlier in this unit.

IMS recent Versions Enhancements

- The following IMS Features can be used to defer the need for *upgrade* IPL of IMS:
 - Type 2 and Type 4 SVCs can be installed using the IMS DFSUSVC0 utility without an IPL
 - Run DFSUSVC0 and specify SVCTYPE= (2,4)

Notes:

IMS can be installed and maintained with fewer environmental requirements than previous releases. With an enhancement to the IMS SVC Update Utility DFSUSVC0, there will no longer be a requirement for a CLPA z/OS IPL when maintaining the DBRC Type 4 SVC into LPALIB.

IMS V14 and 15 news – Check it !

DFSUSVC0 Utility

- IMS SVC Update Utility (DFSUSVC0) ...
 - Since IMS V11 , ...
 - Continues to support dynamic updates of the TYPE 2 SVC
 - Adds support for Type 4 SVC
 - Loads either (or both) of the SVCs from the data set associated with the DFSRESLB DDNAME

```
//STEP001 EXEC PGM=DFSUSVC0,PARM='SVCTYPE=(4)'  
//STEPLIB DD DSN=IMS.SDFSRESL, DISP=SHR  
//DFSRESLB DD DSN=IMS.SDFSRESL, DISP=SHR  
//SYSPRINT DD SYSOUT=*  
// DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605),  
// SPACE=(605,(10,10),RLSE,,ROUND)
```

New Type 2/4
SVC module

- DBRC does not have to be down when the update runs
- Determines the SVCs to be updated:
 - SVCTYPE=(2) or SVCTYPE=(4) or SVCTYPE=(2,4)
 - Dynamically changes the z/OS SVC table to point to new versions of the SVC modules
- IMS still needs to be inactive if the Type 2 SVC is being updated

 MESSIAH CONSULTING

KC110 unit 11d page 12

Notes:

DFSUSVC0 has been enhanced to support dynamic updates of the Type 4 SVC in addition to the previous support for the Type 2 SVC.

As in previous releases, for updates to the Type 2 SVC, the utility checks to see if there is an active IMS using the SVC number. If so, the update fails. The same does not apply to the DBRC Type 4 SVC. DBRC regions are not required to be down when the Type 4 SVC is updated.

The utility determines the SVCs to be updated and dynamically changes the z/OS SVC table to point to the new SVC modules.

IV3D301T - IV3D307T: VTAM – Interface highlights

IMS - Definition		VTAM - Definition	
COMM	APPLID= IMSX , RECANY=(nnn,size), OPTIONS=(VTAMAUTH...)	HUGO	VBUILD APPL AUTH=(ACQ),EAS=50, ACBNAME= IMSX
	.		
	.		
	.		or
TYPE	UNITYTYPE=SLUTYPE2	IMSX	APPL AUTH= (ACQ),EAS=50
TERMINAL	NAME=NODE1		
NAME	termname		
			VBUILD
		RP1	PU PUTYPE=2
		NODE1	LU LOCADDR=2,

Notes:

For Terminals supported via VTAM:

The VTAM-MODETABLE must contain entries for all IMS-Terminals

IMS must be defined as VTAM-APPL

APPLID= can now be overridden with APPLID1= Parameter in DFSPBnnn
Parameter member

Size for SLUTYPE1 and SLUTYPE2 Terminals, the buffer size specified will affect
the maximum size input record that can be received

IMS Gen VTAM Node definition

```
*****
* LOCAL TERMINALS   SNA CONTROLLED      CLASS-ROOM   2-404
*****
*
      TYPE          UNITYPE=SLUTYPE2,TYPE=3270-A02,SIZE=(24,80),
                      FEAT=IGNORE,OPTIONS=TRANRESP,OUTBUF=1536
*
      TERMINAL      NAME=FD079013
      NAME          LT40401S
```

Notes:

Note the hierarchy of IMS VTAM macros. The TYPE macro is supplying values, including the OUTBUF= for TERMINAL macros; alternately, we could code OUTBUF= on each TERMINAL macro.

Since ETO (or other vendor tools as ETA / DELTA/VT from BMC) there is no need for a static GEN'd TERMINAL anymore. See the ETO descriptors for that !

VTAM definition (1 of 2)

× Member ATCCON21 in SYS1.VTAMLST

```
*/-----  
*/ A T C C O N 2 1 - SUBAREA 21 (HEROS1)  
*/-----  
AM01HER1,  
LX7610S,LZ7611S,LZ7612S,LY76620N,LX766A0N,LZ7613S,  
LZ7614S,JES328,CTC21T10,CTC21T11,CTC21T12,CTC21T15,CTC21T22,  
PFADE21,CDRMHB,CDRSCHB
```

× Member **AM01HER1** in SYS1.VTAMLST

```
*****  
*  
*           I M S X  
*  
*****  
FDJAIMS  APPL  EAS=160,                ESTIMATED CONCURRENT SESSIONS  
          ACBNAME=IMSX,                REAL APPLID  
          AUTH=(ACQ,BLOCK,PASS) IMS CAN ACQUIRE & PASS TMLS  
*                               IMS CAN REQUEST BLOCKED INPUT
```

Notes:

Here we see where the IMS VTAM APPLID of IMSX is defined in VTAM.

APPLID / ACBNAME , IMSID and many more PARMs are now dynamically controlled by DFSPBxxx or DFSDC... PARM settings !

VTAM definition (2 of 2)

× Member ATCCON21 in SYS1.VTAMLST

```
AM01HER1,  
LX7610S, LZ7611S, LZ7612S, LY76620N, LX766A0N, LZ7613S,
```

× Member LZ7611S in SYS1.VTAMLST

```
FD079013 LU LOCADDR=14, PORT A12  
DLOGMOD=MOD2,  
MODETAB=MT327BE2
```

× MODETAB member MT327BE2 in SYS1.SOURCE

```
* 327X MODELL 2  
*-----  
MOD2 MODEENT LOGMODE=MOD2, FMPROF=X'03', TSPROF=X'03',  
PRIPROT=X'B1', SECPROT=X'B0', COMPROT=X'3080',  
RUSIZES=X'88C7', <===== FOR OUTBUF ==  
PSERVIC=X'0280000000000000000000000200'  
MODEEND  
END
```

KRIESSICH CONSULTING

KC110 unit 11d page 16

Notes:

Here we continue our VTAM definition with the focus on the node we saw several pages ago in our IMS Gen. We have finally gotten to the RUSIZES parameter for our node.

Formula to verify OUTBUF size

Secondary LU (Sendsize **from** Screen)

Primary LU (Sendsize **to** Screen)

RUSIZES=

X' 8 8 C 7 '

$12 * (2 * * 7) = 12 * 128 = 1536 \text{ bytes}$
for OUTBUF

$8 * (2 * * 8) = 8 * 256 = 2048 \text{ bytes}$

Lower or equal to RECEIVE ANY BUFFER SIZE

Notes:

Here we see how the RUSIZE is encoded with an input and an output size.