

z/OS PARMLIB Successful Practices - User Experience

**Jack Schudel
University of Florida
schudel@ufl.edu**

**Thursday, August 11, 2011
Session 09800**

D PARMLIB

IEE251I 05.00.41 PARMLIB DISPLAY

PARMLIB DATA SETS SPECIFIED

AT IPL

ENTRY	FLAGS	VOLUME	DATA SET	
1	S	MVSIP1	SYS1.SYSRES.PARMLIB	on SYSRES
2	D	SYS901	SYS1.PARMLIB	SHARED
3	S	MVSIP1	SYS1.CPAC.PARMLIB	on SYSRES
4	S	MVSIP1	SYS1.IBM.PARMLIB	on SYSRES

Why multiple PARMLIBs?

- **SYS1.CPAC.PARMLIB**
 - Provided as part of SERVER PAC
 - Contains tailored members IFAPRD00 and others
 - No SMP/E DDDEF

- **SYS1.IBM.PARMLIB**
 - Maintained by SMP/E
 - Release specific IPCS, DAE and other members
 - Currently recommended settings, subject to change

- **SYS1.SYSRES.PARMLIB**
 - Release or maintenance specific members
 - Empty except during migration to new release or maintenance level
- **SYS1.PARMLIB**
 - Installation specific members
 - Common across all lpar
 - Use system symbolics in suffix for lpar specific members

Why SYS1.SYSRES.PARMLIB?

- **New release or maintenance may require new or different options, datasets or OMVS filesystems.**
- **New release may have renamed something.**
- **With a local, overriding PARMLIB, rolling any changes required for a new z/OS release into a plex just requires changing the IPL address.**
- **Even in a mono-plex, having release dependant options on the res pack makes it simple to switch between releases, especially if you need to go back several releases to reproduce a customer problem.**

What's in SYS1.SYSRES.PARMLIB?

- For z/OS 1.11 to 1.12 Conversion:
 - **COMMNDxx**
 - **COM= ' S CSF, ICSFPROG=CSFMMAIN ' (1.11)**
 - **COM= ' S CSF, ICSFPROG=CSFINIT ' (1.12)**
 - **BPXPRMnF**
 - **< . . . >**
 - **MOUNT FILESYSTEM(' SYS1.OMVS.&SYSR1..JAVA6461 ')**
 - **MOUNTPOINT(' /usr/lpp/java/J6.0.1_64 ')**
 - **TYPE(HFS) MODE(RDWR)**
 - **< . . . >**

How long does it stay there?

- Normally the production **SYSRES.PARMLIB** is empty
- Merged into shared **PARMLIB** once all members of the sysplex are stable at the new level.

Where do you specify multiple PARMLIBs?

SYS1.IPLPARM(LOADxx) (subset)

IEASYM (00,L)

SYSPARM 00

NUCLST 00

PARMLIB SYS1.SYSRES.PARMLIB

PARMLIB SYS1.PARMLIB

PARMLIB SYS1.CPAC.PARMLIB

PARMLIB SYS1.IBM.PARMLIB

SYS1.PARMLIB must be in the list or MVS will add it

How does IPL find LOADxx?

MVS IPL LOADPARM (ddddxxM.) specifies:

- **dddd** **IODF and SYSn.IPLPARM device address**
- **xx** **LOADxx suffix**
- **M | P | A** **Messages, Prompt, or both**

- See “Loading the System Software” in MVS System Commands for details

Placement of LOADxx

During IPL, the system looks for LOADxx and NUCLSTxx in the first dataset that it finds, using the following order:

- **SYS0.IPLPARM through SYS9.IPLPARM on the IODF volume.**
 - **SYS1.PARMLIB on the IODF volume.**
 - **SYS1.PARMLIB on the sysres volume.**
-
- see “Placement of LOADxx” in MVS Initialization and Tuning Reference

Placement of LOADxx

“Once the system opens the master catalog, the system reads all other members from the SYS1.PARMLIB data set that is pointed to by the master catalog. This SYS1.PARMLIB might be different from the SYS1.PARMLIB data set to which the LOAD parameter points.”

- see “Loading the System Software” in MVS System Commands

Other Approaches

Jim Petersen, Home Depot



<code>SYS1.PARMLIB</code>	On release catalog volume
<code>CPAC.PARMLIB</code>	On release catalog volume
<code>SYS1.IBM.PARMLIB</code>	On SYSRES

- **New master catalog created for each z/OS Release.**
- **SYS1.PARMLIB copied from old release and updated as needed.**

Jim Petersen, Home Depot



- **Heavy use of Symbolics for such things as:**
 - **Each CA Product Release Level (such as R120)**
 - **Each CA Product Runtime Level (maintenance rollout Level – RT1, RT2, RT3, up to RT5)**
- **Symbolics for zOS Release Level (ZOS112)**
- **Symbolics used for DB2 names and volumes the libraries for that DB2 live on.**

Jim Petersen, Home Depot



- **When doing maintenance code new member such as PROG02N which has changes in it.**
 - **Then rename PROG02 to PROG02\$ and PROG02N to PROG02 before IPLing LPAR2**
 - **Rename IEASYM00 to IEASYM0\$ and IEASYM0N to IEASYM00**
- **This is just a couple of examples of how we maintain SYS1.PARMLIB at Home Depot**
- **It is an evolving work as we discover new ways to make our job easier.**

Frank Lawrence, SunTrust



```
APLEX1    SYS1.ATLA.ATLPLEX.PARMLIB
APLEX1    SYS1.SUNTRUST.ATLPLEX.PARMLIB
PIORAP    SYS1.IBM.PARMLIB
VSAMA1    SYS1.PARMLIB
```

- We still build a new Master Cat for each new release.
- 'SYS1.SUNTRUST.ATLPLEX.PARMLIB' is the production SYSPLEX parmlib of everything that is identical throughout the PLEX, and is shared by all members of the PLEX.
- 'SYS1.ATLA.ATLPLEX.PARMLIB' is for each specific LPAR (one unique to each LPAR). There are six LPARS in the PLEX (ATLA, ATLB, PRPC, PRPD, PRNA, and PRNB).

Sam Knutson, GEICO

```
SYS1.PARMLIB          <-- Shared across Sysplex  
SYS1.GEICO.PARMLIB    <-- Shared across Sysplex  
SYS1.IBM.PARMLIB      <-- on IPL volume
```

- We update the existing master catalog.
- Our PARMLIB's are concatenated in front of the IBM SMP/E maintained PARMLIB on the sysres.
- We use System Symbols to use different members for different releases/LPARs in the same Sysplex.
- SYS1.PARMLIB for config data.
- GEICO.PARMLIB for slip traps and other data managed by non-systems staff.

Skip Robinson, SCE



ENTRY	FLAGS	DATA SET	Sysres?	Source/Content
1	D	SYS1.PARMLIB	No	Tailored content shared across sysplex
2	S	SYS1.CPAC.PARMLIB	Yes	ServerPac/user, not defined to SMPE
3	S	SYS1.IBM.PARMLIB	Yes	ServerPac misc, SMPE target
4	S	SYS1.SHASPARM	Yes	ServerPac JES2 IPCS, SMPE target

SCE PARMLIB Data Sets

- **Shared installation library concatenated first**
- **Avoids unexpected collision with IBM supplied like-named members**
- **We use CPAC.PARMLIB as 'our sysres library'**
 - **Already supplied on sysres**
 - **Already has appropriate RACF profiles**
 - **E.g. Contains tailored IBM PROGxx members**

SCE PARMLIB Data Sets – War Story

In a previous shop we placed the one and only SYS1.PARMLIB on sysres. That sounds sensible, but in practice no one can control the rest of the environment.

We once had a near disaster when the network guy renamed a VTAM library and updated APF list in PARMLIB. The next time we migrated maintenance, PARMLIB contained the old name. VTAM would not come up. The data center was 400 miles away. Fixing the problem remotely was a nightmare. ;-(((

SCE PARMLIB Data Sets

- Member naming conventions
- Use &SYSCONE for system dependent members
 - IEFSSN&SYSCONE
- Use '#' suffix to indicate new (future) version
- Use '@' suffix to indicate old (previous) version
- Place #/@ as far right as possible
 - E.g. successive versions of IEFSSN00
 - IEFSSN0# → IEFSSN00 → IEFSSN0@
 - Rename first 00 → 0@, then 0# to 00
 - Preserves ISPF statistics for tracking purposes
 - Ver.**Mod** embedded in line numbers
 - 000000**23** indicates line changed in 01.**23**

SCE PARMLIB Data Sets

- Follow the bouncing ball
- In LOADxx: IEASYM (01,**02**,03,L)
- In IEASYM**02**: SYSPARM(**X1**)
 - Determined by Hardware Name plus LPAR Name
 - &SYSCclone set to 'X1'
- In IEASYS**X1**:
 - PAGE=(SYS1.PAGELPA0.\$SYS&SYSCclone,
 - SYS1.PAGECOM0.\$SYS&SYSCclone,
 - SYS1.PAGELOC0.\$SYS&SYSCclone,
 - SYS1.PAGELOC1.\$SYS&SYSCclone,
 - SYS1.PAGELOC2.\$SYS&SYSCclone,L)

SCE PARMLIB Data Sets

- **Special handling for PROGxx: IBM vs. the world**
- **PROGlx in CPAC.PARMLIB contains all IBM entries**
 - Derived from ServerPac supplied PROG00
 - I1: Prolog plus IBM-supplied APF entries
 - Includes "DYNAMIC" and "DEFAULTS" statements (R12)
 - I2: IBM-supplied linklist entries
 - I3: "LNKLST ACTIVATE NAME(LNKLST00)"
- **PROG0x in SYS1.PARMLIB contains non-IBM entries**
 - 01: ISV and local APF entries
 - 02: ISV and local linklist entries
 - 03: Dynamic exit definitions
- **In IEASYS00: PROG=(I1,01,I2,02,I3,03)**

SCE PARMLIB Data Sets

Problems with multiple versions of the same PARMLIB member in multiple PARMLIB data sets

- 1. Multiple members may have to be kept in sync for long periods. Who remembers to edit them all every time?**
- 2. Who remembers to collapse 'temporary' differences back into a single shared member?**
- 3. When is the right time to reconsolidate?**

References

- **z/OS V1R12.0 MVS System Commands**
 - SA22-7627-24
 - Loading the System Software
 - Displaying PARMLIB Information
- **z/OS V1R12.0 MVS Initialization and Tuning Reference**
 - SA22-7592-21
 - LOADxx (system configuration data sets)
 - Placement of LOADxx
 - Symbolic Parmlib Parser

Questions?

Appendix

SYS1.PARMLIB(IEASYM00)

```
/* NER1 system using MVSPROD partition */
SYSDEF LPARNAME(MVSPROD) SYSNAME(NER1)
SYSDEF LPARNAME(MVSPROD) SYMDEF(&NERSYSID.='P')
SYSDEF LPARNAME(MVSPROD) SYMDEF(&NERTCPIP.='NER')
/* NER2 system using MVSALT partition */
SYSDEF LPARNAME(MVSALT) SYSNAME(NER2)
SYSDEF LPARNAME(MVSALT) SYMDEF(&NERSYSID.='A')
SYSDEF LPARNAME(MVSALT) SYMDEF(&NERTCPIP.='NERA')
/* NER5 system using MVSYES partition */
SYSDEF LPARNAME(MVSYES) SYSNAME(NER5)
SYSDEF LPARNAME(MVSYES) SYMDEF(&NERSYSID.='Y')
SYSDEF LPARNAME(MVSYES) SYMDEF(&NERTCPIP.='NERY')
/* all system defaults */
SYSDEF SYSCLONE(&SYSNAME(-1:1))
SYSDEF SYMDEF(&SYSR2='&SYSR1(1:4).S&SYSR1(6:1)')
SYSDEF SYMDEF(&SYSR3='&SYSR1(1:4).T&SYSR1(6:1)')
SYSDEF SYMDEF(&NERGRSRP.='JOIN')
SYSDEF SYMDEF(&NERRMFCH.='NOCACHE')
/* NER1 exceptions to override defaults */
SYSDEF LPARNAME(MVSPROD) SYMDEF(&NERRMFCH.='CACHE')
/* NER5 exceptions to override defaults */
SYSDEF LPARNAME(MVSYES) SYMDEF(&NERGRSRP.='NONE')

/* 1 character system id
/* 2nd IPL volume
/* 3rd IPL volume
/* GRS default
/* RMF default
/* CACHE stats only 1
/* GRS=NONE on sandbox
```

SYS1.PARMLIB(IEASYS00)

```
ALLOC=&SYSCONE.0 ,  
CMD= ( &SYSCONE.0 ,00 ) ,  
CON= ( &SYSCONE.0 ,L ) ,  
COUPLE=&SYSCONE.0 ,  
GRS=&NERGRSRP ,  
LOGREC=SYS1.&SYSNAME..LOGREC ,  
OMVS= ( &SYSCONE.0 ,&SYSCONE.F ) ,  
PROD= ( 00 ,&SYSCONE.0 ) ,  
PROG= ( 00 ,L0 ,E0 ) ,  
SMF=&SYSCONE.0 ,  
PAGE= ( SYS1.&SYSNAME..PAGE.PLPA ,           +  
        SYS1.&SYSNAME..PAGE.COMM ,           +  
        SYS1.&SYSNAME..PAGE.LOCAL01 ,L )
```