

# **Stop Those Unnecessary IPLs – The Dynamic OS/390 Functions**

**Kenneth.Tomiak@Schunk-Associates.com**

**February 28, 2003**

**Session 2812**

**Trademarks and Copyrights**

- Trademarks and Registered trademarks used in this presentation are the property of their respective owners.
- Copyright ©2003 Schunk and Associates, Inc.
- Permission is hereby granted to SHARE, Inc. to distribute this presentation to conference attendees and in the conference proceedings. All other rights reserved.
- The information presented here has not been subject to any formal review. Use with your own discretion. No warranty is expressed nor implied.

Visit [www.Schunk-Associates.com](http://www.Schunk-Associates.com) to see this and other presentations made by the staff of Schunk and Associates, Inc. The latest and most up-to-date version is always found here.

**Agenda**

- When is an IPL required?
- Other avoidable reasons to IPL
  - Hardware changes
  - System parameter changes
  - Working with Subsystem data sets
  - Software changes
- Summary

**When is an IPL Required**

- Changing the PROCLIB/PARMLIB concatenation in SYS1.PARMLIB(MSTJCLxx)
- Changes in SYS1.NUCLEUS
- Addition of a product which adds a new SVC (type 1, 2, or 6)
- Addition of a product subsystem (start order dependent)
- Addition of a product module to LPA (called by Program Control)
- Addition of non-dynamic hardware
- Need to change an IPL required parameter (PAGTOTL)
- Power outage

***Other Avoidable Reasons to IPL***

- Four major reasons why an IPL was scheduled in the past:
  - Hardware changes
  - System parameter changes
  - Working with Subsystem data sets
  - Software changes
- But this meant incurring an outage. Installations looking to achieve 24 by 7 availability wanted ways to make these changes without an IPL.
- In this session we will discuss how you can effect implementation of these types of changes while avoiding IPLs.

***Hardware Change: IPL Avoidance***

- Hardware Configuration Definition (HCD) provides dynamic I/O reconfiguration management.
- This function allows you to change your peripheral hardware and software definitions dynamically - such as adding or changing devices, channel paths, and control units. Without performing a Power On Reset (POR) or an Initial Program Load (IPL).
- The HCD component of MVS consolidates the hardware and software I/O configuration processes under a single, interactive, end-user interface.
- HCD does validation checking as you enter data to help eliminate errors before you attempt to use the I/O configuration.
- You may also perform software only changes before the hardware has been installed.

***System Parameter Change: Reasons***

- LOADxx changes - modify PARMLIB concatenation
- IEFSSNxx additions - add a Subsystem
- IKJTSOxx changes - TSO/E parameters
- IFAPRDxx changes - Product authorization

**System Parameter Change: SETLOAD**

In OS/390 it is now possible to concatenate PARMLIB.

- You can concatenate up to ten (10) parmlib datasets in the LOADxx member using the PARMLIB keyword.
- The SETLOAD command allows you to switch dynamically from one parmlib concatenation (logical parmlib) to another without having to initiate an IPL.
- The SETLOAD command specifies the LOADxx member that contains the PARMLIB statements to use for the switch.
- Use the 'D PARMLIB' command to display the PARMLIB concatenation before and after using the SETLOAD.

IEE251I 09.07.04 PARMLIB DISPLAY 958

PARMLIB DATA SETS SPECIFIED

AT IPL

ENTRY FLAGS VOLUME DATA SET

```

1   D   OSIODF SYS1.PARMLIB
2   S   OSR2P1 SYS1.IBM.PARMLIB

```

**System Parameter Change: SETLOAD**

- This command tells the system to process the PARMLIB statements in member LOAD02, which resides in a data set in the existing parmlib concatenation.  
SETLOAD 02,PARMLIB
- This command tells the system to process the PARMLIB statements in member LOAD03. Member LOAD03 resides in the data set 'sys4.parmlib'. Data set 'sys4.parmlib' is catalogued in the master catalog.  
SETLOAD 03,PARMLIB,DSN=sys4.parmlib
- Add VOL= if the data set is not catalogued.

**System Parameter Change: SETLOAD**

There are two issues to be aware of when using concatenated PARMLIBs.

- Concatenation search order applies, the first data set containing the member is used. Be sure to concatenate data sets in the correct order so as to not invoke an incorrect member.
- The current HWNAME, LPARNAME, and VMUSERID filters are compared to the actual system values.
  - Libraries specified on every PARMLIB keyword that apply to this system are concatenated in the order they are processed.

**System Parameter Change: SETSSI**

A subsystem is a service provider that provides one or more functions, but only when it is requested to. Although the term "subsystem" is used in other ways, a subsystem must be either the master subsystem (the master subsystem (MSTR) is a part of MVS and is not user-defined) or be defined to MVS in one of the following ways:

- In the IEFSSNxx parmlib member during an IPL
- Issuing the SETSSI system command

Some examples of IBM-supplied subsystems that use the SSI:

- Primary
  - JES2 or JES3
- Secondary
  - Netview, DB2, CICS, OPC

**System Parameter Change: SETSSI**

Use the SETSSI to dynamically add, activate or deactivate a subsystem.

- You can issue the SETSSI command from one of the following:
  - A console that has master authority
  - A console to which an operator with sufficient RACF authority has logged on.
- If you issue a SETSSI ACTIVATE or DEACTIVATE command for a subsystem that does not allow SETSSI commands, the system ignores the command and issues an error message to the console.

**SETSSI DEACTIVATE, SUBNAME=KENT**

```
IEFJ023I SETSSI DEACTIVATE COMMAND FOR SUBSYSTEM KENT COMPLETED WITH ERRORS
IEFJ036I SUBSYSTEM KENT IS NOT ENABLED FOR THE SETSSI COMMAND
```

**System Parameter Change: SETSSI Examples**

- The SETSSI ADD command allows any subsystem except the primary subsystem to be dynamically defined.
- To define the 'CAW' subsystem to the system, call its initialization routine and pass the specified parameter to the initialization routine:
 

**SETSSI ADD, SUBNAME=CAW, INITRTN=CAWINIT, INITPARM=HELLO**
- To temporarily stop new function requests to the subsystem to see if one of the function routines in the 'CAW' subsystem is causing abends:
 

**SETSSI DEACTIVATE, SUBNAME=CAW**

**System Parameter Change: TSO PARMLIB**

- Display the specifications in the active IKJTSOxx member of SYS1.PARMLIB
- Check the syntax of any IKJTSOxx member of SYS1.PARMLIB
- Dynamically change the active member without an IPL
  - one system
  - all systems in a parallel sysplex, or
  - a subset of systems in a parallel sysplex
- Note: You can also place IKJTSOxx members in data sets other than SYS1.PARMLIB.

**System Parameter Change: Product Authorization**

- You license a product
- You unlicense a product
- Update the alternate IFAPRDxx member for appropriately licensed products and issue the set command:

**T PROD=xx**

Upon successful completion implement this member for the next IPL.

**Working with Subsystem data sets: IPL Avoidance**

- JES2 PROCLIB
- PAGE Data Sets: ADD/REMOVE Page Data sets
- RACF Database: Switch/Rename RACF Data base
- Catalogs: ALLOCATE(OPEN)/CLOSE ICF User Catalogs

**System Parameter Change: JES2 PROCLIB**

- Use the JCLLIB statement in your batch jobs instead.
- The active JES2 PROCxx concatenations can be changed by updating the JES2 Procedure JCL and doing a hot start of JES2.
- A hot start is a warm start of an abnormally terminated JES2 member without an intervening IPL.
- JES2 performs a hot start when a particular JES2 member has stopped but other systems have continued to function and have not experienced problems.
- When JES2 hot starts, all address spaces continue to execute as if JES2 had never terminated. Jobs that were processing on output devices are re-queued as if a \$I command had been issued. Jobs on input devices must be resubmitted and lines must be restarted. Hot starts have no affect on other members in a MAS configuration.

**System Parameter Change: PAGE Data set ADD**

PAGEADD adds auxiliary storage space (local page data sets) to the system. The added page data sets remain available to the system until you IPL with the CLPA (create link pack area) option, IPL with the CVIO (clear virtual I/O) option, or issue the PAGEDEL command. PAGEADD can also direct VIO pages away from a page data set as it is being added.

An indication you may need additional auxiliary storage space is message:  
IRA200I AUXILIARY STORAGE SHORTAGE

The number of page data sets that can be in use by the system is limited by the number specified on the PAGTOTL system parameter in the IEASYSxx member (see OS/390 MVS Initialization and Tuning Guide). If these limits are exceeded, the system issues a message, and you cannot add any more data sets of that type for the duration of this IPL.

**System Parameter Change: PAGE Data set ADD**

- The page data sets must exist before you can issue the PAGEADD command.
- If the volume containing the data set is not mounted the system issues a mount message.
- The system rejects LOGONs and START commands and might delay the starting of certain initiators until the shortage is relieved.
- To identify the page and swap data sets the system is currently using or the status of the PAGEDEL command, issue the DISPLAY ASM command.  
**D ASM, ALL**

```
IEE200I 09.05.55 DISPLAY ASM 885
TYPE  FULL STAT  DEV DATASET NAME
PLPA   81%  OK 10A3 PAGE.M5PROD.PLPA
COMMON 0%   OK 10A3 PAGE.M5PROD.COMMON
LOCAL  0%   OK 10A3 PAGE.M5PROD.LOCAL1
NO SWAP DATASETS ARE IN USE
PAGEDEL COMMAND IS NOT ACTIVE
```

**System Parameter Change: PAGE Data set ADD**

- To add one page data set:  
**PA PAGE=sys1. local 3**  
IEE783I PAGEADD COMMAND- SYS1.LOCAL3 PAGE DATA SET  
NOW AVAILABLE FOR SYSTEM USE
- To add SYS1.PAGE01 as a page data set, not to be used for VIO paging:  
**PA NONVIO=sys1. page01**

**System Parameter Change: PAGE Data set DELETE**

- PAGEDDEL may be used to delete, replace, or drain local page data sets. This command allows local page data sets to be removed or replaced without requiring an IPL.
- You might need to delete, replace or drain local page data sets for any of the following reasons:
  - The hardware is being reconfigured.
  - The hardware is generating I/O errors.
  - The page or swap configuration is being changed.
  - System tuning requires the change.

NOTE: Misuse of this command can have a serious negative impact on system performance.

**System Parameter Change: PAGE Dataset DELETE**

- You cannot use PAGEDDEL to delete, replace, or drain the PLPA, common, duplex, or the last local page data sets.
- When you enter a PAGEDDEL command, the system issues a highlighted, non-rollable message to indicate that the command is accepted. The message remains on the console screen until the PAGEDDEL command completes.
- When you enter a PAGEDDEL command while a PAGEDDEL command is already in progress, the system issues a message to indicate that the command is not accepted.
- The system rejects a PAGEDDEL command that decreases the amount of auxiliary storage below a fixed percentage of the available auxiliary storage.

**System Parameter Change: PAGE Dataset DELETE**

- To delete a local page data set:  
**PD DELETE, PAGE=SYS1. LOCAL3**  
IEE205I PAGEDEL COMMAND - LOCAL PAGE DATA SET SYS1.LOCAL3 DELETED
- To replace SYS1.PAGE01, a local page data set, with SYS1.PAGE04:  
**PD REPLACE, PAGE=(sys1. page01, sys1. page04)**

**System Parameter Change: RACF Data base Switch**

- Use the RACF command to perform RACF Data base Activate/Deactivate: Switch from using a primary data base to its corresponding backup data base because of a failure related to the primary data base.
- Deactivate a RACF data base in order to de-allocate the data set.
  - Deactivating an active primary data base will cause requests to that data base to fail. You should SWITCH first.
  - Deactivating an active backup data base will cause RACF to stop duplicating information in that data base. You should INACTIVATE the backup first.
- Activate a RACF data base data set in order to allocate the data set before you SWITCH to the data base.

**System Parameter Change: CATALOG CLOSE/ALLOCATE**

- Use the MODIFY CATALOG command to communicate with the catalog address space (CAS) to display information or request a specified service.
- When an operator issues any MODIFY CATALOG command (except for MODIFY CATALOG,RESTART), messages return to that console exclusively.
- For MODIFY CATALOG,RESTART, both the master console and the console issuing the command receive messages.

**System Parameter Change: CATALOG CLOSE/ALLOCATE**

- These two commands will free up a user catalog from the catalog address space. Once the user catalog is unallocated you can perform the maintenance action you need.

```
F CATALOG, CLOSE(UCAT. APPDEVL1)
F CATALOG, UNALLOCATE(UCAT. APPDEVL1)
```

**System Parameter Change: CATALOG CLOSE**

- Closes an integrated catalog facility catalog dynamically, without affecting any existing allocations.
- With this parameter, you can dynamically change catalog attributes, such as share options and the number of strings. In the past, changing these attributes on a master catalog would have required an IPL, or the termination and restart of a job or online system to affect a user catalog.
- All of the catalog address-space private storage associated with the catalog is freed.
- **The catalog is reopened with a new set of control blocks the next time a request is processed for that catalog.**
- Rebuilding of the control blocks is transparent to the users of the catalog.

**Working with Subsystem data sets: LNKLST****XCFAS CONSIDERATIONS:**

XCFAS holds a SYSDSN ENQ on the active LNKLST sets, which has to be removed to change LNKLST sets. This is a sample procedure using the SETPROG operator command:

```
SETPROG LNKLST, UNALLOCATE
P LLA
```

Continue with the appropriate LNKLST set changes:

```
SETPROG LNKLST, ADD...
SETPROG LNKLST, ACTIVATE...
SETPROG LNKLST, ALLOCATE
S LLA
```

**Software Change: Reasons**

- System parameter - see previous pages
- Change to the System search order
- DD statements - task search order
- LPA
- LNKLST
- Change to System exit order

**Software Change: System Search Order**

When a program requests a module, the system searches for the requested module in various system areas and libraries, in the following order:

1. Modules that were loaded under the current task (Load List Elements) or the job pack area (JPA), area of your storage where programs are loaded,
2. Tasklib, steplib, joblib, or any libraries that were indicated by a DCB specified as an input parameter to the macro used to request the module (LINK, LINKX, LOAD, ATTACH, ATTACHX, XCTL or XCTLX),
3. Active Link Pack Area (LPA), which contains the FLPA and MLPA,
4. Pageable link pack area (PLPA),
5. SYSLIB and libraries concatenated through LNKLST.

**Software Change: System Search Order**

- When searching TASKLIBs, STEPLIBs, JOBLIBs, a specified DCB, or the LNKLST concatenation for a module, the system searches each data set directory for the first directory entry that matches the name of the module.
- The directory is located on DASD with the data set.
- The directory entry contains information about the module and its location on DASD storage.

**Software Change: System Search Order**

- Fixed LPA, Modifiable LPA, and Pageable LPA comprise the Link Pack Area, an area of storage that contains re-enterable routines that are loaded at IPL time and can be used concurrently by all tasks in the system.
  - LNKLST is a user-specified set of libraries that form part of the search order the systems uses to locate programs.
  - LLA (LNKLST LookAside) caches, in its address space, a copy of the directory entries for the libraries it manages. For modules that reside in LLA-managed libraries, the system can quickly search the directories in virtual storage instead of using I/O to search the directories on DASD.
- LPA, LNKLST and LLA are defined and initiated at IPL time.

**Software Change: IPL Avoidance**

- JOBLIB/STEPLIB DD Statement
- LLA: Modify LLA (LNKLST LookAside)
- LNKLST: Dynamic LNKLST
- LPA: Dynamic LPA
- EXIT: Dynamic EXIT

**Software Change: JOBLIB/STEPLIB DD**

Use a JOBLIB DD statement to define a private library that the system is to use for an entire job. Use a STEPLIB to define a private library that the system is to use for one step. If you include a JOBLIB DD statement and a STEPLIB DD statement, the system searches the step library and not the JOBLIB for the program requested in the EXEC statement for that step. The system always continues the search in SYSLIB/LNKST if the program is not found in the private library.

```
//ANYJOB JOB . . .
//JOBLIB DD DISP=SHR, DSN=HLQ.PRIVATE.LIB4
//STEP1 EXEC PGM=SCAN
//STEPLIB DD DISP=SHR, DSN=HLQ.COMMON.LIB5
//STEP2 EXEC PGM=UPDATE
```

The system searches for the program SCAN in HLQ.PRIVATE.LIB4, then in SYSLIB/LNKLST. The system searches for the program UPDATE in HLQ.COMMON.LIB5, then in SYSLIB/LNKST.

**Software Change: Modify LLA**

- When an LLA-managed library is updated, the version of a module that is located by a directory entry saved in LLA will differ from the version located by the current directory entry on DASD for that module.
- If you update a load module in a library that LLA manages, affect the update to the system, by issuing the appropriate form of the MODIFY LLA command to refresh LLA's cache with the latest version of the directory information from DASD.
- If LLA is NOT refreshed the system will continue to use the un-updated version of the load module.
- Use the MODIFY LLA command to cause the library lookaside (LLA) program to build a new copy of EITHER all or only part of the library directory indexes.

**Software Change: Modify LLA****MODIFY LLA,REFRESH**

- Rebuilds LLA's directory for the entire set of libraries managed by LLA.
- This action is a complete refresh of LLA.  
MODIFY LLA,UPDATE=xx  
xx identifies the CSVLLAxx member that contains the names of the libraries for which directory information is to be refreshed.
- Rebuilds LLA's directory only for specified libraries or modules.
- This action is called a selective refresh of LLA.

**Software Change: Coding PARMLIB(PROGxx)**

EXAMPLE : PROGxx To support Dynamic APF, LNKLST, and EXIT.

```
APF FORMAT(DYNAMIC)
APF ADD DSNAME(SYS1.LINKLIB) VOLUME(&SYSR1)
APF ADD DSNAME(SYS3.NCP.LOADLIB) SMS
APF ADD DSNAME(SYS1.EMPTY.LOADLIB) VOLUME(*MCAT*)
APF ADD DSNAME(SYS1.TEMP.APFLIB) VOLUME(D98RP1)
SYSLIB LINKLIB(SYS2.MYLOAD)
SYSLIB LPALIB(SYS2.MYLPALIB)
LNKLST DEFINE NAME(PROD00)
LNKLST ACTIVATE NAME(PROD00)
EXIT ADD EXITNAME(SYS1.IEFU83)
MODNAME(IEFU83A) DSNAME(ISV1.BETA.APFLIB)
EXIT ADD EXITNAME(SYS1.IEFU83)
MODNAME(IEFU83A) DSNAME(ISV1.BETA.APFLIB)
```

**Software Change: DYNAMIC LNKLST**

Dynamic LNKLST supports modification of LNKLST after an IPL:

- Define a LNKLST set of data sets for the LNKLST concatenation,
- Add data sets to or delete data sets from the LNKLST set,
- Remove the definition of a LNKLST set from the system,
- Test for the location of a specific module in the LNKLST concatenation,
- Activate a LNKLST set as the LNKLST concatenation for the system,
- Update an address space for jobs to use a LNKLST set.

Dynamic LNKLST changes are not retained across IPLs.

To implement a change permanently, it is necessary to make the change in the PROGxx member of SYS1.PARMLIB.

**Software Change: DYNAMIC LNKLST**

- The LNKLST set is defined at IPL time.
- Multiple LNKLST sets can be defined, however, only ONE LNKLST may be the CURRENT LNKLST set at any given time.
- The CURRENT LNKLST may not be modified...
  - It is necessary to create a new LNKLST,
  - Modify the new LNKLST,
  - Activate this new LNKLST to make it the new CURRENT LNKLST.
- LNKLST set can be dynamically changed by using the SETPROG, LNKLST command.
- When a new LNKLST is activated, LLA is automatically refreshed. The new current LNKLST set will then be managed by LLA as the LNKLST.

**Software Change: DYNAMIC LNKLST**

- Data sets from the previously current LNKLST continue to be managed by LLA, but only on an individual data set basis.
- The only way to remove them is to either Stop and Start LLA, or issue the MODIFY LLA, REFRESH or MODIFY LLA, UPDATE with the library to be removed.
- Here are some example SETPROG commands input from a system console:
  1. Create a new LNKLST set from the CURRENT LNKLST set.  
**SETPROG LNKLST, DEFINE, NAME=PROD01, COPYFROM=PROD00**

**Software Change: DYNAMIC LNKLST**

2. Remove the old version product library from the new LNKLST set.  
**SETPROG LNKLST, DELETE, NAME=PROD01, DSNAME=ISV1. QREF510. LINKLIB**
3. Add the new version product library to the new LNKLST set.  
**SETPROG LNKLST, ADD, NAME=PROD01, DSNAME=ISV1. QREF520. LINKLIB**
4. Activate this new LNKLST set.  
**SETPROG LNKLST, ACTIVATE, NAME=PROD01**

**Software Change: DYNAMIC LPA**

Dynamic LPA supports modification of LPA after an IPL:

- Modules that are to be added to the LPA following IPL.
- Modules that are to be deleted from the LPA following IPL.
- Threshold values for minimum amounts of CSA storage that still must be available after an ADD operation.

Dynamic LPA changes are not retained across IPLs.

To implement a change permanently, it is necessary to make the change in one of the members in SYS1.PARMLIB.

**Software Change: DYNAMIC LPA**

- LPA is searched in this order:
  - Dynamic LPA modules, as specified in PROGxx members
  - Fixed LPA (FLPA) modules, as specified in IEAFIXxx members
  - Modified LPA (MLPA) modules, as specified in IEALPAXx members
  - Pageable LPA (PLPA) modules, loaded from libraries specified in LPALSTxx or PROGxx
- A module added to LPA dynamically will be found BEFORE one with the same name added to LPA during IPL.

**Software Change: DYNAMIC LPA**

- Modules that are added to the LPA LST using the dynamic LPA function are placed in CSA or ECSA. The CSAMIN parameter, can be used to ensure that a certain minimum amount of CSA and ECSA will be available after a module is added.
- You need to confirm the product supports dynamic LPA.
- Dynamic LPA can only delete modules that were added dynamically. It does not support deletion of modules from PLPA, FLPA or MLPA.
- Storage used by a dynamically ADDED module, will be freed when/if that module is dynamically DELETED.

**Software Change: DYNAMIC LPA**

To add a module or to replace an existing LPA version:

**SETPROG LPA, ADD, MODNAME=IGC0024G, DSNAME=SYS2. MYLPA**

To delete this same modified version of the module, and revert to the LPA version, if one exists:

**SETPROG LPA, DELETE, MODNAME=IGC0024G**

**Software Change: DYNAMIC EXIT**

Use the SETPROG EXIT command to do the following:

- Add an exit routine to an exit
- Change the state of an exit routine
- Delete an exit routine from an exit
- Undefine an implicitly-defined exit
- Change the attributes of an exit.

You can use the SETPROG EXIT command to control exits that have been defined to the dynamic exits facility.

**Software Change: DYNAMIC EXIT**

Dynamic exits services are implemented by the EXIT statement of the PROGxx parmlib member:

- The EXIT statement of PROGxx allows an installation to add exit routines to an exit, delete an exit routine for an exit, change the state of an exit routine, change the attributes of an exit, and undefine an implicitly defined exit.
- The PROGxx EXIT statement interacts with the PROG=xx parameter of IEASYSxx and the SET PROG=xx command. At IPL, you can use PROG=xx to specify the particular PROGxx parmlib member the system is to use. During normal processing, you can use the SET PROG=xx command to set a current PROGxx parmlib member.

**Software Change: DYNAMIC EXIT**

- Associate exit routine MYMOD with the SMF exit known as SYS.IEFUJI, defined through the SYS statement in a SMFPRMxx parmlib member.
- The load module is in data set SYS2.MYLINK.  
**SETPROG EXIT, ADD, EXITNAME=SYS. IEFUJI, MODNAME=MYMOD,  
DSNAME=SYS2. MYLINK, STATE=ACTIVE**

**Summary**

- HCD and dynamic hardware changes,
- System commands to effect dynamic system parameter changes,
- System commands to effect dynamic software changes,
- Okay, schedule that IPL if you have to.

**Bibliography**

MVS JCL User's Guide

MVS JCL Reference

MVS Init and Tuning Guide

MVS Init and Tuning Reference

MVS Conversion Notebook

MVS System Commands

HCD User's Guide

HCD Planning

TSO/E System Programming Command Reference

Visit <http://www.schunk-associates.com/presentations> to download the Adobe PDF format of this presentation and handout. They appear under *Kenneth Tomiak – Share100*.