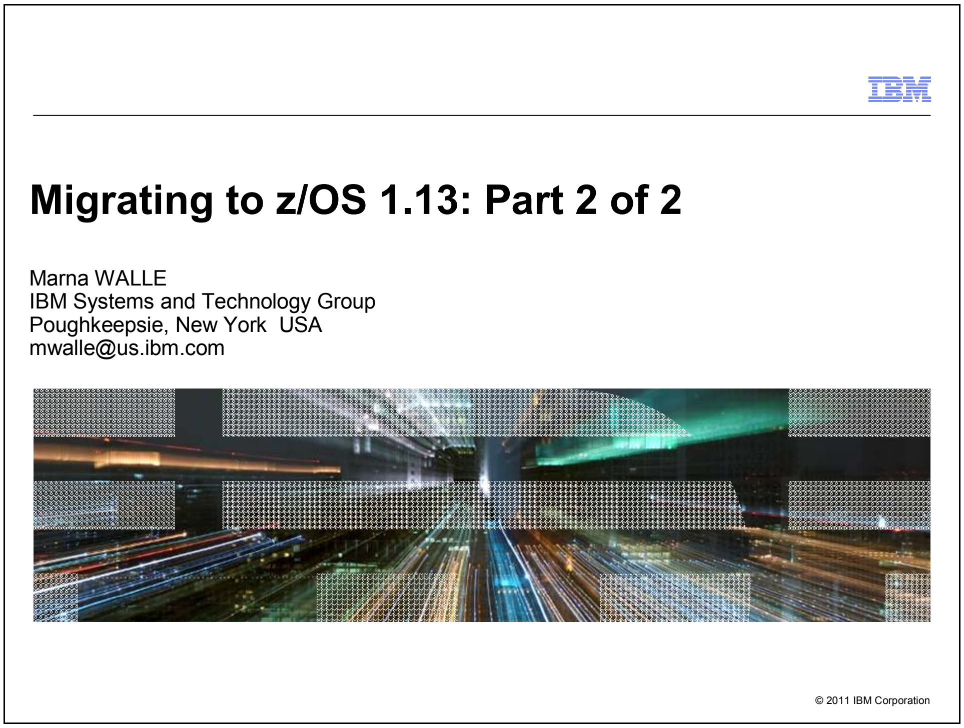


Migrating to z/OS 1.13: Part 2 of 2

Marna WALLE
IBM Systems and Technology Group
Poughkeepsie, New York USA
mwalle@us.ibm.com

Abstract: Migrating to z/OS 1.13: Part 2 of 2

This is part two of a two-part session that will be of interest to system programmers and their managers who are migrating to z/OS 1.13 from either z/OS 1.11 or z/OS 1.12. It is strongly recommended that you attend both sessions for a complete migration picture.

The general availability date for z/OS V1 R13 is planned for September 30, 2011.

Trademarks

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

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

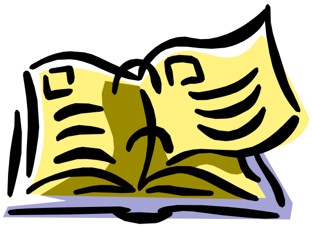
Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

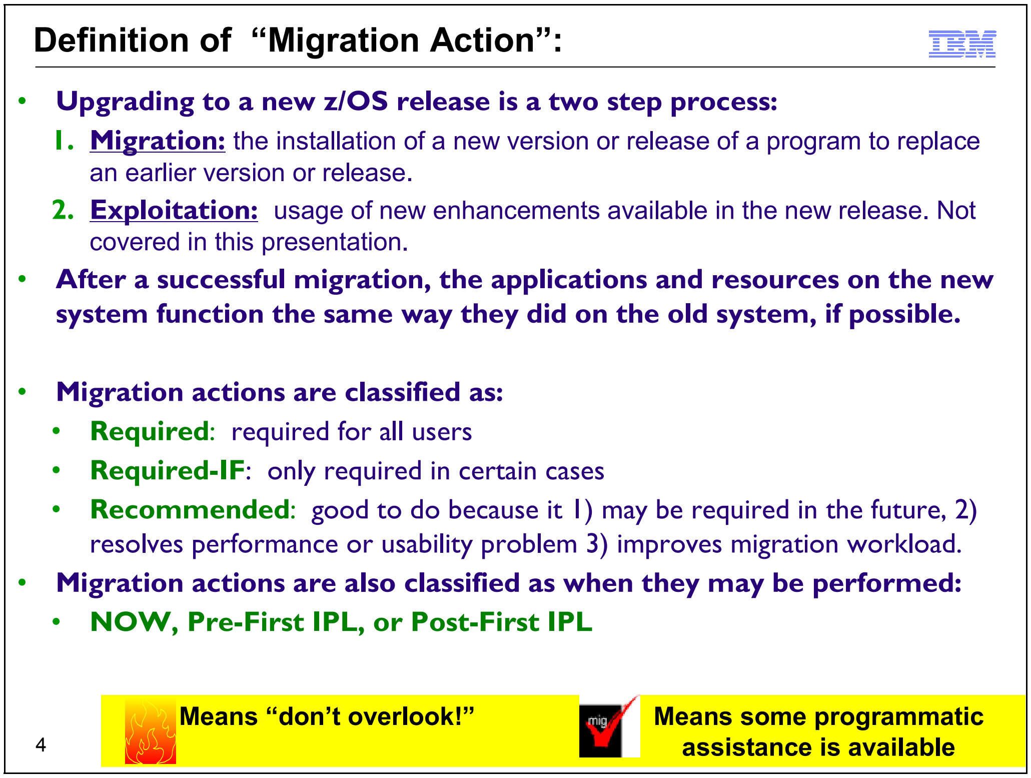
* All other products may be trademarks or registered trademarks of their respective companies.

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.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Migrating to z/OS R13: Part 2 of 2 Agenda

- **Migration Actions for z/OS R11 and R12:**
  - General Migration Actions
  - Element-Specific Migration Actions:
    - BCP
    - Communications Server
    - DFSMS
    - Distributed File Service: zFS
    - Infoprint Server
    - JES2 and JES3
    - z/OS UNIX
- **Summary**





Definition of "Migration Action":

- **Upgrading to a new z/OS release is a two step process:**
  1. **Migration:** the installation of a new version or release of a program to replace an earlier version or release.
  2. **Exploitation:** usage of new enhancements available in the new release. Not covered in this presentation.
- **After a successful migration, the applications and resources on the new system function the same way they did on the old system, if possible.**
- **Migration actions are classified as:**
  - **Required**: required for all users
  - **Required-IF**: only required in certain cases
  - **Recommended**: good to do because it 1) may be required in the future, 2) resolves performance or usability problem 3) improves migration workload.
- **Migration actions are also classified as when they may be performed:**
  - **NOW, Pre-First IPL, or Post-First IPL**

4

Means "don't overlook!"

Means some programmatic assistance is available

Migration Definitions and Classifications

Migration is the first of two stages in upgrading to a new release of z/OS. The two stages are:

- **Stage 1: Migration.** During this stage you install your new system with the objective of making it functionally compatible with the previous system. After a successful migration, the applications and resources on the new system function the same way (or similar to the way) they did on the old system or, if that is not possible, in a way that accommodates the new system differences so that existing workloads can continue to run. Migration does not include exploitation of new functions except for new functions that are now required.
- **Stage 2: Exploitation.** During this stage you do whatever customizing and programming are necessary to take advantage of (exploit) the enhancements available in the new release. Exploitation follows migration.

Migration Requirement Classification and Timing

The migration actions are classified as to their requirement status:

- ***Required***. The migration action is required in all cases.
- ***Required-IF.*** The migration action is required only in a certain case. Most of the migration actions in this presentation are in this category.
- ***Recommended.*** The migration action is not required but is recommended because it is a good programming practice, because it will be required in the future, or because it resolves unacceptable system behavior (such as poor usability or poor performance) even though resolution might require a change in behavior.

To identify the timing of migration actions, this presentation uses three types of headings:

- ***Now.*** These are migration actions that you perform on your current system, either because they require the current system or because they are possible on the current system. You don't need the z/OS V1R13 level of code to make these changes, and the changes don't require the z/OS V1R13 level of code to run once they are made. Examples are installing coexistence and fallback PTFs on your current system, discontinuing use of hardware or software that will no longer be supported, and starting to use existing functions that were optional on prior releases but required in z/OS V1R13.
- ***Pre-First IPL.*** These are migration actions that you perform after you've installed z/OS V1R13 but before the first time you IPL. These actions require the z/OS V1R13 level of code to be installed but don't require it to be active. That is, you need the z/OS V1R13 programs, utilities, and samples in order to perform the migration

actions, but the z/OS V1R13 system does not have to be IPLed in order for the programs to run. Examples are running sysplex utilities and updating the RACF database template.

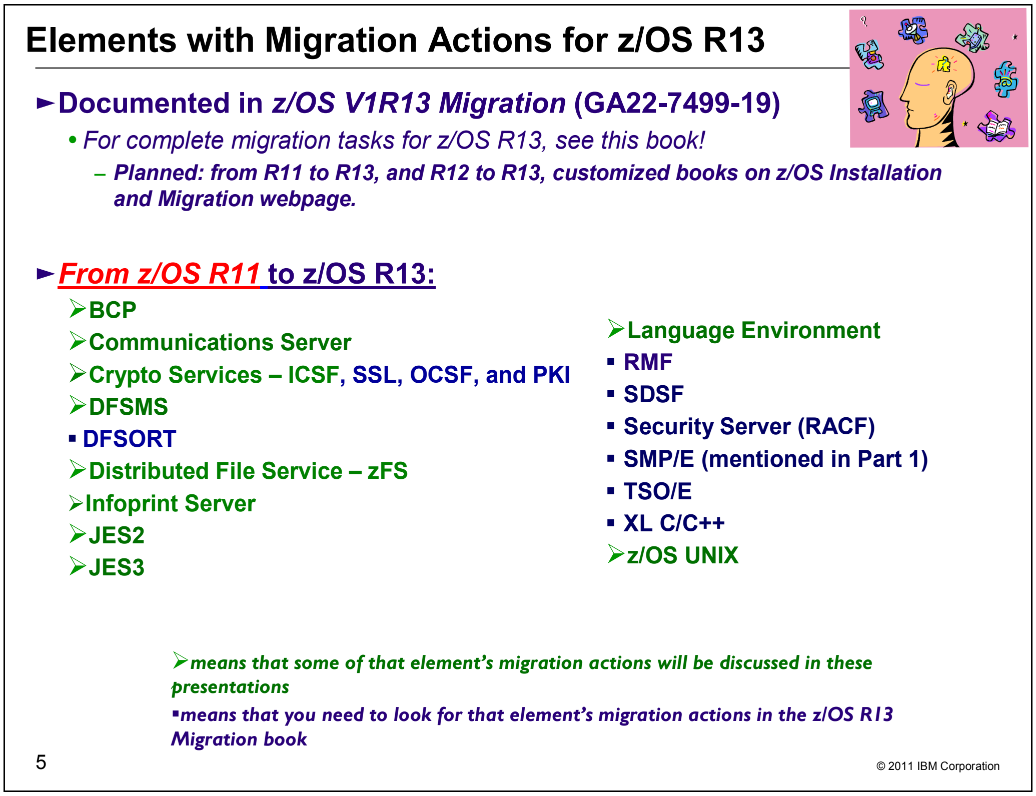
It is possible to perform some of the migration actions in this category even earlier. If you prepare a system on which you will install z/OS V1R13 by making a clone of your old system, you can perform migration actions that involve customization data on this newly prepared system before installing z/OS V1R13 on it. Examples of such migration actions are updating configuration files and updating automation scripts.

- ***Post-First IPL.*** These are migration actions that you can perform only after you've IPLed z/OS V1R13. You need a running z/OS V1R13 system to perform these actions. An example is issuing RACF commands related to new functions. Note that the term "first IPL" does not mean that you have to perform these actions after the very first IPL, but rather that you need z/OS V1R13 to be active to perform the task. You might perform the task quite a while after the first IPL.

Icons used in this presentation:

🔥 means that you shouldn't overlook this migration action.

mig ✓ means that an IBM Health Check (using the IBM Health Checker for z/OS function) can help you with this migration action.



Elements with Migration Actions for z/OS R13

► **Documented in *z/OS V1R13 Migration* (GA22-7499-19)**

- *For complete migration tasks for z/OS R13, see this book!*
  - ***Planned: from R11 to R13, and R12 to R13, customized books on z/OS Installation and Migration webpage.***

► ***From z/OS R11*** **to z/OS R13:**

➢ **BCP**
➢ **Communications Server**
➢ **Crypto Services – ICSF, SSL, OCSF, and PKI**
➢ **DFSMS**
▪ **DFSORT**
➢ **Distributed File Service – zFS**
➢ **Infoprint Server**
➢ **JES2**
➢ **JES3**
➢ **Language Environment**
▪ **RMF**
▪ **SDSF**
▪ **Security Server (RACF)**
▪ **SMP/E (mentioned in Part 1)**
▪ **TSO/E**
▪ **XL C/C++**
➢ **z/OS UNIX**

➢ ***means that some of that element's migration actions will be discussed in these presentations***

▪ ***means that you need to look for that element's migration actions in the z/OS R13 Migration book***

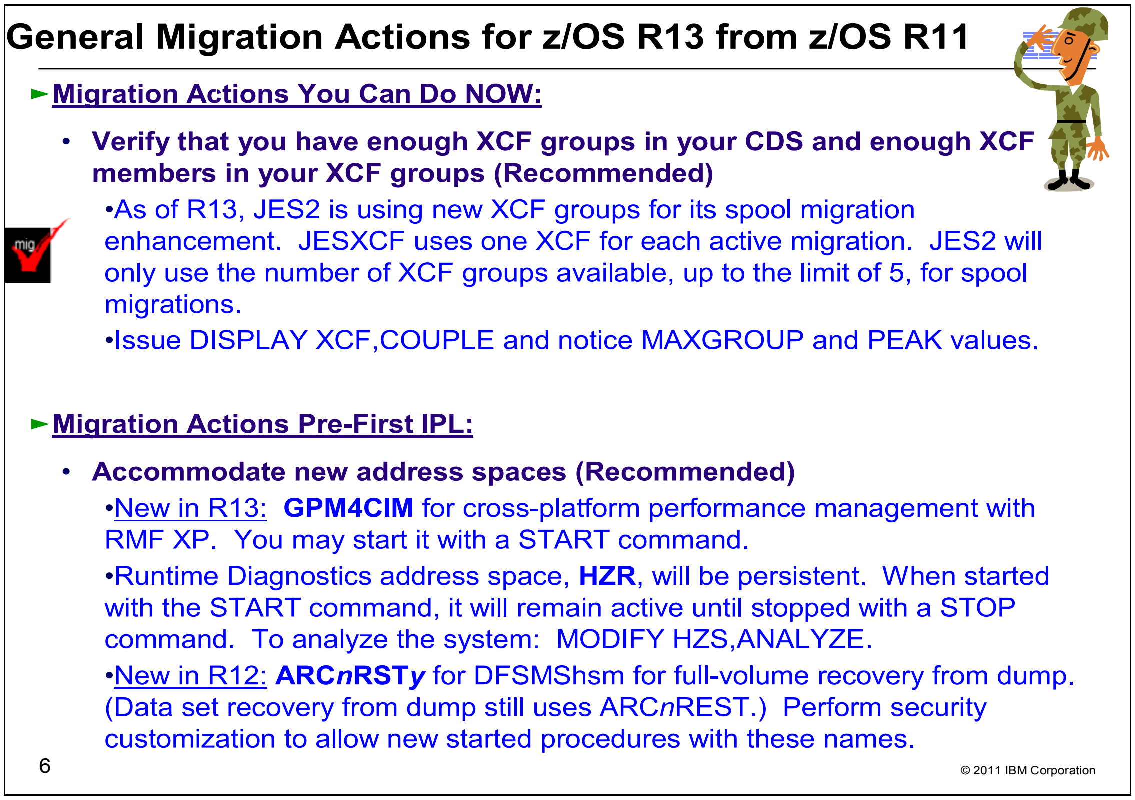
5

© 2011 IBM Corporation

Migration Actions for Elements Between z/OS R11 and z/OS R13

When migrating from z/OS R11 to z/OS R13, the specified elements in the slide above have required migration actions. Refer to *z/OS V1R13 Migration* for complete information on the required migration actions for all elements. Some migration actions for selected elements follow in this presentation.

If you are migrating from z/OS R12 to R13, there are fewer migration actions that are applicable. Use the customized *z/OS V1R13.0 Migration - From z/OS V1R12.0 to z/OS V1R13.0* book to see just the migration actions appropriate for this path, which is planned to be available. The customized books can be found on this webpage: http://www.ibm.com/systems/z/os/zos/installation/ .



General Migration Actions for z/OS R13 from z/OS R11

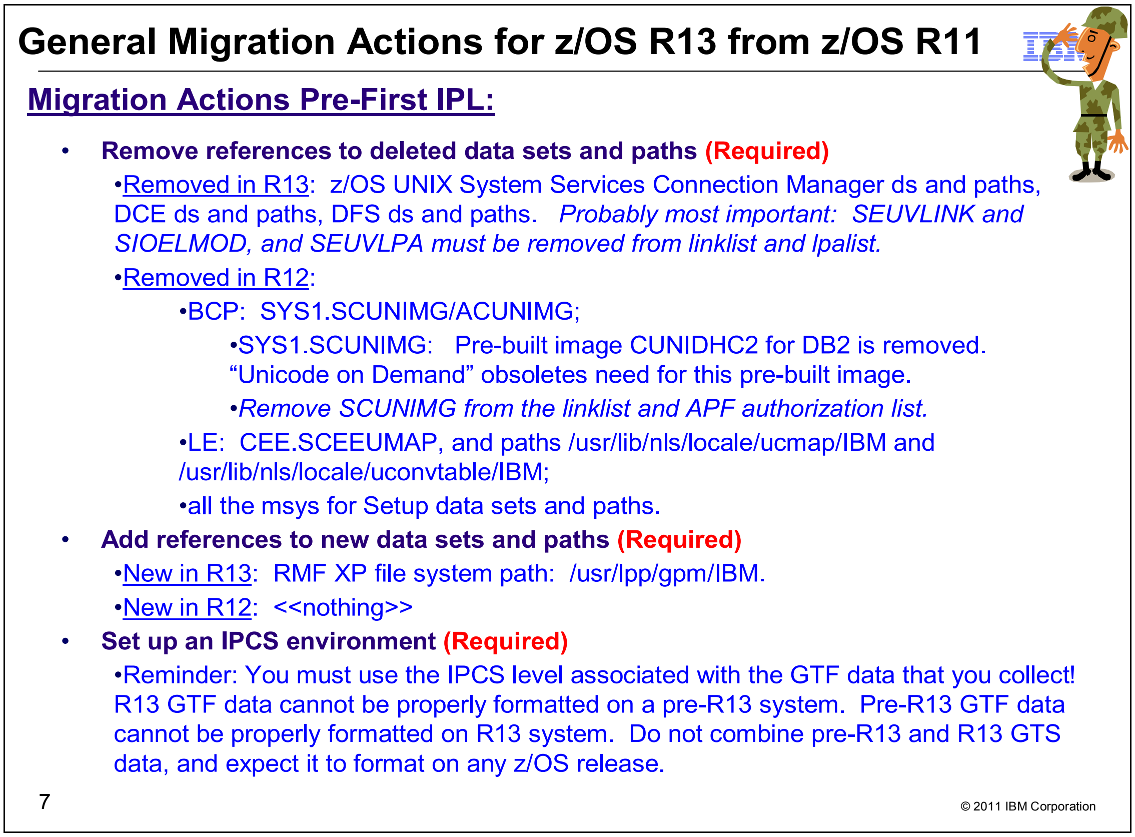
►**Migration Actions You Can Do NOW:**

- **Verify that you have enough XCF groups in your CDS and enough XCF members in your XCF groups (Recommended)**
  - •As of R13, JES2 is using new XCF groups for its spool migration enhancement. JESXCF uses one XCF for each active migration. JES2 will only use the number of XCF groups available, up to the limit of 5, for spool migrations.
  - •Issue DISPLAY XCF,COUPLE and notice MAXGROUP and PEAK values.

►**Migration Actions Pre-First IPL:**

- **Accommodate new address spaces (Recommended)**
  - •New in R13: **GPM4CIM** for cross-platform performance management with RMF XP. You may start it with a START command.
  - •Runtime Diagnostics address space, **HZR**, will be persistent. When started with the START command, it will remain active until stopped with a STOP command. To analyze the system: MODIFY HZS,ANALYZE.
  - •New in R12: ***ARCnRSTy*** for DFSMShsm for full-volume recovery from dump. (Data set recovery from dump still uses ARC*n*REST.) Perform security customization to allow new started procedures with these names.

6



General Migration Actions for z/OS R13 from z/OS R11

Migration Actions Pre-First IPL:

- **Remove references to deleted data sets and paths (Required)**
  - •Removed in R13: z/OS UNIX System Services Connection Manager ds and paths, DCE ds and paths, DFS ds and paths. *Probably most important: SEUVLINK and SIOELMOD, and SEUVLPA must be removed from linklist and lpalist.*
  - •Removed in R12:
    - •BCP: SYS1.SCUNIMG/ACUNIMG;
      - •SYS1.SCUNIMG: Pre-built image CUNIDHC2 for DB2 is removed. “Unicode on Demand” obsoletes need for this pre-built image.
      - •*Remove SCUNIMG from the linklist and APF authorization list.*
    - •LE: CEE.SCEEUMAP, and paths /usr/lib/nls/locale/ucmap/IBM and /usr/lib/nls/locale/uconvtable/IBM;
    - •all the msys for Setup data sets and paths.
- **Add references to new data sets and paths (Required)**
  - •New in R13: RMF XP file system path: /usr/lpp/gpm/IBM.
  - •New in R12: <<nothing>>
- **Set up an IPCS environment (Required)**
  - •Reminder: You must use the IPCS level associated with the GTF data that you collect! R13 GTF data cannot be properly formatted on a pre-R13 system. Pre-R13 GTF data cannot be properly formatted on R13 system. Do not combine pre-R13 and R13 GTS data, and expect it to format on any z/OS release.

7

General Migration Actions Between z/OS V1 R11 and z/OS V1 R13

These migration actions were taken from *z/OS R13 Migration.* Some descriptions and actions have been shortened for inclusion in this presentation. For the complete descriptions and actions, refer to *z/OS R13 Migration.*

General Migration Actions You Can Do Now

Install coexistence and fallback PTFs (Required)

Migration action: Install coexistence and fallback PTFs on your systems to allow those systems to coexist with z/OS V1R13 systems during your migration, and allow backout from z/OS V1R13 if necessary. Use the SMP/E V3R5 REPORT MISSINGFIX command in conjunction with the FIXCAT type of HOLDDATA as follows:

1. Acquire and RECEIVE the latest HOLDDATA onto your pre-z/OS V1R13 systems. Use your normal service acquisition portals or download the HOLDDATA directly from http://service.software.ibm.com/holdata/390holddata.html. Ensure you select **Full** from the Download NOW column to receive the FIXCAT HOLDDATA, as the other files do not contain FIXCATs.
2. Run the SMP/E REPORT MISSINGFIX command on your pre-z/OS V1R13 systems and specify a Fix Category (FIXCAT) value of **"IBM.Coexistence.z/OS.V1R13**". The report will identify any missing coexistence and fallback PTFs for that system. For complete information about the REPORT MISSINGFIX command, see *SMP/E Commands*.
3. Periodically, you might want to acquire the latest HOLDDATA and rerun the REPORT MISSINGFIX command to find out if there are any new coexistence and fallback PTFs.

Note that it is no longer possible to do z/OS coexistence verification with the EPSPT tool.

Add or change volumes to keep your z/OS root file system in a single data set (Recommended)

Recommended for ease of management if your z/OS root (or subsystem) file system resides on a 3390-3 volume (or another DASD volume that is close to the 3390-3 limit of 3339 cylinders).

Because of enhancements and PTF service, the z/OS root file system (or "version root file system") and subsystem file systems continue to grow in size from release to release. As of z/OS V1R13, the size of the z/OS root file system is approximately 3293 3390 cylinders when HFS, and approximately 3309 3390 cylinders when zFS. This is closely approaching the 3390-3 size limit of 3339 cylinders. It is advisable to have the z/OS root file system and subsystem file systems within a single data set for ease of management.

Migration action: To keep the z/OS root file system in a single data set, do one of the following:

- Move your z/OS root file system to a larger DASD volume geometry.
- Use multiple volumes for the z/OS root file system data set.

If your z/OS root data set cannot fit on the volume or volumes you have defined for it, divide the z/OS root, with the smaller file systems being managed together.

Remember that all systems to which you deploy the z/OS root file system need sufficient DASD space to hold the z/OS root.

Tips:

- File systems for subsystems and products other than the z/OS product itself might also increase in size. When examining the volume for how much space your z/OS file system is using, check other product file system sizes too.
- Use IBM Health Checker for z/OS check CHECK(IBMUSS,ZOSMIGREC_ROOT_FS_SIZE) to determine whether a volume has enough space for the z/OS root file system.
- This capability is also available in z/OS V1R9 with APAR OA28684 (PTF UA49361) and z/OS V1R10 with APAR OA28684 (PTF UA49362).

Use SOFTCAP to identify the effect of capacity changes (Recommended)

Not required, but is recommended to help in assessing processor capacity and available resources when migrating to new software levels, and when migrating to z/Architecture.

Migration action:

- Download SoftCap from one of the following Web sites:
  - Customers: http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS268

- Business partners: http://partners.boulder.ibm.com/src/atsmastr.nsf/Web/Techdocs. Note that this requires an ID on PartnerWorld®.Run SoftCap to determine your expected increase in CPU utilization (if any) and to identify your storage requirements, such as how much storage is needed to IPL.

Reference information: *SoftCap User's Guide*, which is provided with the tool.

Upgrade Windows 2000, 95, 98, and NT clients (Recommended)

Recommended because z/OS does not support service for client operating systems whose service is withdrawn by the operating system manufacturer.

z/OS no longer supports service for client operating systems whose service is withdrawn by the operating system manufacturer. As a result, IBM no longer supports service for clients running Windows 2000, Windows 95, Windows 98, or Windows NT Workstation 4.xx.

Migration action: Use a supported follow-on to Windows 2000, Windows 95, Windows 98, or Windows NT Workstation 4.xx.

Reference information: For client software supported with z/OS, see *z/OS Planning for Installation*.

Verify that you have enough XCF groups in your CDS and enough XCF members in your XCF groups (Recommended)

Recommended to ensure you have an adequate number of XCF groups and members formatted in your sysplex couple data sets.

Starting with z/OS V1R13, JES2 is using new XCF groups for its spool migration enhancement. JES spool migration utilizes tasks on all members of a MAS to manage the migration of a spool volume's data and the access to that migrating or migrated data. These various tasks communicate using messages sent through JESXCF services. The JESXCF services utilize one XCF group for each active migration to identify what messages are for which active migration. XCF groups are a limited system resource, so JES2 limits the number of concurrent active migrations to five. If you plan to perform spool migrations, verify that you have up to five XCF groups available if you intend to have up to five spool migrations active at any given time. JES2 will only utilize the number of XCF groups available, up to five, for spool migrations.

Migration action:

1. Issue the DISPLAY XCF,COUPLE command on your current system. Notice the values of MAXGROUP and PEAK for your sysplex couple data sets. These values show you the maximum number of XCF groups that the couple data sets can support, and the peak number of XCF groups ever in use in the sysplex. Also notice the values of MAXMEMBER and PEAK for your sysplex couple data sets. These values show you the maximum number of members that the couple data set can support in one group, and the greatest number of members ever in use in the largest group in the sysplex.
2. If your peak member value is close to the maximum member value, you might want to reformat your sysplex couple data sets to support a larger maximum number of members to be used by any one group.

Tip: Use IBM Health Checker for z/OS check XCF_SYSPLEX_CDS_CAPACITY, which checks the adequacy of the number of groups, members, and systems for which a sysplex CDS is formatted.

General Migration Actions Pre-First IPL

Set up your IPCS environment (Required)

The version and release level of IPCS must match the level of the system that produced the dump or trace. You must use the z/OS MVS libraries of IPCS code, for example, to analyze a dump or trace produced by an z/OS MVS system.

Migration action:

- z/OS V1R13 GTF data cannot be properly formatted on a pre-z/OS V1R13 system. If you attempt to format z/OS V1R13 GTF data with IPCS on a pre-z/OS V1R13 system, the fields will be unusable. Conversely, pre-z/OS V1R13 GTF data cannot be properly formatted on a z/OS V1R13 system. You must use z/OS V1R13 IPCS for z/OS V1R13 GTF data and pre-z/OS V1R13 IPCS for pre-z/OS V1R13 GTF data. You cannot merge pre-z/OS V1R13 GTF data with z/OS V1R13 GTF data and expect to format it on any z/OS release. Plan on using the IPCS level associated with the GTF data that you collect.
- Set up an IPCS environment. For guidance, use the documents listed in the reference information below. During setup, ensure that your logon procedure points to the target system's level of IPCS data sets, which are shown in *z/OS Migration.*

Reference information: For more information about IPCS, see *z/OS MVS IPCS Customization*. For more information about the correct logon procedure updates, see the *z/OS Program Directory*. For information about setting up the JES2 IPCS environment, see *z/OS JES2 Diagnosis*. For information about setting up the JES3 IPCS environment, see *z/OS JES3 Diagnosis.*

Use IBM-supplied PARMLIB and PROCLIB (Required)

Migration action: For parmlib, add the data set pointed to by the z/OS V1R13 PARMLIB DDDEF to your parmlib concatenation. The data set should generally be added last in the concatenation, and you should make sure that the other data sets in the concatenation don't have members with the same names as IBM-supplied members. If you place the data set on the system residence volume and use an indirect catalog entry, future migrations won't require this particular migration step.

- For proclib:
  - Ensure that the default proclib members have been copied to your default proclib to pick up the new and changed members.
  - Update individual sample members provided and ensure they are accessible to the system, as shown in the table of proclib member updates in *z/OS Program Directory*.
  - Ensure that the procedure libraries listed in the table of libraries to be added to the proclib concatenation in *z/OS Program Directory* have been placed in the necessary procedure library concatenations and are available to the system.

Reference information: For lists of parmlib and proclib members that are shipped, see *z/OS Program Directory*.

Migrate /etc and /var system control files (Required)

Migration action: The /etc and /var directories contain system control files: the /etc directory contains customization data that you maintain and the /var directory contains customization data that IBM maintains. During installation, subdirectories of /etc and /var are created. If you install z/OS using ServerPac, some files are loaded into /etc and /var due to the customization performed in ServerPac. You have to merge the files in /etc and /var with those on your previous system. If you install z/OS using CBPDO, you should copy the files from your old system to the z/OS V1R13 /etc and /var subdirectories.

Copy files from your old system to the z/OS V1R13 /etc and /var subdirectories, and then modify the files as necessary to reflect z/OS V1R13 requirements. If you have other files under your existing /var directory, then you will have to merge the old and new files under /var. The easiest way to do this is to create a copy of your current /var HFS and then copy the new /var files into the copy.

The following z/OS V1R13 elements and features use /etc:

- BCP (Predictive Failure Analysis), as of z/OS V1R12
- CIM
- Communications Server – IP
- Cryptographic Services – PKI Services and System SSL
- DCE Base Services
- DFSMSrmm
- Distributed File Service. The SMB server uses /etc/dfs.
- IBM HTTP Server
- IBM Tivoli Directory Server – uses /etc/ldap.
- Infoprint Server uses /etc/Printsrv.
- Library Server
- z/OS UNIX System Services

The following z/OS V1R13 elements and features use /var:

- Cryptographic Services – OCSF
- DFSMSrmm
- IBM Tivoli Directory Server – uses /var/ldap.
- Infoprint Server
- Integrated Security Services - Network Authentication Service uses /var/skrb.
- z/OS UNIX System Services

Reference information: For information about copying your existing /etc and /var directories, see *z/OS Migration*.

Verify that virtual storage (MEMLIMIT) is set properly (Required)

Migration action: Determine how much virtual storage use to allow above the 2 GB bar. While there is no practical limit to the number of virtual addresses an address space can request above the bar, the system can limit the amount of virtual storage above the bar that an address space is allowed to use. The amount of virtual storage above the bar is determined as follows. The MEMLIMIT parameter in parmlib member SMFPRMxx sets the default system-wide limit, which defaults to 2 GB as of z/OS V1R10 (and zero prior to z/OS V1R10). However, the system-wide default MEMLIMIT can be overridden by specifying REGION=0M or MEMLIMIT on JOB or EXEC statements in JCL. To set a limit on the use of virtual storage above the bar, use the SMF exit IEFUSI. For more information, see "Limiting the use of memory objects" in *z/OS MVS Programming: Extended Addressability Guide*.

If you want to control the use of virtual storage above the 2 GB bar, do one or more of the following:

- For MEMLIMIT prior to z/OS V1R10, you must specify a nonzero MEMLIMIT in an active SMFPRM*xx* member of parmlib to establish a system default other than zero for available virtual storage above 2 GB. (The default MEMLIMIT is zero prior to z/OS V1R10.) As of z/OS V1R10, the MEMLIMIT default is 2 GB. If this 2 GB default value is acceptable to you, no change to SMFPRM*xx* is necessary.
- You can specify MEMLIMIT explicitly in JCL to override the system default that was set (or allowed to default) in SMFPRM*xx*.
- You can specify REGION=0M on the job statement in JCL to implicitly set MEMLIMIT to NOLIMIT, which also overrides the system default (from SMFPRM*xx*).
- You can use IEFUSI both to establish a system default MEMLIMIT for different classes of work (for example, job, TSO, STC) and limit the amount of virtual storage that can be used above the bar, provided that an explicit or implicit nonzero MEMLIMIT is in effect from JCL or SMFPRM*xx*. As of z/OS V1R10, keyword HONORIEFUSIREGION | NOHONORIEFUSIREGION is available in SCHED*xx* to identify if the region and MEMLIMIT settings specified through or otherwise affected by the IEFUSI exit are to take effect for a program.

Tip: Use IBM Health Checker for z/OS to help determine whether your virtual storage limits are set properly. The check RSM_MEMLIMIT checks the current setting for the MEMLIMIT parameter in SMFPRM*xx*, which affects the amount of virtual storage above the 2 GB bar that is available to jobs. This check verifies that a nonzero MEMLIMIT value is in use.

Reference information: Information about how to evaluate the central storage configuration can be found in the Washington Systems Center white paper *z/OS Performance: Managing Processor Storage in a 64-bit Environment - V1* at http://www.ibm.com/support/techdocs (Search for "WP100269".)

Back virtual storage with real and auxiliary storage (Required)

Migration action: As you exploit additional virtual storage by defining additional address spaces or by exploiting memory objects, ensure that you have defined sufficient real and auxiliary storage. Review real storage concentration indicators via an RMF report to evaluate if additional real or auxiliary storage is needed:

- Check UIC and average available frames.
- Check demand page rates.
- Check the percentage of auxiliary slots in use.

Reference information: For more information about memory objects, see *z/OS MVS Programming: Extended Addressability Guide* and Washington Systems Center flash 10165 at http://www.ibm.com/support/techdocs. (Search for "flash10165".)

Remove references to deleted data sets and path (Required)

Migration action: Using the table in *z/OS Migration, "*Data sets and paths deleted from z/OS V1R13, and R12" as a guide, remove references to data sets and paths that no longer exist. Remove the references from the following places:

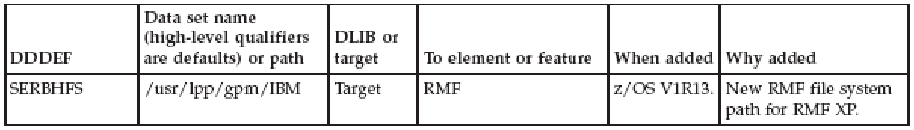
- Parmlib
- Proclib
- Logon procedures
- Catalogs
- Security definitions, including program control definitions
- DFSMS ACS routines
- /etc/profile
- SMP/E DDDEF entry

- Backup and recovery procedures, as well as any references to them in the table, the high-level qualifiers in the data set names are the default qualifiers.

Note: Ensure that references to the DCE target library EUV.SEUVLINK and DFS target library IOE.SIOELMOD have been removed from your LNKLST concatenation. Ensure that any reference to DCE target library EUV.SEUVLPA has been removed from the LPALST concatenation.
Note: Do not remove any data sets, paths, or references that are needed by earlier-level systems until those systems no longer need them, and you are sure you won't need them for fallback.
Reference information: *z/OS Migration* contains the list of all removed data sets and paths in z/OS R12 and R13.

Add references to new data sets (Required)

Migration action: The only data set of path that has been added to z/OS R12 or R13 was the following:



DDDEF	Data set name (high-level qualifiers are defaults) or path	DLIB or target	To element or feature	When added	Why added
SERBHFS	/usr/lpp/gpm/IBM	Target	RMF	z/OS V1R13.	New RMF file system path for RMF XP.

Accommodate new address spaces (Recommended)

Not required, but recommended to keep interested personnel aware of changes in the system and to ensure that your MAXUSER value in parmlib member IEASYSxx is adequate.
There are two new address spaces in z/OS V1R13.

- GPM4CIM is an address space to be used for cross-platform performance management with RMF XP. You can start it by means of procedure SYS1.PROCLIB(GPM4CIM) from the console as started task with the following command: s gpm4cim[.identifier],os=A|X|Z
  Since you can run multiple GPM4CIM instances simultaneously, it is recommended to assign an identifier that you can use for subsequent STOP or MODIFY commands. You may already have created the userID GPMSERVE as owner of the GPMSERVE procedure. The GPM4CIM started task can be assigned to the same userID with the following command: RDEFINE STARTED GPM4CIM.* STDATA(USER(GPMSERVE) TRUSTED(YES)). For more information, refer to *z/OS RMF User's Guide*.
- The Runtime Diagnostics address space HZR will be a persistent address space. When the HZR address space is started with the START command **S HZR,SUB=MSTR**, it will remain active until stopped with the STOP command **P HZR**. To analyze a system, enter the **MODIFY HZR,ANALYZE** command. See migration action "Start Runtime Diagnostics at system initialization" in *z/OS R13 Migration* for more information.

There is one new address space in z/OS V1R12. ARC*n*RST*y* is the address space identifier for full-volume recovery from dump, where *n* is the DFSMShsm host ID and *y* is the instance of the DFSMSdss started task (a number from 1 to 4). Data set recovery from dump will still use ARC*n*REST. See "DFSMShsm: Configure your security system to permit started procedures using new address space identifier" later in this presentation for more information.

The MAXUSER value in parmlib member IEASYS*xx* specifies a value that the system uses to limit the number of jobs and started tasks that can run concurrently during a given IPL. You might want to increase your MAXUSER value to take new address spaces into account. (A modest overspecification of MAXUSER should not hurt system performance. The number of total address spaces is the sum of M/S, TS USERS, SYSAS, and INITS. If you change your MAXUSER value, you must re-IPL to make the change effective.)

Rework and install user modifications (Required-IF)

Required if you have made any user modifications that necessitate changes.
Migration action: Use the z/OS SMP/E Planning Migration Assistant to help determine which user modifications need to be reworked and which just have to be reinstalled. The Top or New Intermediate Product Migration Changes Report uses data found on your system, combined with IBM-supplied information from the Software Information Base, to show you the current levels of products available as well as product migration and functional changes using a comparison of FMIDs. You can use this report to determine the product migration impacts by reviewing the "changed" FMIDs. This can help you assess how many user modifications have to be reworked if you issued the LIST SYSMOD USERMOD FORFMID (listing the "changed" FMIDs) command. All other user modifications can be reinstalled without having to be reworked.
Note: IBM recommends using exit routines for any user modifications where possible, and installing the exit routines with SMP/E. By using SMP/E, it is easier to bring forward desired modifications to the z/OS release you are installing.

Several elements and features have their default options set by assembling and link editing one or more modules. These include:

- XL C/C++
- DFSORT. Consider using ICEPRM*xx* parmlib members, introduced in z/OS V1R10, to eliminate the assembler language installation option modules
- HLASM
- ISPF (specifically, the ISPF configuration table)
- Language Environment. Investigate using CEEROPT, which can be used to specify run-time options for CICS, IMS LRR, and other LRR users. Even better, consider using the function added in z/OS R7 to eliminate your assembler language run-time option modules in CEEPRMxx parmlib member!
- SDSF (ISFPARMS customization). Use dynamic statements for ISFPARMS to avoid reassembly!

Reconnect subsystems and non-IBM products (Required-IF)

Required if you use any ISV products and need to reconnect them after performing a ServerPac installation, or if you intend to use any subsystems with your z/OS system.
Migration action: Follow the instructions for each ISV product that you use to reconnect it to your z/OS V1R13 ServerPac.

Ensure that any required service is installed prior to using the subsystem with the new z/OS V1R13 system, as well as any required SVCs, system modifications, parmlib setup, and proclib setup. Follow the instructions for the subsystem that you need to reconnect.
Reference information: For a list of independent software vendors (ISVs) that support z/OS, as well as announcements, testimonials, and other information, see http://www.ibm.com/eserver/zseries/solutions/s390da/. For a directory of ISV products that support z/OS, see the Global Solutions Directory at http://www.ibm.com/software/solutions/isv.

Update operational and other procedures (Required)

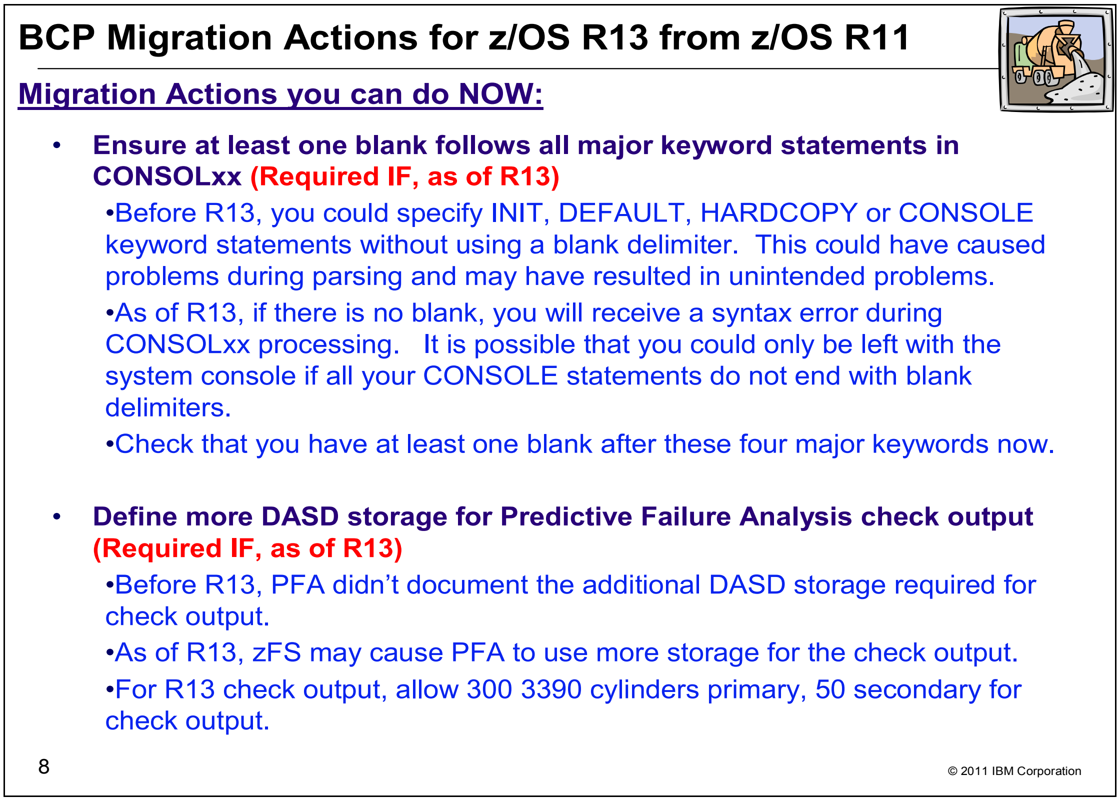
Migration action: Review your operation, automation, administration, security, backup, and recovery procedures, and make any necessary changes depending on how you installed and which functions you plan to exploit. Some possible changes are:

- Allowing applicable users access to new high-level qualifiers that you may have. There are no new default high-level qualifiers introduced since z/OS R9.
- Updating and testing your backup and recovery procedures to accommodate the new target system.
- Updating and testing any disaster recovery procedures.
- Updating and testing any automation procedures to take advantage of new functions.
- Updating security system definitions, such as defining new users and resources, permitting users to use new resources, and defining new profiles in the RACF FACILITY class.

Reference information: For information about the new functions incorporated into z/OS V1R13, see *z/OS Introduction and Release Guide*.

General Migration Actions Post-First IPL

<none>



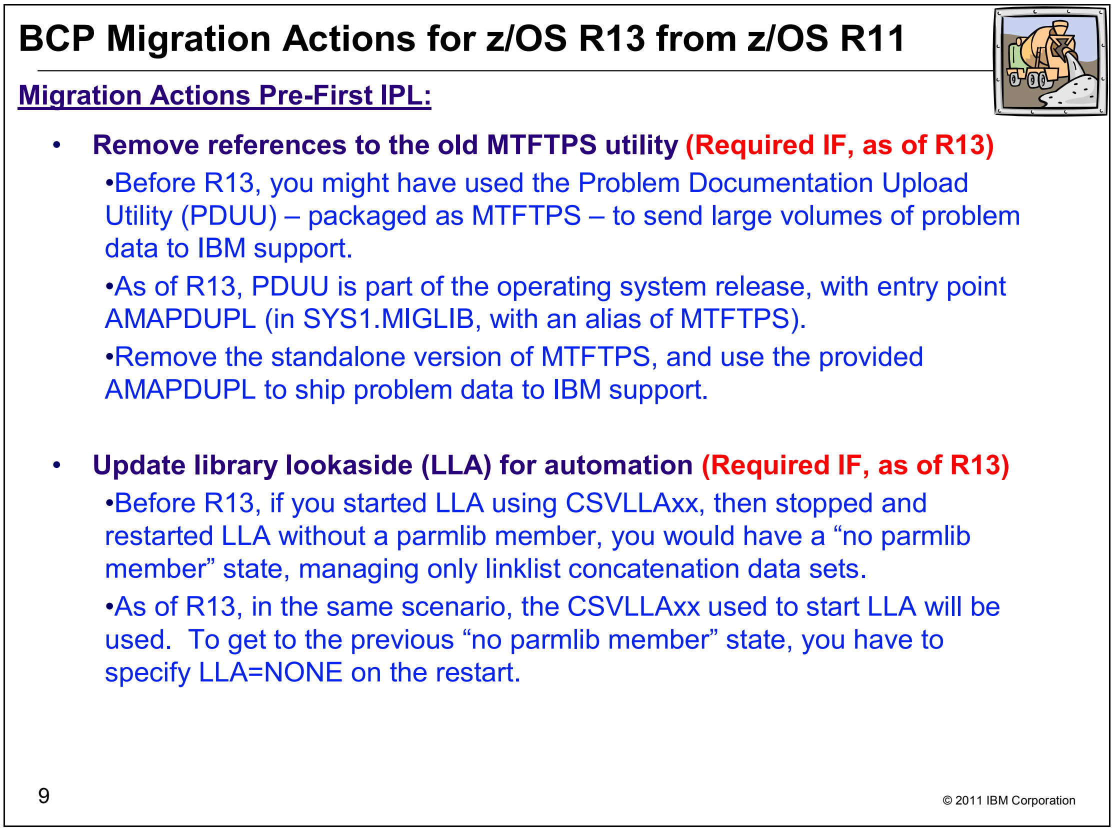
BCP Migration Actions for z/OS R13 from z/OS R11

Migration Actions you can do NOW:

- **Ensure at least one blank follows all major keyword statements in CONSOLxx (Required IF, as of R13)**
  - Before R13, you could specify INIT, DEFAULT, HARDCOPY or CONSOLE keyword statements without using a blank delimiter. This could have caused problems during parsing and may have resulted in unintended problems.
  - As of R13, if there is no blank, you will receive a syntax error during CONSOLxx processing. It is possible that you could only be left with the system console if all your CONSOLE statements do not end with blank delimiters.
  - Check that you have at least one blank after these four major keywords now.
- **Define more DASD storage for Predictive Failure Analysis check output (Required IF, as of R13)**
  - Before R13, PFA didn't document the additional DASD storage required for check output.
  - As of R13, zFS may cause PFA to use more storage for the check output.
  - For R13 check output, allow 300 3390 cylinders primary, 50 secondary for check output.

8

© 2011 IBM Corporation



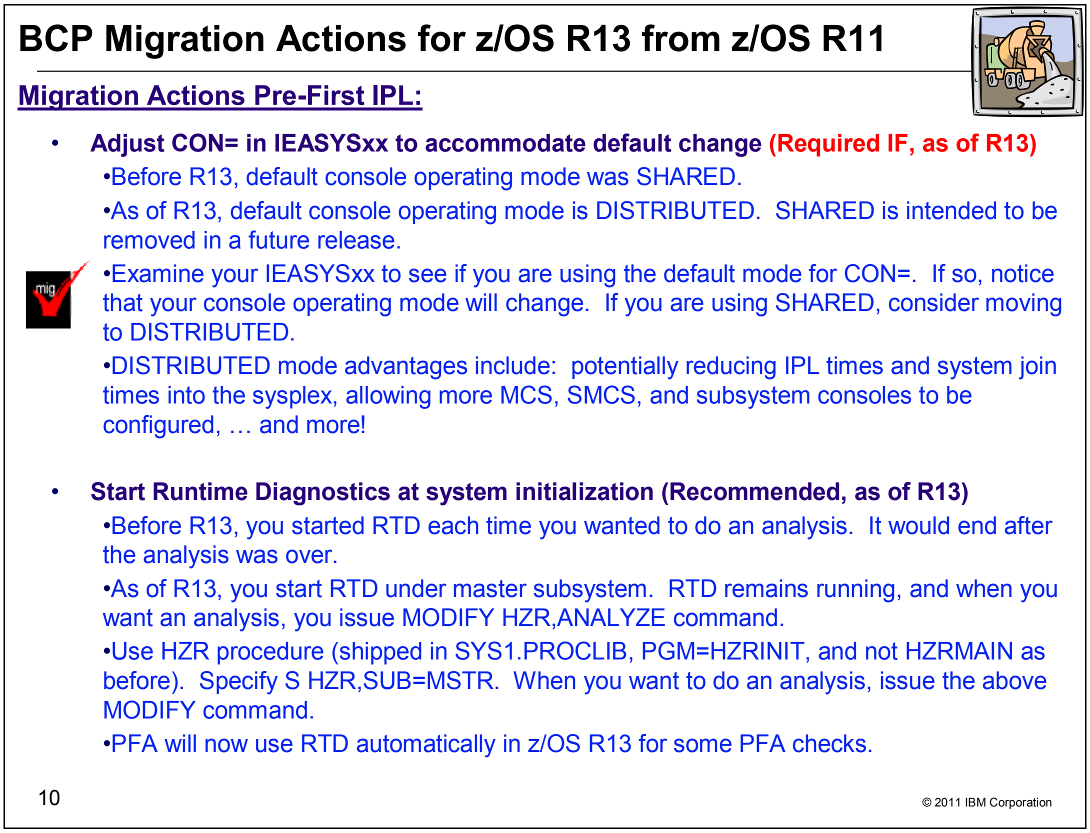
BCP Migration Actions for z/OS R13 from z/OS R11

Migration Actions Pre-First IPL:

- **Remove references to the old MTFTPS utility (Required IF, as of R13)**
  - Before R13, you might have used the Problem Documentation Upload Utility (PDUU) – packaged as MTFTPS – to send large volumes of problem data to IBM support.
  - As of R13, PDUU is part of the operating system release, with entry point AMAPDUPL (in SYS1.MIGLIB, with an alias of MTFTPS).
  - Remove the standalone version of MTFTPS, and use the provided AMAPDUPL to ship problem data to IBM support.
- **Update library lookaside (LLA) for automation (Required IF, as of R13)**
  - Before R13, if you started LLA using CSVLLAxx, then stopped and restarted LLA without a parmlib member, you would have a "no parmlib member" state, managing only linklist concatenation data sets.
  - As of R13, in the same scenario, the CSVLLAxx used to start LLA will be used. To get to the previous "no parmlib member" state, you have to specify LLA=NONE on the restart.

9

© 2011 IBM Corporation

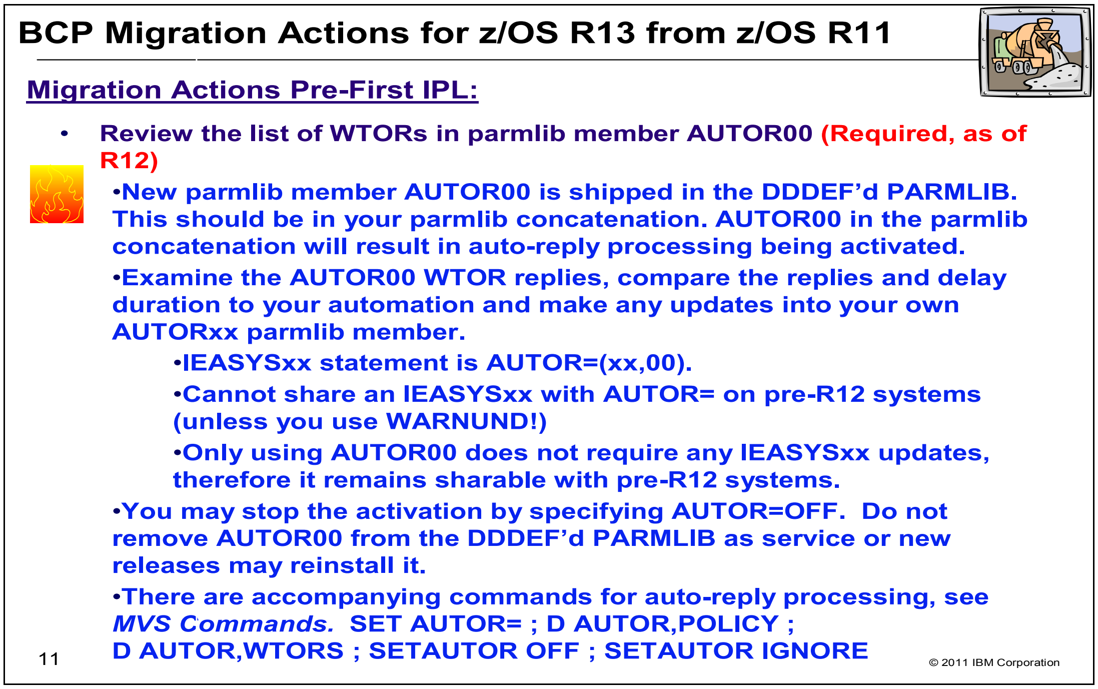


BCP Migration Actions for z/OS R13 from z/OS R11

Migration Actions Pre-First IPL:

- **Adjust CON= in IEASYSxx to accommodate default change (Required IF, as of R13)**
  - •Before R13, default console operating mode was SHARED.
  - •As of R13, default console operating mode is DISTRIBUTED. SHARED is intended to be removed in a future release.
  - mig •Examine your IEASYSxx to see if you are using the default mode for CON=. If so, notice that your console operating mode will change. If you are using SHARED, consider moving to DISTRIBUTED.
  - •DISTRIBUTED mode advantages include: potentially reducing IPL times and system join times into the sysplex, allowing more MCS, SMCS, and subsystem consoles to be configured, … and more!

- **Start Runtime Diagnostics at system initialization (Recommended, as of R13)**
  - •Before R13, you started RTD each time you wanted to do an analysis. It would end after the analysis was over.
  - •As of R13, you start RTD under master subsystem. RTD remains running, and when you want an analysis, you issue MODIFY HZR,ANALYZE command.
  - •Use HZR procedure (shipped in SYS1.PROCLIB, PGM=HZRINIT, and not HZRMAIN as before). Specify S HZR,SUB=MSTR. When you want to do an analysis, issue the above MODIFY command.
  - •PFA will now use RTD automatically in z/OS R13 for some PFA checks.



BCP Migration Actions for z/OS R13 from z/OS R11

Migration Actions Pre-First IPL:

- **Review the list of WTORs in parmlib member AUTOR00 (Required, as of R12)**
  - **•New parmlib member AUTOR00 is shipped in the DDDEF'd PARMLIB. This should be in your parmlib concatenation. AUTOR00 in the parmlib concatenation will result in auto-reply processing being activated.**
  - **•Examine the AUTOR00 WTOR replies, compare the replies and delay duration to your automation and make any updates into your own AUTORxx parmlib member.**
    - **•IEASYSxx statement is AUTOR=(xx,00).**
    - **•Cannot share an IEASYSxx with AUTOR= on pre-R12 systems (unless you use WARNUND!)**
    - **•Only using AUTOR00 does not require any IEASYSxx updates, therefore it remains sharable with pre-R12 systems.**
  - **•You may stop the activation by specifying AUTOR=OFF. Do not remove AUTOR00 from the DDDEF'd PARMLIB as service or new releases may reinstall it.**
  - **•There are accompanying commands for auto-reply processing, see *MVS Commands.* SET AUTOR= ; D AUTOR,POLICY ; D AUTOR,WTORS ; SETAUTOR OFF ; SETAUTOR IGNORE**

BCP Migration Actions for z/OS R13 from z/OS R11

Migration Actions Pre-First IPL:

- Review the list of WTORs in parmlib member AUTOR00 (Required, as of R12)
  - •…continued…
  - •*Appendix B* in *MVS Planning: Operations* contains the AUTOR00 shipped parmlib member for you to review and plan for:

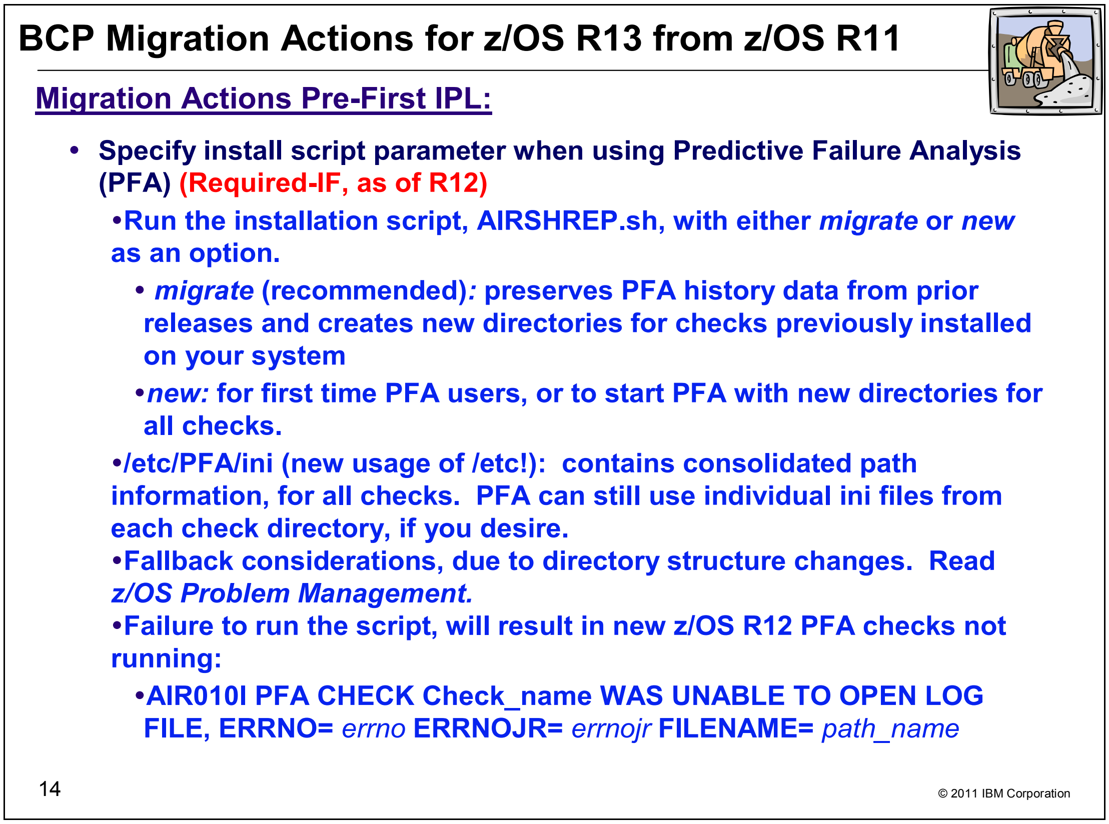
```
/*********************************************************************/
/* $HASP294 WAITING FOR RESERVE (VOL volser). REPLY 'CANCEL' TO     */
/* END WAIT                                                          */
/* Rule: 1 System Detected Problem during exec of op init action    */
Msgid(?HASP294) Delay(30S) Reply(CANCEL)
/*********************************************************************/
/*********************************************************************/
/* IEA029D {IEASVC|ALLOC|SCHED} PARMLIB MEMBER HAS AN UNBALANCED    */
/* COMMENT. REPLY YES TO CONTINUE IPL OR NO TO RESPECIFY            */
/* {SVC|ALLOC|SCHED} PARM                                            */
/* Rule: 5 Allow the system to continue to IPL                      */
Msgid(IEA029D) Delay(60S) Reply(YES)
/*********************************************************************/
…
```

The question mark "?" is supported, because some messages (e.g., JES2 messages) start with an installation-specified character (not always $). See *MVS Init and Tuning Ref* for more information.

BCP Migration Actions for z/OS R13 from z/OS R11

Migration Actions Pre-First IPL:

- Change default locations for LCCA or PCCA control blocks to retain 24-bit virtual storage location (Required-IF, as of R12)
  - •Before R12: if not specified in DIAGxx, default was in 24-bit virtual storage.
  - •As of R12: if not specified, default is now 31-bit virtual storage. For only one standard CP online during IPL, then the LCCA and PCCA of the IPL CPU will be 24-bit storage regardless of specification.
- Accommodate new REUSASID default (Required-IF, as of R12)
  - •In R9-R11: DIAGxx's REUSASID default was NO
  - •As of R12: REUSASID is YES.
  - •Verify programs can tolerate reusable ASIDs, and have no 0D3 abends.
- Use the new 16M default buffer size (BUFSIZE) for trace options with the CTIGRSxx member (Required IF, as of R12)
  - •Before R12, IBM-supplied CTIGRS00 parmlib member would result in BUFSIZE of 128K, maximum was 16MB, and the buffer was below the line private storage.
  - •As of R12, CTIGRS00 will now result in BUFSIZE of 16MB, maximum is 2047MB, and is above the bar private storage in its own memory object.
  - •If you specify your own CTIGRSxx member on the CTRACE option in the GRSCNFxx parmlib member, change the BUFSIZE in your CTIGRSxx to 16MB. Or you could just use the IBM-provided CTIGRS00.



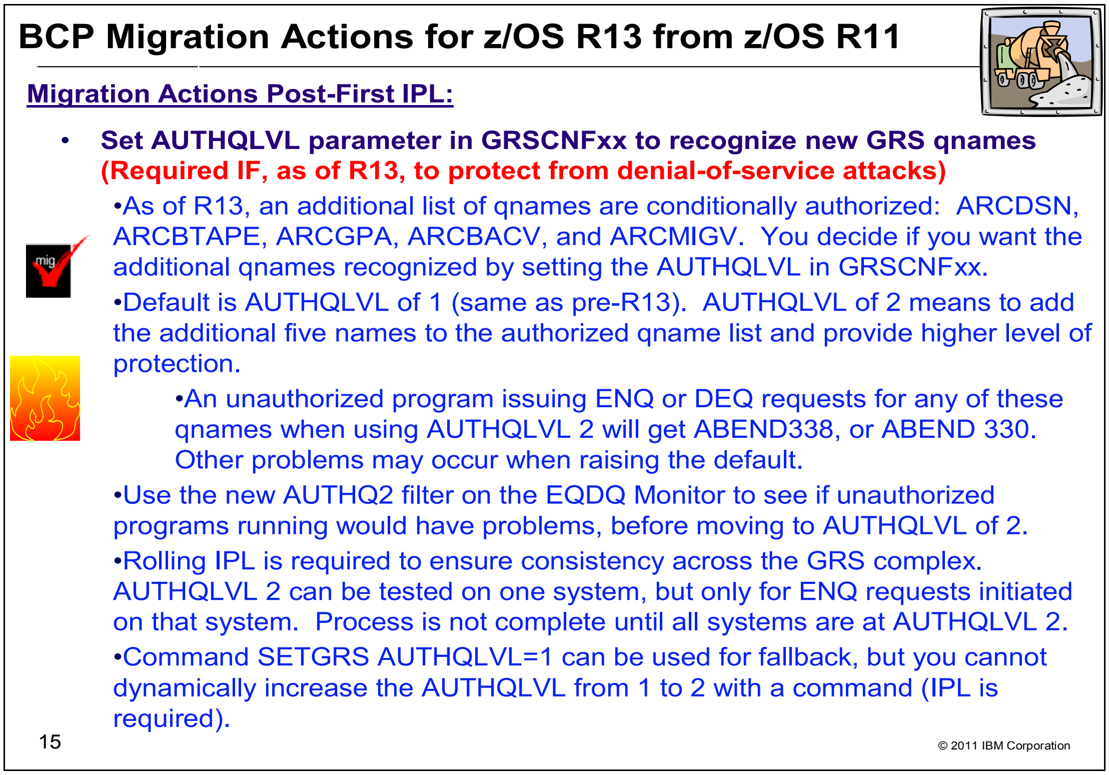
BCP Migration Actions for z/OS R13 from z/OS R11

Migration Actions Pre-First IPL:

- **Specify install script parameter when using Predictive Failure Analysis (PFA) (Required-IF, as of R12)**
  - **Run the installation script, AIRSHREP.sh, with either *migrate* or *new* as an option.**
    - ***migrate* (recommended)*:* preserves PFA history data from prior releases and creates new directories for checks previously installed on your system**
    - ***new:* for first time PFA users, or to start PFA with new directories for all checks.**
  - **/etc/PFA/ini (new usage of /etc!): contains consolidated path information, for all checks. PFA can still use individual ini files from each check directory, if you desire.**
  - **Fallback considerations, due to directory structure changes. Read *z/OS Problem Management.***
  - **Failure to run the script, will result in new z/OS R12 PFA checks not running:**
    - **AIR010I PFA CHECK Check_name WAS UNABLE TO OPEN LOG FILE, ERRNO=** *errno* **ERRNOJR=** *errnojr* **FILENAME=** *path_name*

14

© 2011 IBM Corporation



BCP Migration Actions for z/OS R13 from z/OS R11

Migration Actions Post-First IPL:

- **Set AUTHQLVL parameter in GRSCNFxx to recognize new GRS qnames (Required IF, as of R13, to protect from denial-of-service attacks)**
  - As of R13, an additional list of qnames are conditionally authorized: ARCDSN, ARCBTAPE, ARCGPA, ARCBACV, and ARCMIGV. You decide if you want the additional qnames recognized by setting the AUTHQLVL in GRSCNFxx.
  - Default is AUTHQLVL of 1 (same as pre-R13). AUTHQLVL of 2 means to add the additional five names to the authorized qname list and provide higher level of protection.
    - An unauthorized program issuing ENQ or DEQ requests for any of these qnames when using AUTHQLVL 2 will get ABEND338, or ABEND 330. Other problems may occur when raising the default.
  - Use the new AUTHQ2 filter on the EQDQ Monitor to see if unauthorized programs running would have problems, before moving to AUTHQLVL of 2.
  - Rolling IPL is required to ensure consistency across the GRS complex. AUTHQLVL 2 can be tested on one system, but only for ENQ requests initiated on that system. Process is not complete until all systems are at AUTHQLVL 2.
  - Command SETGRS AUTHQLVL=1 can be used for fallback, but you cannot dynamically increase the AUTHQLVL from 1 to 2 with a command (IPL is required).

15

© 2011 IBM Corporation

BCP Migration Actions Between z/OS V1 R11 and z/OS V1 R13

These migration actions were taken from *z/OS Migration*. Some descriptions and actions have been shortened for inclusion in this presentation. For the complete descriptions and actions, refer to *z/OS Migration.*

BCP Migration Actions You Can Do Now

Evaluate your stand-alone dump data set allocations and your IPCS processing of them (Recommended)

Not required, but recommended because of changes to stand-alone dump processing (that reorder dump records with the intent of recording more important data early), and especially recommended if you deploy any LPARs with significantly more main storage than previously used.

In z/OS V1R6, support was introduced for extended-format sequential data sets, a form of data set that is SMS-managed and can occupy more than 64 K tracks per volume. In z/OS V1R7, this support was supplemented with support for large format sequential data sets (DSNTYPE=LARGE), a form of data set that is essentially the same as conventional sequential data sets except that more than 64 K tracks may be spanned per volume. If your stand-alone dump data sets are spread over more volumes than you want, both types of support can help you gain better control over the number of volumes used for each stand-alone dump data set.

Migration action:

- Use multi-volume stand-alone dump data sets. Adjust the number of volumes and their separation to achieve tolerable stand-alone dump capture times.
- Use extended-format sequential data sets or large format sequential data sets. Copy their contents to an extended-format, compressed, striped data set using the IPCS COPYDUMP subcommand prior to analysis. Use the same or a larger striping factor than you used for your stand-alone dump data sets. Dump data sets to which stand-alone dump can write may be neither compressed nor striped, but both attributes are advantageous for the target of the copy operation. Starting with z/OS V1R12, stand-alone dump data sets can be placed in track-managed space as well as cylinder-managed space on Extended Address Volumes (EAV).
- Use a large CISIZE and striping for IPCS dump directories, and use blocking, striping, and compression for the stand-alone dump data set. Very large stand-alone dumps might require that you define your directory with the extended addressing attribute, allowing it to hold more than 4 GB.

Tips: Control interval sizes less than 24K have been shown to be more vulnerable to fragmentation when used as IPCS dump directories, and IPCS performance can be degraded when such fragmentation occurs. In this background, warning message BLS21110I will be issued and you might recreate the DDIR by using the CLIST BLSCDDIR. BLS21110I CISIZE(cisize) is less than 24K. It may degrade IPCS performance

Consider exploiting WARNUND for new IEASYS*xx* statements (Recommended)

Not required, but recommended to assist in sharing IEASYSxx members between z/OS V1R13, and pre-z/OS V1R13 systems, when new enhancements in z/OS V1R13 are to be exploited.

Starting in z/OS V1R13 (and rolled back to z/OS V1R12 and z/OS V1R11 in OA35929), you can specify the WARNUND statement in IEASYS*xx*. When used, this statement indicates that warning message IEA660I be issued when undefined statements are encountered, rather than prompting for a correct statement. Usage of WARNUND can be particularly useful when specifying new parmlib options in IEASYS*xx* (such as the new IXGCNF and IGGCAT system parameters which are introduced in z/OS V1R13), and allowing these new IEASYS*xx* specifications to be shared with pre-z/OS V1R13 systems.

Migration action:

- Install the PTF for APAR OA35929 on all pre-z/OS V1R13 systems.
- As you add new statements in IEASYS*xx* for functional exploitation and you wish to share those modified IEASYS*xx* members with pre-z/OS V1R13 systems, add **WARNUND** to the beginning of IEASYS00 as that will cover updates in all IEASYS*xx* members.

Define DASD storage for Predictive Failure Analysis (Required, as of R13)

Required, if you use PFA.

Before z/OS V1R13, Predictive Failure Analysis (PFA) did not document the requirement for additional DASD storage to accommodate check output. Starting with z/OS V1R13, *z/OS Problem Management* contains DASD requirements to ensure PFA has enough space to update and create files in the z/OS UNIX file system to store check output. In addition, because zFS no longer stores data in 1K fragments, zFS for z/OS V1R13 might need more DASD storage to store the same amount of data than was required in previous releases. For additional information about zFS requirements, see “zFS: Accommodate new DASD space requirements”, in *z/OS R13 Migration.*

Migration action: Define additional DASD storage for PFA. The total space for the PFA file system for each LPAR depends on the release of z/OS you are running.
z/OS V1R11 (HBB7760): 200 cylinders primary; 50 cylinders secondary on a 3390 device.
z/OS V1R12 (HBB7770): 200 cylinders primary; 50 cylinders secondary on a 3390 device.
z/OS V1R13 (HBB7780): 300 cylinders primary; 50 cylinders secondary on a 3390 device.

Migrate from SNMP to z/OS BCPii for communication to the HMC or SE (Recommended)

Not required, but recommended because the migration to BCPii will be required in a future release.
IBM intends for z/OS V1R13 to be the final release in which SNMP as supported protocol for the communication to the Hardware Management Console (HMC) or Support Element (SE) is available. If you are currently using SNMP for
the communication, it is recommended that you now migrate to BCPii. The migration includes enabling the communication through BCPii for the provisioning manager user and adding a new key to the Capacity Provisioning Manager parameter file.
Migration action:

- You can use the tracking facility to help with this migration action. In tracking facility output, look for violations that start with CPO-W:SNMP usage *domain name*, where *domain name* is replaced with the actual name of the affected domain. Exploit the z/OS tracking facility on z/OS V1R12 or z/OS V1R11 by installing the PTF for APAR OA35284. If you are using the tracking facility and have no instances of affected domains after starting Capacity Provisioning Manager, then this migration action is not applicable to you.
- Set up BCPii as described in *z/OS MVS Programming: Callable Services for High-Level Languages*.
- Define the required security profiles to allow the Capacity Provisioning Manager user to access the hardware information.
- Add the Topology.Protocol=INTERNAL key to the Capacity Provisioning Manager parameter file. Using the default values, the file is the member CPO.DOMAIN1.PARM(PARM).

Verify that at least one blank follows all major keyword statements (Required-IF, as of R13)

Required if you do not have at least one blank after any of the four major keywords INIT, DEFAULT, HARDCOPY, and CONSOLE.
Before z/OS V1R13, you could specify INIT, DEFAULT, HARDCOPY and CONSOLE keyword statements without using a blank delimiter. This can cause a problem if other keywords are misplaced or misspelled. For example, if INTIDS(Y) is misspelled as INITIDS(Y), the parser considers this an INIT statement. This could result in a console not being defined correctly, or even having a system with no consoles after initialization except the system console.

Starting with z/OS V1R13, if you do not have a blank character after the four major keywords (INIT, DEFAULT, HARDCOPY, and CONSOLE), you will receive a syntax error during CONSOL*xx* parmlib processing indicated by message IEA195I or message IEA196I as shown in the example below:

```
- IEA196I CONSOLM1 03E0: NAME REQUIRED FOR CONSOLE.
- IEA196I CONSOLM1 INIT: DUPLICATE SPECIFICATION IGNORED.
- IEA196I CONSOLM1 03E0: UNRECOGNIZED KEYWORD INITDS(Y) IGNORED.
- IEA196I CONSOLM1 03E0: UNRECOGNIZED KEYWORD INITDS(Y) IGNORED.
- IEA195I CONSOLM1 LINE1: UNRECOGNIZED STATEMENT TYPE IGNORED.
- IEA195I CONSOLM1 LINE1: UNRECOGNIZED STATEMENT TYPE IGNORED.
```

Also, if you do not have a blank after the major keywords INIT, DEFAULT, and HARDCOPY, the default values will be used. In the case of the major keyword, CONSOLE, you will be left with only the system console if all of your CONSOLE statements do not end with a blank characters.
Migration action:

1. Examine your CONSOL*xx* parmlib member to verify that you have at least one blank after all of your major keyword statements.
2. If, you do not have a blank, update your CONSOL*xx* parmlib member by entering one or more blanks between the major keyword statements and their associated keywords.

Examine source for dynamic allocation callers that set the S99DSABA and S99ACUCB flags (Required-IF, as of R13)

Required if you have unauthorized dynamic allocation callers that set the S99DSABA and S99ACUCB flags.
TIOTs and XTIOTs contain entries for each DD statement allocated by either batch (JCL) or dynamic allocation. The TIOT is a below-the-line control block that contains contiguous DD entries, which allows for a sequential search. Because of limits on its size and structure, a TIOT can only accommodate a specific number of DD statements (for example, 3273 single unit DD statements for a TIOT size of 32k.)

To overcome this restriction, device allocation introduced XTIOTS or extended TIOTs above the 16M line, but the support was limited to authorized dynamic allocation callers only because the authorized flag S99TIOEX had to be set in order to build XTIOTs. Later, this restriction was relaxed when unauthorized dynamic allocation callers could request XTIOTs by setting S99ACUCB; however, the ability to get an above-the-line data set association block (DSAB) that contains a pointer to the TIOTs/ XTIOTs was limited to requests by authorized callers only, because the S99DSABA flag (which can be set by authorized or unauthorized callers) is honored only if the authorized S99TIOEX flag also has been set.

In z/OS V1R12, the Basic Access Method (BAM) added support for XTIOTs. Because it makes sense to allow unauthorized callers to get DSABs above the line, in z/OS V1R13, device allocation added support to build DSABs above the line when the S99DSABA bit flag is set and either S99ACUCB or S99TIOEX is also set. Thus, unauthorized users can fully utilize the virtual storage constraint relief (VSCR) capabilities provided by allocation and get the benefits of both the above-the-line DSABs and XTIOTs.

If any unauthorized dynamic allocation caller indicates through S99DSABA that above-the- line DSABs are supported but encounters a programming error in the user code when referencing above-the-line DSABs, action is required. Before z/OS V1R13, if the dynamic allocation callers set the S99DSABA and S99ACUCB flags, allocation built below-the-line DSABs, scanned the below-the-line DSAB queue, and found them below the line. For z/OS V1R13, if dynamic allocation callers request above-the-line DSABs through S99DSABA and S99ACUCB, allocation builds above-the-line DSABs, scans the above-the-line DSAB queue, and finds them above the line. If the dynamic allocation callers have an existing programming error when they attempt to reference above-the-line DSABs , they will continue to encounter errors. If these dynamic allocation callers need to use below-the-line DSABS , they should not set the S99DSABA.
Migration action: Examine source for use of S99DSABA. If found, verify that field DSQDSABA is not used and that 4 byte (31 bit) pointers are used if the DSAB is accessed by the program itself.

Discontinue use of PGSER to protect and unprotect the READONLY nucleus (Required-IF, as of R13, and in z/OS R12 with OA33782)

Required if you use the PGSER macro to protect or unprotect the READONLY nucleus.
Starting in z/OS V1R12, most of the READONLY nucleus is backed by 1 MB pages. This makes protecting or unprotecting the READONLY nucleus with the PGSER macro difficult because the macro can only handle virtual storage pages backed by 4 KB pages. Therefore, the PGSER macro is changed, with APAR OA33782, to no longer support requests to protect and unprotect the READONLY nucleus if it is backed by 1 MB pages.
Migration action: Do not use PGSER to protect or unprotect the READONLY nucleus when it is backed by 1 MB pages. Users requiring the modification of READONLY nucleus should use the DATOFF macro.

Failure to discontinue use of PGSER to protect and unprotect READONLY nucleus that is backed 1 MB pages will result in the following ABEND18A reason codes:

- FF070411– The caller issued a PGSER macro with the PROTECT parameter for virtual storage in the READONLY nucleus that is backed by 1 MB pages. This storage area cannot be specified with the PROTECT keyword.
- FF080411 – The caller issued a PGSER macro with the UNPROTECT parameter for virtual storage in the READONLY nucleus that is backed by 1 MB pages. This storage area cannot be specified with the UNPROTECT keyword.

BCP Migration Actions Pre-First IPL

Create IPL text (Required)

Migration action: Update and run the IPLTEXT job to write a new copy of the IPL text. If you install z/OS with a ServerPac, an installation dialog job is provided to perform this action. If you install z/OS with a CBPDO, instructions to perform this action are provided in *z/OS Program Directory*.
Note: When the IPLTXTEXIST parameter (which was introduced by ICKDSF R17 APAR PK16403) is specified with the REFORMAT command using the IPLDD parameter, WTOR message ICK21836D is suppressed if IPL text already exists

Reassemble the stand-alone dump program (Required)

The stand-alone dump one-step (now called one-stage) JCL is changed in z/OS V1R12, and in prior levels with APAR OA31077. If using one-stage JCL to create your stand-alone dump program, ensure that appropriate

changes are made to your one-stage JCL. Sample JCL can be found in *z/OS MVS Diagnosis: Tools and Service Aids*.
Migration action: Reassemble the stand-alone dump program. If you install z/OS with a ServerPac, an installation dialog job is provided to perform this action (DMPBLD). If you install z/OS with a CBPDO, instructions to perform this action are provided in *z/OS Program Directory.* Once the stand-alone dump program is properly created on a DASD residence volume, it resides in the SYS1.PAGEDUMP.Vvolser data set.

Remove references to the MTFTPS utility (Recommended, as of R13)

Not required, but recommended if you previously used the stand-alone version of PDUU (MTFTPS).
Before z/OS V1R13, you might have used the problem documentation upload utility (PDUU), packaged as MTFTPS, to send large volumes of problem documentation, such as stand-alone dumps, to IBM support. Beginning with z/OS V1R13, the z/OS problem documentation upload utility (PDUU) is a standard part of the base operating system with entry point name AMAPDUPL.
Migration action: To avoid possible conflicts, remove the stand-alone version of the PDUU utility and begin using the supported version:

1. Remove any prior version of MTFTPS from your system. The PDUU utility name is AMAPDUPL (in SYS1.MIGLIB), although MTFTPS is shipped as an alias entry point to AMAPDUPL
2. Begin using the PDUU as the primary utility for sending large volumes of product documentation to IBM Support.

Change value for ARM restart processing (Required-IF, as of R13)

Required if you have the PTF for APAR OA35357 applied and a five-minue timeout for member cleanup is not possible.
Before performing cross-system restart, automatic restart management (ARM) waits for member cleanup for the terminated system to complete. ARM proceeds with cross-system restart if cleanup takes longer than a certain amount of time. Before z/OS V1R13, this time was two minutes. Support for a new parameter, CLEANUP_TIMEOUT, is available with the PTFs for APAR OA35357 applied to z/OS V1R13, z/OS V1R12, z/OS V1R11, and z/OS V1R10. The default for this new parameter is five minutes. That is, ARM will wait five minutes for member cleanup for a terminated system to complete before performing cross-system restart for an element.

Starting with z/OS V1R13, the CLEANUP_TIMEOUT parameter can be used to indicate that ARM is to wait additional time for member cleanup for a terminated system to complete. To get the two minute timeout behavior that existed before the default change, CLEANUP_TIMEOUT(120) must be added to the ARM policy. If you do not specify CLEANUP_TIMEOUT(120), the system issues the following message to the system log to record when CLEANUP_TIMEOUT has an effect on cross-system restart processing:

```
IXC815I MEMBER CLEANUP FOR SYSTEM sysname1 NUMBER sysnum1 INCOMPLETE
```

The new five minute default, or any use of the CLEANUP_TIMEOUT parameter other than CLEANUP_TIMEOUT(120), is not fully effective until all systems in the sysplex have support for the CLEANUP_TIMEOUT parameter. APAR OA35357 provides support for the CLEANUP_TIMEOUT parameter.
Migration action: If you prefer to use the two minute value for ARM restart processing, do the following:

1. Use the z/OS V1R13 version of IXCMIAPU to define an ARM policy with CLEANUP_TIMEOUT(120).
2. Use the SETXCF START command to start the new or updated policy.

Modify automation that references output from D XCF,SYSPLEX console commands (Required-IF, as of R13)

Required if you use automation programs or other procedures to handle message IXC351I.
Before z/OS V1R13, when some **D XCF** console commands were issued, the resulting messages contained output information from the command depending on the options specified. Starting with z/OS V1R13, the output message for a **D XCF,SYSPLEX** command is changed to IXC336I, which provides more basic information about a system.
Migration action: Modify automation that references output from **D XCF,SYSPLEX**, **D XCF,SYSPLEX,ALL**, and **D XCF,SYSPLEX,***systemname* commands. Message IXC337I replaces IXC335I. IXC335I is no longer issued.
Reference information: For details about the message output for IXC334I, IXC336I, and IXC337I, see *z/OS MVS System Messages, Vol 10 (IXC-IZP).*

Update LLA for automation (Required-IF, as of R13)

Required if you have automation or operator procedures that restart LLA and you want LLA to restart with no parmlib member even when you previously started LLA with a parmlib member.
Before z/OS V1R13, if you started library lookaside (LLA) using a CSVLLA*xx* parmlib member, and then stopped and restarted LLA without using a parmlib member, LLA honored the "no parmlib member" state and managed only

the data sets in the LNKLST concatenation. Beginning with z/OS V1R13, the same scenario results in using the CSVLLA*xx* parmlib member with which LLA previously started. To get back to the "no parmlib member" state, you must specify **LLA=NONE** when starting LLA.
Migration action: If you have automation in place to restart LLA and you want automation to restart without a parmlib member even when you had started LLA with a parmlib member, you must change it to use the LLA=NONE parameter.

Accommodate OPERLOG EMCS console name change (Required-IF, as of R13, and back to R9 with APAR OA31913)

Required if you depend on the OPERLOG EMCS console name.
Starting with z/OS V1R13 (and z/OS V1R12, z/OS V1R11, and z/OS V1R10 with the PTF for APAR OA31913 applied), the OPERLOG EMCS console name *OPLOG*yy* is generated using the two character System Clone value (&SYSCLONE). The default &SYSCLONE value is obtained from the System Name (&SYSNAME) (for example, System Name = SYSTEM1 / System Clone = M1). This naming convention is similar to the SYSLOG EMCS console (*SYSLG*yy*).
Migration action: The change of OPERLOG EMCS console name spans all configurations (MULTISYSTEM, XCFLOCAL, MONOPLEX, in GRS RING or STAR mode). If you depend on the name of OPERLOG EMCS console in your own procedure, it must be adjusted to reflect this change. For example, the following will display the OPERLOG EMCS console name:

```
D C,KEY=OPERLOG (message IEE892I)
D EMCS (message IEE129I)
D EMCS,CN=*OPLOG* (message IEE129I)
```

Note: With the PTF for APAR OA30757 applied to z/OS V1R11 or z/OS V1R10, and in z/OS V1R12, this change was already in effect.

Adjust CON= system parameter to accommodate default change (Required-IF, as of R13)

Required if there is no specification on the CON= system parameter and SHARED mode is required.
Before z/OS V1R13, the default console operating mode was SHARED. Beginning with z/OS V1R13, the default console operating mode is changing from SHARED mode to DISTRIBUTED mode. SHARED mode will be removed in a future release. DISTRIBUTED mode is now the preferred mode of operations.
Migration action: Examine the system parameters used to IPL the system or sysplex. The initial mode is specified on the CON= system parameter. Use the **D OPDATA,MODE** to find the current mode, which is displayed in message CNZ9006I.

- If DISTRIBUTED is specified, no action is required.
- If SHARED is specified, an action is not currently required, but DISTRIBUTED mode will become a required action in the future.
- If there is no specification on the CON= system parameter, DISTRIBUTED mode is now the default.
- If there is no specification on the CON= system parameter and SHARED mode is required, you have to explicitly request the SHARED mode on the CON= system parameter. This allows the system or systems to continue functioning in the same manner as they do today. Use the SETCON MODE=SHARED command to request SHARED mode.

Tip: When you activate the OPERCMDS FACILITY class, you must have the CONTROL access authority to the profile when issuing the SETCON MODE command.

Start Runtime Diagnostics at system initialization (Recommended, as of R13)

Not required, but recommended to use Runtime Diagnostics.
Before z/OS V1R13, Runtime Diagnostics ran as a as a started task under the master subsystem and had to be started each time you wanted an analysis. It was started, did its analysis, then ended. Beginning with z/OS V1R13, you can start Runtime Diagnostics to run as an address space under the master subsystem. After you start the Runtime Diagnostics address space (HZR), it remains running until stopped using the STOP command. Use the MODIFY HZR,ANALYZE command to generate a Runtime Diagnostics analysis and report.
Migration action: To start the Runtime Diagnostics address space (HZR) on z/OS V1R13:

1. Ensure the hzrproc (HZR) points to PGM=HZRINIT, not PGM=HZRIMAIN as in z/OS V1R12. The hzrproc (HZR) ships in the SYS1.PROCLIB data set.
2. If you want to start Runtime Diagnostics address space (HZR) during system initialization, specify COM='S HZR,SUB=MSTR' in the COMMND*xx* parmlib member. Otherwise, the HZR address space must be started manually: S HZR,SUB=MSTR.

3. After the Runtime Diagnostics address space (HZR) is started, use the MODIFY HZR,ANALYZE command to generate Runtime Diagnostics' reports.

Reference information: For complete details about using Runtime Diagnostics, see the topic on Runtime Diagnostics overview in *z/OS Problem Management*.

Ensure all modules of an application are compiled with the same version of the IRARASD macro (Recommended, as of R12)

Not required, but recommended to avoid incompatible work area sizes between program and declaration when declaration of the work area id done outside the program (DSECT).

Starting with z/OS V1R12, the IRARASD macro (field RQFASDWA) defines a 512-byte (changed from 338-byte) work area for SYSEVENT REQFASD. When you recompile a module that includes the IRARASD macro, you must also recompile all other modules that include the IRARASD macro and belong to the same application.

It is only necessary to ensure that all modules of an application use the same version of the IRARASD macro; it is not necessary to recompile all modules with the z/OS V1R12 version of the IRARASD macro.

Migration action: Ensure all modules of an application are compiled with the same version of the IRARASD macro.

Issue commands from the system console regardless of problem determination mode (Recommended, as of R12 and R11 with APAR OA34731)

Not required, but recommended to immediately enter commands from the system console during critical situations.

Before z/OS V1R12, the system console could not issue any commands (except **REPLY** and **VARY CN(*),ACTIVATE**) unless it was placed in problem determination (PD) mode through the **VARY CN(*),ACTIVATE** command. Starting with z/OS V1R12, processing changed so that commands could always be entered at the system console regardless of problem determination mode. With APAR OA34731, commands can only be entered at the system console if the **CONSOL***xx* keyword ALLOWCMD(Y) is specified on the system console definition.

Migration action: Add **ALLOWCMD(Y)** to the system console definition in the **CONSOL***xx* member of PARMLIB if you wish to allow commands to be issued from the system console regardless of problem determination mode.

Use the new 16M default buffer size for trace options with the CTIGRSxx member (Required-IF, as of R12)

Required if you modified any IBM-supplied procedures.

Before z/OS V1R12, the default buffer value (BUFSIZE) for the trace option with the GRS component in the IBM-supplied CTIGRS00 parmlib member was 128K, the maximum value was 16MB, and the buffer was in GRS below the bar private storage. Starting with z/OS V1R12, the default size used by CTIGRS00 is increased to 16MB, the maximum is 2047MB, and the buffer is in GRS above the bar private storage in its own memory object. If you specify your own CTIGRS*xx* member on the CTRACE option in GRSCNF*xx* parmlib member, change the BUFSIZE in CTIGRS*xx* to 16MB.

Migration action: Change the trace option BUFSIZE in your CTIGRS*xx* parmlib member to 16M or make the new default value (16M) active.

Specify valid user exits for the IFASMFDL and IFASMFDP Programs (Required-IF, as of z/OS V1R12, and z/OS V1R11, z/OS V1R10, and z/OS V1R9 with APAR OA29894, then changed in OA33696)

Required if you use user exit routines.

When you run the SMF log stream dump program (IFASMFDL) or the SMF data set dump program (IFASMFDP), the name of an installation-written exit routine that is given control at the indicated times can be specified by the USER*x* (*x*=1, 2, or 3) parameter.

Beginning with z/OS V1R12 (and z/OS V1R11 and z/OS V1R10 with the PTFs for APAR OA29894 applied), to allow user exits to be called by the SMF dump programs, the exits must now be pre-defined to the system using the new keywords shipped in the SMFPRM*xx* parmlib member. The SMFDLEXIT keyword allows exits to be specified for the IFASMFDL program, and the SMFDPEXIT keyword allows exits to be specified for the IFASMFDP program.

Both keywords have the same suboptions, USER1, USER2 and USER3. The suboptions allow multiple exits to be specified for each user exit point in the respective dump program. The syntax follows.

```
SMFDLEXIT( USER1( exit1, exit2, ... ) | NOUSER1 ,
      USER2( exit1, exit2, ... ) | NOUSER2 ,
      USER3( exit1, exit2, ... ) | NOUSER3 )
```

```
SMFDPEXIT( USER1( exit1, exit2, ... ) | NOUSER1 ,
      USER2( exit1, exit2, ... ) | NOUSER2 ,
      USER3( exit1, exit2, ... ) | NOUSER3 )
```

Note: In z/OS V1R13, or with the PTF for SMF APAR OA33696 applied to z/OS V1R12 or z/OS V1R11, IFASMFDP can once again run in a non-APF authorized environment when using the DUMP option. Also, when IFASMFDP is executed in an unauthorized environment, the exits specified with USER1, USER2 and USER3 will not be verified against what is in the SMFPRM*xx* parmlib member.
Migration action: You must apply the PTF for APAR OA29894 if you share the SMFPRM*xx* parmlib member, with the keyword SMFDLEXIT or SMFDPEXIT specified, on multiple systems.

- Register an exit specified for IFASMFDL or IFASMFDP using the USER1, USER2 or USER3 parameters through the SMFPRM*xx* parmlib member. If you fail to do this, you will receive the following error message, and IFASMFDL or IFASMFDP will stop the processing: `IFA840I USER EXIT` *`xxxxxxxx`* `NOT REGISTERED WITH SYSTEM`
- View the D SMF,O command output to determine if the exits are successfully registered.

Note: You do not need to explicitly define the exits used by the operation of RACF SMF unload utility; they are registered by default:

```
SMFDLEXIT(USER2(IRRADU00),USER3(IRRADU86))
SMFDPEXIT(USER2(IRRADU00),USER3(IRRADU86))
```

Make IFASMFDL and IFASMFDP run in an authorized environment (Required-IF, as of z/OS V1R12, and z/OS V1R11, z/OS V1R10, and z/OS V1R9 with APAR OA29894, then changed with OA33696)

Required if you do not invoke the SMF dump program as a jobstep task.
When you run the SMF log stream dump program (IFASMFDL) or SMF data set dump program (IFASMFDP), the DUMP function is permitted even in an unauthorized environment. (Note: The CLEAR function of the SMF data set dump program requires APF authorization. No such function exists for the SMF log stream dump program. Beginning with z/OS V1R12 (and z/OS V1R11 and z/OS V1R10 with the PTFs for APAR OA29894 applied), the IFASMFDL and IFASMFDP programs are required to run in an authorized environment. Otherwise, the programs will lose authorization and you will get an S338 followed by a S330 abend, especially if the programs are being executed under TSO/E.
Note: In z/OS V1R13, or with the PTF for SMF APAR OA33696 applied to z/OS V1R12 or z/OS V1R11, IFASMFDP can once again run in a non-APF authorized environment when using the DUMP option.
Migration action:

- If you run the SMF dump program using JCL, the APF-authorization assigned to it is preserved, and no action is required.
- For the TSO/E environment, you need to add IFASMFDL and IFASMFDP to the AUTHPGM section of the IKJTSO*xx* parmlib member when the programs are being invoked using a TSO/E CALL command.
- If LINKMVS or ATTCHMVS is used in a REXX program to invoke IFASMFDL or IFASMFDP, change the invocations of IFASMFDL and IFASMFDP using LINKMVS or ATTCHMVS to use the TSO/E CALL command. In addition, add the IFASMFDL and IFASMFDP program to the AUTHPGM section of the IKJTSO*xx* parmlib member.
- If calling IFASMFDL or IFASMFDP from another program, that program must be authorized.

Note: A ++HOLD(ACT) is being shipped with the PTFs for APAR OA32258 to notify that IFASMFDL and IFASMFDP need to be executed in an authorized environment.

Specify installation script parameter when installing Predictive Failure Analysis (Required-IF, as of R12)

Required if you use PFA.
Before z/OS V1R12, when installing Predictive Failure Analysis (PFA) it was not necessary for you to specify how to handle the existing data in the check directories. Beginning with z/OS V1R12, you must append a parameter, *migrate* or *new*, on the installation script to specify if the PFA check directories retain data from the prior release. If you do not append the *migrate* or *new* parameter, the AIRSHREP.sh script fails.
Migration action: Provide the *migrate* or *new* parameter when running the PFA install script, AIRSHREP.sh, or when using the sample JCL for batch provided in SYS1.SAMPLIB.

- *migrate*: Use the *migrate* parameter to preserve PFA history data from the prior release. The *migrate* option is recommended for all installations that previously used PFA. When you specify the *migrate* parameter, the install script preserves data from the prior release and creates new directory structures for checks not previously installed on your system.

- *new*: Use the *new* parameter if you are installing PFA for the first time or if you want to delete everything from prior releases and start PFA with new directories. When you specify the *new* parameter, the install script deletes the existing check directories and creates a new directory structure for all the checks.

For example, to run the install script from your home directory:

1. On the OMVS command line, go to the home directory for the PFA user: cd /pfa
2. Run the install script appending the migrate parameter: /usr/lpp/bcp/ AIRSHREP.sh migrate

Refer to *z/OS Problem Management* for considerations and actions required for Predictive Failure Analysis (PFA) when falling back to z/OS V1R11 or z/OS V1R10 from z/OS V1R12.

Note that PFA is a new user of /etc. /etc/PFA/ini is a new file that contains consolidated path information. If found, this file will be used for all checks. However, if you want to specify different paths for different checks, you can continue to use the ini file found in each check directory.

Verify MAXFILEPROC for PFA usage (Required-IF, as of R12)

Required if you use PFA, and MAXFILEPROC is less than 5000.

Starting in z/OS V1R13, Predictive Failure Analysis (PFA) has two additional checks, PFA_JES_SPOOL_USAGE and PFA_ENQUEUE_REQUEST_RATE. These additional checks add to the number of open files for the PFA started task id.

Migration action: PFA requires the maximum number of open files under Unix System Services to be 5000 or greater. This can be done by:

1. Updating BPXPRMxx MAXFILEPROC or preferably
2. Updating the FILEPROCMAX field in the OMVS segment of the PFA started task id, i.e: TSO ALU *pfauser* OMVS(FILEPROCMAX(5000))

Note: If your installation has MAXFILEPROC greater than 5000, then this action is not required.

Change default locations for LCCA or PCCA control blocks to retain 24-bit virtual storage location (Required-IF, as of R12)

Required if you want to keep the 24-bit virtual storage location for IHALCCA or IHAPCCA.

In parmlib member DIAG*xx*, you can specify the virtual location of the LCCA control block (mapped by the IHALCCA macro) or the location of the PCCA control block (mapped by the IHAPCCA macro) to be used when a CPU is brought online. Before z/OS V1R12, if the IHALCCA or IHAPCCA structures specified on the CBLOC VIRTUAL24 or VIRTUAL31 keyword of DIAG*xx* did not specify either the 24-bit (VIRTUAL24) or 31-bit (VIRTUAL31) location, the default location for the LCCA or PCCA was in 24-bit virtual storage.

Beginning with z/OS V1R12, the default location for the LCCA and PCCA is now in 31-bit virtual storage if you do not specify VIRTUAL24 or VIRTUAL31 for the structures IHALCCA and IHAPCCA. If only one standard processor is online during the IPL, the LCCA and PCCA of the IPL CPU will be in 24-bit storage regardless of the specification.

Migration action: If you want to retain the 24-bit virtual storage location for IHALCCA or IHAPCCA, you specify VIRTUAL24 on the CBLOC keyword in DIAG*xx* parmlib member:

```
VIRTUAL24(IHALCCA)
VIRTUAL24(IHAPCCA)
```

Tip: The following z/OS V1R12 migration checks, (available with APAR OA32015 on z/OS V1R11 and z/OS V1R10), determine if your current settings match RMODE 31 for the LCCA or PCCA control blocks: CHECK(IBMSUP,ZOSMIGV1R12_SUP_LCCA_ABOVE_16M) and CHECK(IBMSUP,ZOSMIGV1R12_RCF_PCCA_ABOVE_16M). As is the convention, these checks are shipped inactive and are to be activated when exploring migration actions.

Remove reference to Unicode Services pre-built image CUNIDHC2 (Required-IF, as of R12)

Required if the SYS1.SCUNIMG is in the LNKLST specification.

Starting in z/OS V1R7, Unicode Services provided support to dynamically load tables into storage when a Unicode service request is made and the table is not already in storage. This enhancement is colloquially known as “Unicode On Demand.” If the appropriate table needed to satisfy the service request is not in storage, Unicode Services will load the table dynamically without disrupting the caller’s request. All tables needed for character conversion, case conversion, normalization, and collation services are now loaded automatically into storage when they are required and not already present. These tables are added to other tables already in storage.

Starting in z/OS V1R12, the pre-built image CUNIDHC2 has been eliminated. This pre-built image contained all the conversion tables supported by DB2 and would be loaded into storage when you had an empty Unicode environment (no UNI=*xx* in the IEASYS*xx* member) and the first requestor of a Unicode conversion service would be DB2. Given that most customers would use only a handful of these tables and given that Unicode Service has the capability to dynamically load tables into storage, the need for pre-built image has become obsolete. Unicode Services will no longer ship the pre-built image SYS1.SCUNIMG(CUNIDHC2) and will no longer automatically load the pre-built image. We recommend that you use the Unicode On Demand capability to load all tables.

Migration action:

- Remove SYS1.SCUNIMG from the LNKLST specification.
- Remove SYS1.SCUNIMG from the APF authorization list. Note that if your installation is running with the entire LNKLST as APF authorized (LNKAUTH=LNKLST), you do not need to take any additional action. If your installation is not running with the LNKLST as APF authorized (LNKAUTH=APFTAB), you need to explicitly remove the APF authorization for this data set.
- Remove the catalog entry for SYS1.SCUNIMG.
- Remove the following DDDEF entries:

Name	**Data Set Name**
ACUNIMG	SYS1.ACUNIMG
SCUNIMG	SYS1.SCUNIMG

Note: If you want to create and load an image with the same conversion tables as the eliminated pre-built image, do the following:

1. Invoke the image generator program using the JCL provided in SYS1.SCUNJCL(CUNJIUTL). Use the conversion statements specified in SYS1.SAMPLIB(CUNSISM6). CUNSISM6 contains all the conversion statements needed to build the pre-built image.
2. After building the image, specify the image name in the CUNUNI*xx* parmlib member.
3. Specify the corresponding UNI=*xx* parameter in your IEASYS*xx* parmlib member and re-IPL.

Remove classification rules with the ETC work qualifier (Recommended, as of R12)

Not required, but recommended, otherwise you will have to delete the classification rules the next time you use the WLM ISPF application to modify the EWLM subsystem type.

Beginning with z/OS V1R12, the workload management (WLM) service definition no longer supports the work qualifier EWLM transaction class name (ETC) for classification rules of the subsystem type EWLM. If the activated service definition contains classification rules with qualifier type ETC, they are simply ignored by WLM. However, the next time you modify the classification rules for subsystem EWLM with the WLM ISPF Application, you have to delete the ETC rules before you save your modifications. Although z/OS V1R12 disregards classification rules with the ETC work qualifier, we recommend that you remove them. If you do not remove the rules, you will have to delete them the next time you use the WLM ISPF application to modify the EWLM subsystem type.

Migration action: If your WLM service definitions contain classification rules for subsystem type EWLM with the ETC work qualifier, start the WLM ISPF application and choose the Classification Rules option from the Definition Menu. Use the Modifiy option (3) for the IBM-supplied subsystem type EWLM. Delete all rows with the ETC qualifier type by using the Delete row Option (D).

Update the SFM policy to control automatic termination of impaired critical members (Recommended, as of R12)

Not required, but recommended when you want to designate how long XCF will wait before initiating termination of the impaired critical members.

Starting with z/OS V1R12, a member of an XCF group can identify itself as being critical to the operation of the group or the system. If a critical member appears to be inoperative (impaired) and the condition persists long enough, XCF automatically terminates the member in an attempt to resolve the problem. For a member that is critical to the operation of the system, this termination causes the system to be removed from the sysplex. For more information about XCF groups, see *z/OS MVS Setting Up a Sysplex*.

Members of the SYSGRS group, for instance, are critical to the operation of the system. If any GRS member is impaired, ENQ processing is likely impacted throughout the sysplex. Failure to perform ENQ processing in a timely fashion has significant negative impact. Thus if a GRS member appears to be impaired, XCF will automatically remove from the sysplex, the system on which that member resides.

You can set the MEMSTALLTIME parameter in your sysplex failure management (SFM) policy to control how long XCF allows a critical member to persist in an impaired state before it initiates termination of the member (or the member's system). If the MEMSTALLTIME specification resolves to NO (either implicitly or explicitly), XCF will terminate an impaired critical member if the condition persists as long as the failure detection interval (INTERVAL) of the system on which the member resides, or if the condition persists as long as two minutes, whichever is greater. To determine which groups are using the critical support, issue the appropriate XCF display command.

The MEMSTALLTIME parameter also determines how long XCF allows a signaling sympathy sickness condition to persist before terminating a stalled group member that is contributing to the problem.

The MEMSTALLTIME parameter indicates the number of seconds that XCF should wait before it terminates a member that is impacting the sysplex. A MEMSTALLTIME value of 120 (two minutes) seems to suit many installations as it provides some additional time for the system to resume normal operation, yet allows automatic action to resolve the problem before the sympathy sickness condition critically impacts the sysplex. Installations that resolve such conditions through manual intervention sometimes use a higher value to allow time for such intervention to be accomplished. Installations that are less able to tolerate sympathy sickness conditions sometimes set lower values.

Accommodate new REUSASID default (Required-IF, as of R12)

Required, as 0D3 abends might occur with downlevel products that provide no toleration support for reusable ASIDs. Because reusable ASIDs have been available since z/OS V1R9, it is reasonable to expect that the current levels of products are tolerant of reusable ASIDs.

In z/OS V1R9, the REUSASID(YES|NO) parameter in parmlib member DIAG*xx* was introduced with a default of NO. Starting with z/OS V1R12, the default is changed to YES.

When a reusable ASID is requested by the START command or the ASCRE macro, this reusable ASID is assigned if REUSASID(YES) is specified in DIAG*xx*. If REUSASID(NO) is specified in DIAG*xx*, an ordinary ASID is assigned. The default is REUSASID(YES). The use of reusable ASIDs might result in system 0D3 abends, if products or programs have not been upgraded to tolerate reusable ASIDs. For more information about reusable ASIDs, see *z/OS MVS Programming: Extended Addressability Guide*.

Migration action:

1. On z/OS V1R11 or z/OS V1R10 systems, specify REUSASID(YES) in the parmlib member DIAG*xx*. On z/OS V1R12 systems, keep REUSASID(YES), or allow it to default to YES. You can use the SET DIAG=*xx* command to change the REUSASID (YES|NO) option.
2. Verify that no 0D3 abends occur as a result.
3. If 0D3 abends do occur, apply appropriate maintenance to the affected code.
4. If this is not possible, specify REUSASID(NO)

Review the list of WTORs in parmlib member AUTOR00 (Required, as of R12)

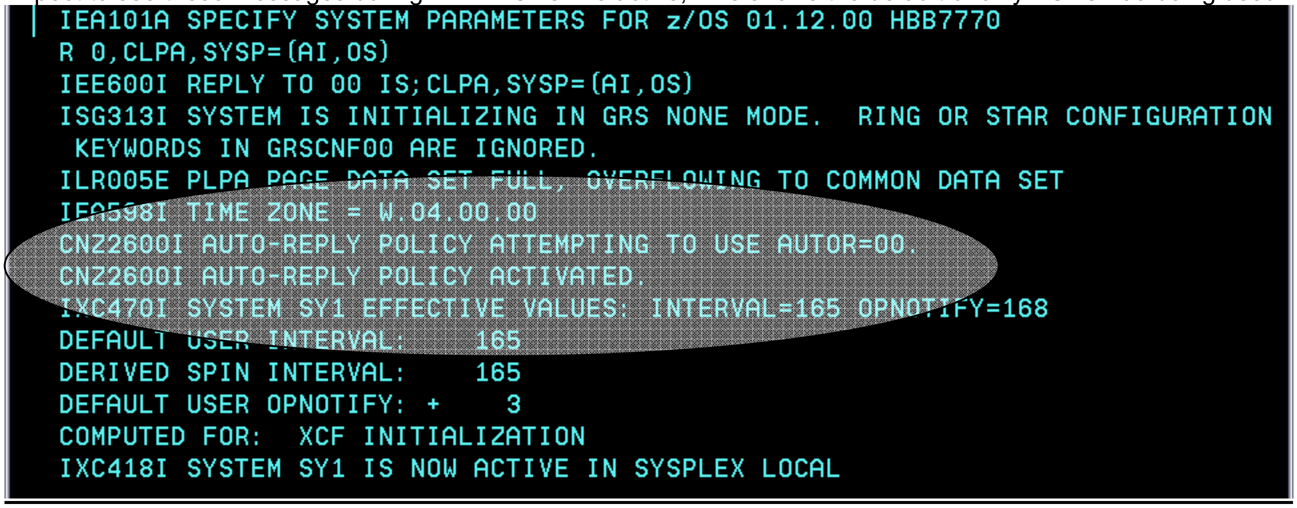
In z/OS V1R12, the DDDEF"d PARMLIB provides an AUTOR00 member. This member should be found in your parmlib concatenation during IPL and will result in auto-reply processing being activated. If the WTORs listed in AUTOR00 are automated by your existing automation product, ensure that the replies in AUTOR00 are appropriate. You can see the contents of the default in AUTOR00 in *MVS Planning: Operations* and verify that the proper actions are taken.

Migration action: Examine the WTOR replies in the AUTOR00 parmlib member. If the replies or delay duration are not desirable, you can create a new AUTOR*xx* parmlib member and make corresponding changes. Also compare the replies to what your automation product would reply to these WTORs. Make sure that the AUTOR00 replies are in accordance with the replies from your automation product. IBM does not recommend making updates to AUTOR00, because updates to AUTOR00 might be made by the service stream or in new z/OS releases.

Notes:

1. If you have created an AUTOR*xx* parmlib member, update the IEASYS*yy* parmlib member that you use for IPL. Add the following statement to the IEASYS*yy* member: AUTOR=(xx,00) . Here *xx* corresponds to the AUTORxx parmlib member that you created. The IEASYS*yy* members specifying AUTOR cannot be shared with prior z/OS releases. If you only need the default AUTOR00 settings, you can omit specifying AUTOR= in IEASYS*yy*, and other z/OS levels can continue to use IEASYS*yy*. Even if AUTOR= is not specified in IEASYS*yy*, AUTOR00 is used if it exists.
   Note: If you have automation already in place for a WTOR and that WTOR now appears in AUTOR, the AUTOR version still can be used. Review the WTORs in AUTOR and your automation; either remove the WTOR from AUTOR or from your automation.
2. If you don't want to activate auto-reply processing, specify AUTOR=OFF in the parmlib member IEASYS*xx* or in response to message `IEA101A SPECIFY SYSTEM PARAMETERS`. It is not recommended that you remove AUTOR00 from parmlib, because service or new releases might reinstall AUTOR00. If there is no AUTOR00 member in parmlib, auto-reply is not activated and the following messages are produced:
   ```
   CNZ2600I AUTO-REPLY POLICY ATTEMPTING TO USE AUTOR=00.
   IEA301I AUTOR00 NOT FOUND IN PARMLIB
   CNZ2601I AUTO-REPLY POLICY NOT ACTIVATED.
   NO ENTRIES SPECIFIED
   ```
3. The IEASYSyy members specifying AUTOR=OFF cannot be shared with prior z/OS releases.

Expect to see these messages during IPL if AUTOR is active, This shows the default of only AUTOR00 being used:



```
IEA101A SPECIFY SYSTEM PARAMETERS FOR z/OS 01.12.00 HBB7770
R 0,CLPA,SYSP=(AI,OS)
IEE600I REPLY TO 00 IS;CLPA,SYSP=(AI,OS)
ISG313I SYSTEM IS INITIALIZING IN GRS NONE MODE.  RING OR STAR CONFIGURATION
 KEYWORDS IN GRSCNF00 ARE IGNORED.
ILR005E PLPA PAGE DATA SET FULL, OVERFLOWING TO COMMON DATA SET
IEA598I TIME ZONE = W.04.00.00
CNZ2600I AUTO-REPLY POLICY ATTEMPTING TO USE AUTOR=00.
CNZ2600I AUTO-REPLY POLICY ACTIVATED.
IXC470I SYSTEM SY1 EFFECTIVE VALUES: INTERVAL=165 OPNOTIFY=168
DEFAULT USER INTERVAL:     165
DERIVED SPIN INTERVAL:     165
DEFAULT USER OPNOTIFY: +     3
COMPUTED FOR:  XCF INITIALIZATION
IXC418I SYSTEM SY1 IS NOW ACTIVE IN SYSPLEX LOCAL
```

Update automation that handles some deleted SMF messages: IEF374I and IEF376I (Required-IF, as of R12)

Required if you use automation programs or other procedures to handle messages IEF374I and IEF376I.

Starting in z/OS V1R12, message IEF374I is replaced by message IEF032I and message IEF376I is replaced by message IEF033I. If you use automation programs to handle messages, or you have operator or other procedures that deal with messages, you should update the programs or procedures appropriately.

Migration action: Modify automated actions for IEF374I so they now work with message IEF032I. Modify automated actions for IEF376I so they now work with message IEF033I.

BCP Migration Actions Post-First IPL

Set AUTHQLVL parameter in GRSCNFxx parmlib member to recognize new GRS qnames (Required, as of R13)

Required, to protect authorized programs utilizing these qnames from denial-of-service attacks.

Beginning with z/OS V1R13, global resource serialization (GRS) provides an additional list of qnames that are conditionally authorized: ARCDSN, ARCBTAPE, ARCGPA, ARCBACV, and ARCMIGV. You can set the new AUTHQLVL parameter in the GRSCNF*xx* parmlib member to indicate whether the system is to recognize the second list of authorized qnames in addition to the original list. The value is either 1 (default) or 2.

The AUTHQLVL setting of 1 (default) denotes that the existing IBM default list for authorized qnames (that is, the list in effect for systems at z/OS V1R12 and earlier) is in effect for the system in the global resource serialization (GRS) complex. The AUTHQLVL setting of 2 denotes the addition of the five new qnames (ARCDSN, ARCBTAPE, ARCGPA, ARCBACV, ARCMIGV) to the authorized qname list and provides a higher level of protection; however, it can cause some products to fail.

An unauthorized program issuing ENQ or DEQ requests for any of these qnames when AUTHQLVL of 2 is in effect will get ABEND338 or ABEND330, respectively. ISGENQ requests with COND=NO will get similar ABENDs and ISGENQ requests with COND=YES will get return code 8, reason code xxxx081E, ISGENQRsn_NotAuthorizedForQName.

Migration action:

1. Products that are designed to interact with resources that have these qnames need to run authorized. In order to help determine if your installation has any of these products running on your system, there is a new AUTHQ2 filter on the EQDQ Monitor. Before the new AUTHQLVL is increased to 2, this filter shows ENQ requests that are made with any of these new qnames from an unauthorized program.
2. The AUTHQLVL setting in GRSCNF*xx* refers to a specific system. A rolling IPL is required to ensure consistency across the GRS complex. The increased AUTHQLVL value can be tested on one system but

only for ENQ requests initiated on that system. The AUTHQLVL=2 migration process is not complete until all systems across the GRS complex are at 2.

3. If ABEND338 or ABEND330 occurs from the change because one of the required products is missed, the **SETGRS** command supports a fallback to 1 on any given system by issuing **SETGRS AUTHQLVL=1**; however, you cannot dynamically increase the AUTHQLVL. Another IPL is required

Examine use of the CMDS ABEND command (Recommended, as of R13)

Not required, but recommended if you use the CMDS ABEND command or have automation that does.

Before z/OS V1R13, the CMDS ABEND command ended an executing command if the command was hung. In z/OS V1R12, or with the PTFs for APAR OA30527 installed on z/OS V1R11 and z/OS V1R10, the command processors were allowed to specify the new non-abendable attribute to set themselves non-abendable. When the new attribute was specified for a target command, the CMDS ABEND command rejected with message CNZ6002I. The CMDS ABEND attempted to terminate the hung command.. Starting in z/OS V1R13, the new parameter FORCE is added to the CMDS command so that a CMDS FORCE specification overrides the non-abendable attribute and the command will be terminated as it is today. Separating the ABEND and FORCE requests allow different RACF profiles to be defined so that installations can allow CMDS ABEND, but not CMDS FORCE. FORCE is intended to be used where the only alternative is to re-IPL the system.

Migration action: Automation that uses the CMDS ABEND command is affected because the termination of a running command can be rejected. For some commands, this rejection is important because it can prevent a system or sysplex outage. If you use the CMDS ABEND command or have automation that does, certain commands will no longer be terminated by the CMDS ABEND command.

- If you must terminate a command, continue to use the CMDS ABEND command. If the command is in a state making it non-abendable, use the CMDS FORCE command after understanding the following recommendations associated with the FORCE parameter:
  - After issuing CMDS FORCE, you might have to re-IPL the system or, depending on the command being terminated, a sysplex-wide IPL may be required.
  - You should ensure that the target command is hung and not just requiring additional time to complete.

Ensure Runtime Diagnostics is installed before invoking Predictive Failure Analysis (Required-IF, as of R13)

Required if you plan on using the mentioned checks below.

Beginning with z/OS V1R13, Predictive Failure Analysis (PFA) calls Runtime Diagnostics to analyze and report insufficient metric activity from the following checks:

- PFA_ENQUEUE_REQUEST_RATE
- PFA_MESSAGE_ARRIVAL_RATE
- PFA_SMF_ARRIVAL_RATE

Therefore, PFA requires the Runtime Diagnostics address space (HZR) to be active on any system it analyzes.

Migration action: Ensure the HZR is defined as a system address space.

Carry over your existing CPPC policy (Required-IF, as of R13)

Required if you want to continue using your existing CPCC workspace.

The Capacity Provisioning Control Center (CPCC) can coexist with older versions of the CPCC on a workstation. However, z/OS V1R13 CPCC uses a different default workspace. If you want to continue using your existing CPCC workspace, point to the respective directory during the installation of z/OS V1R13 CPCC. It is also possible to change the workspace when the CPCC starts.

You must run a level of the CPCC client that corresponds to the z/OS system that you target. For example, run the z/OS V1R13 CPCC client on the workstation when interacting with the z/OS V1R13 Capacity Provisioning Manager, and run the z/OS V1R12 CPCC client on the workstation when interacting with the z/OS V1R12 Capacity Provisioning Manager.

Migration action:

- Install z/OS V1R13 CPCC using information in documentation referenced below.

  Note: With z/OS V1R13, the location of the default workspace changes from, for example,
  C:\Documents and Settings\user\Application Data\IBM\IBM Capacity Provisioning Control Center
  To C:\Documents and Settings\user\Application Data\IBM\IBM Capacity Provisioning Control Center V1R13\
  The actual directories depend on the Windows version that you are using.
- If you want to continue using your existing (pre-z/OS V1R13) CPCC workspace, point to the desired workspace during the installation process. You can also switch the workspace at a later time by navigating to the desired workspace when the CPCC starts.

Evaluate applications that parse AMBLIST command LISTLOAD or LISTIDR output (Required-IF, as of R12)

Required if your application depends on the output generated by the AMBLIST command LISTLOAD or LISTIDR control statement processing.

1. Prior to z/OS V1R12, the AMBLIST command LISTLOAD output for load modules, in the MODLIST area of the output, AMBLIST printed only the hexadecimal representation of TEXT records. Beginning with z/OS V1R12, AMBLIST command LISTLOAD output for load modules is made to work similarly to the same output for program objects. In particular, the output now includes the TEXT records output in EBCDIC format to the right of the hexadecimal representation. This also means that some of the columns for the hexadecimal output have been shifted.
2. Prior to z/OS V1R12, the AMBLIST commands LISTLOAD and LISTIDR output header for UNIX program objects printed the constant ****UNIX**** for both MEMBER NAME and LIBRARY, in the output header for each processed UNIX program object. Beginning in z/OS V1R12, AMBLIST commands LISTLOAD and LISTIDR output header for UNIX program objects display the actual directory path name and file name of the UNIX file. It is represented this way regardless of whether or not the MEMBER is specified as a relative path name. Note that MEMBER NAME and LIBRARY still only utilize a single output line each, therefore, if either exceeds the number of allotted characters they will be truncated on the left and preceded by two periods and a space (..).

Migration action: Output from the AMBLIST command is not an intended programming interface.

1. Evaluate applications that parse AMBLIST command LISTLOAD output to ensure that either there is no dependency on the TEXT output alignment or the additional data to the right of the hexadecimal output.
2. Evaluate applications that parse AMBLIST command LISTLOAD or LISTIDR output to ensure that there is no dependency on either the MEMBER NAME or LIBRARY in the output header being the constant ****UNIX**** or any other particular format.

To reduce future impact and maintenance, IBM suggests migrating the parsing routines or applications to use an IBM provided programming interface such as the Program Management APIs. See Using the binder application programming interfaces (APIs) in*z/OS MVS Program Management: Advanced Facilities*.

Ensure analysis tools interacting with HIS output accommodate HIS state change events (Required-IF, as of R12)

Required if you start HIS and have programs that analyze the HIS output.

Hardware instrumentation services (HIS) is a function that collects hardware event data for processors in SMF records type 113, subtype 2, as well as UNIX System Services output files. You can only use HIS for IBM System z10 or later machines. In z/OS V1R12 (and in z/OS V1R11 and z/OS V1R10 with the PTF for APAR OA30486 installed), functionality is added to the HIS component which causes changes in the output filename formats produced by HIS (.CNT, .SMP, .MAP), as well as introducing additional lines to the .CNT file possibly causing incompatibilities. In addition, an increase in SMF Type 113 records might be noticed.

Any tools that programmatically open the HIS output files (.CNT, .MAP, or .SMP), and any tools that programmatically analyze the HIS output .CNT file, should be analyzed and updated to accommodate the new formats.

Migration action: What to look for:

- The new filename output format is an indication that the support is installed.
- The first line of the .CNT file indicates output version 2.
- New output message HIS032I “STATE CHANGE DETECTED ACTION=action.”
- An additional line in the output of DISPLAY HIS command, which describes the action that should be taken should a state change event occur (specified by the operator at the start of a collection run).

What to do:

- If programmatically opening files, ensure the new output file format is handled.
- If programmatically parsing the .CNT file, ensure to check the VERSION identifier in the header. If the identifier is VERSION 2, be prepared for the new STATECHANGE line.

Detect program objects that have multiple INITIAL LOAD segments (Required-IF, as of R12)

Required if the following conditions Only users or ISVs that produce program object programs, which might have multiple INITIAL LOAD segments, need to take action if all of the items listed below are true:

- *Program is required to reside in a program object.*
- *Program has multiple segments.*
- *Program has multiple initial load classes.*
- *Program has mixed RMODEs.*

- *Program is link-edited with the RMODE option to override the binder default.*

Note: In most cases, even if a program has multiple segments containing INITIAL LOAD classes, no action is required.

Prior to z/OS V1R12, the binder RMODE option only applied to the first module segment, which contains some but possibly not all the initial load classes, though always the class wherein lies the entry point. Subsequent segments containing other initial load classes were not affected by the RMODE option, and thus the binder determined the RMODE based on the attributes of the classes contained therein.

Beginning with z/OS V1R12, the binder RMODE option applies to all initial load classes by default. Thus all segments containing initial load classes are affected. This new behavior takes affect only when the RMODE binder option is specified.

Also the RMODE option has been expanded so that either the new or previous behavior can be explicitly requested.

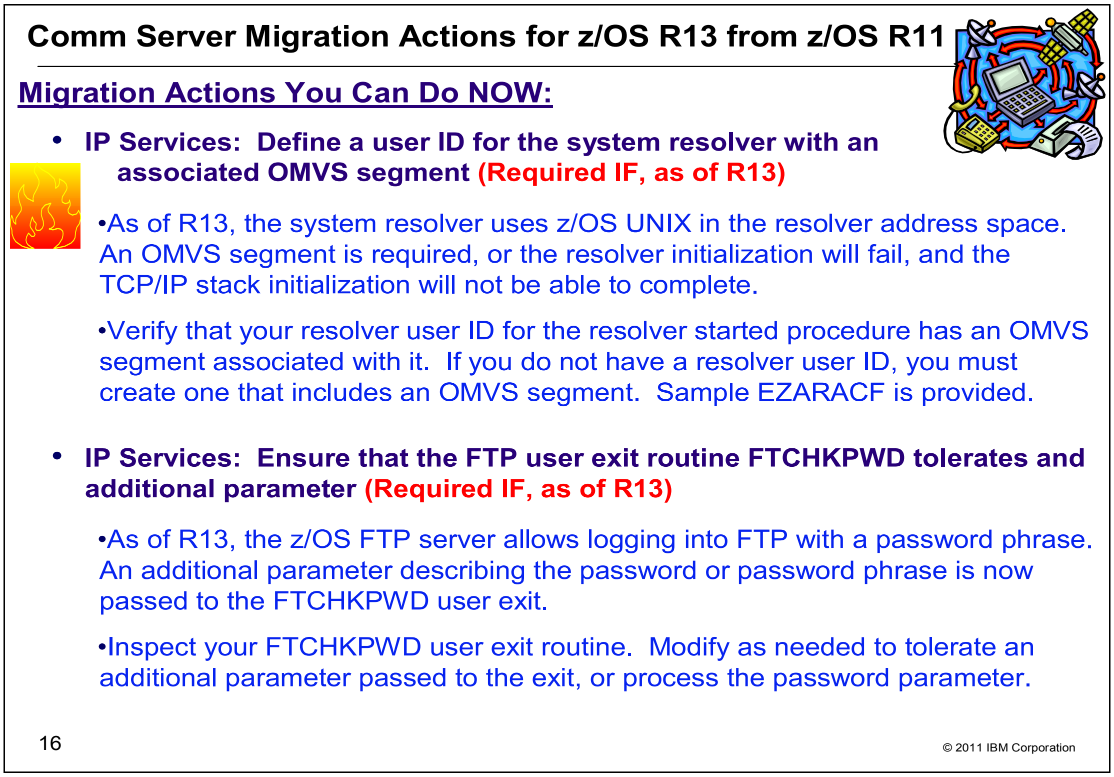
Migration action: To determine whether you might be affected by this change, you can run the AMBLIST service aid against your program objects and examine the output. However, if you do not explicitly specify the RMODE option when you bind the program, it is not affected even if the following two conditions are true. Run AMBLIST using the LISTLOAD control statement. Using the OUTPUT=MAP option results in a smaller amount of output that still contains the pertinent information.

The following two conditions must both be true for the program to be affected by
the new RMODE option behavior:

1. In the Module Summary, only programs that are program objects and are PO FORMAT: 3 or higher are affected.
2. In the SEGMENT MAP, only programs for which there is class entry SEGMENT 2, and TYPE INITIAL, are affected. **Note:** The RMODE differs for the SEGMENT 1, INITIAL load classes, and the SEGMENT 2, INITIAL load classes.

Given that these two conditions are true, there are then generally three possible situations:

1. If the RMODE option is not being specified, there is no need to do anything, as the behavior is identical in z/OS V1R12.
2. If you specified RMODE=ANY, expected all segments to be RMODE=31, but find that one segment is RMODE=24, part of the program is using below-the-line storage. In most cases, this is not desirable and the new binder behavior will remedy the situation. It is possible that you rely on having part of the program use below-the-line storage. You can use RMODE(ANY,COMPAT) to revert to the pre-z/OS V1R12 behavior.
3. If you specified RMODE=24, expected all segments to be RMODE=24, but find that one segment is RMODE=31, part of the program is using above-the-line storage. In most cases, this is desirable and the new binder behavior could cause a problem by causing the program to use more below-the-line storage. It is also possible that you rely on having part of the program use above-the-line storage. You can use RMODE(24,COMPAT) to revert to the pre-z/OS V1R12 behavior.



Comm Server Migration Actions for z/OS R13 from z/OS R11

Migration Actions You Can Do NOW:

- **IP Services: Define a user ID for the system resolver with an associated OMVS segment (Required IF, as of R13)**

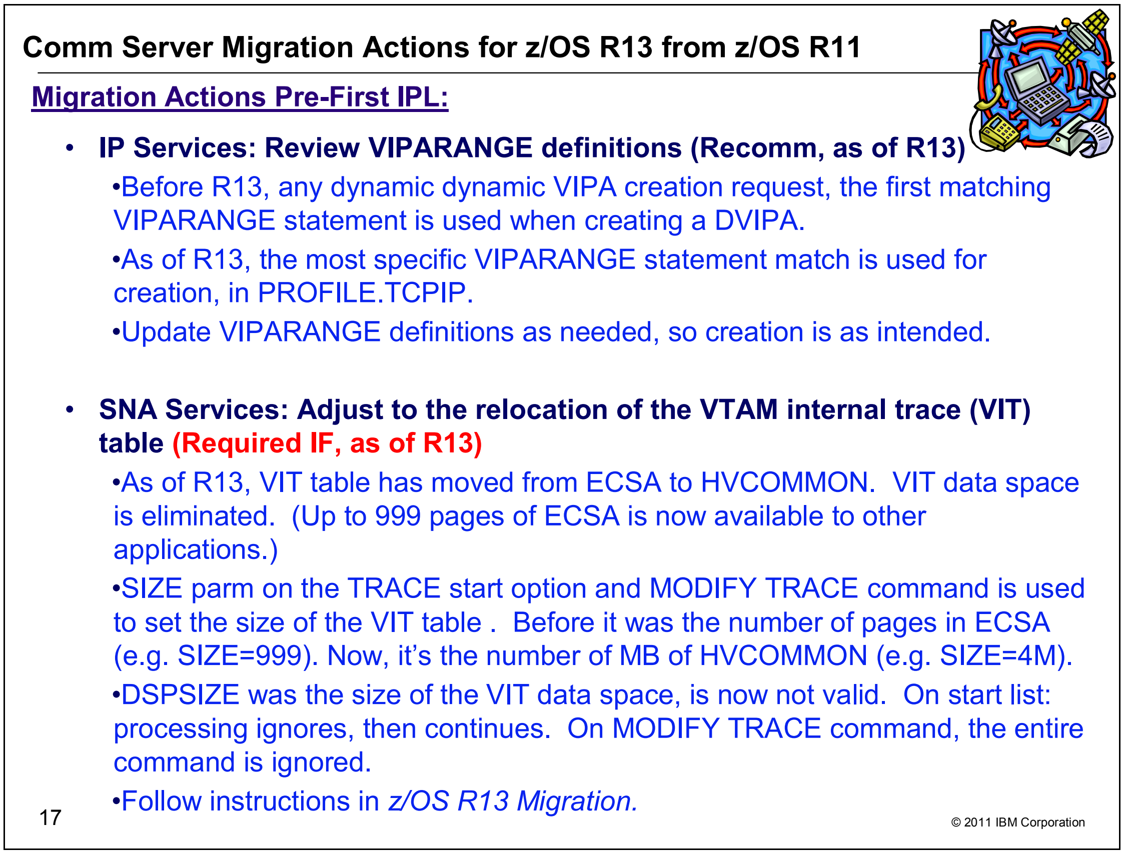
  •As of R13, the system resolver uses z/OS UNIX in the resolver address space. An OMVS segment is required, or the resolver initialization will fail, and the TCP/IP stack initialization will not be able to complete.

  •Verify that your resolver user ID for the resolver started procedure has an OMVS segment associated with it. If you do not have a resolver user ID, you must create one that includes an OMVS segment. Sample EZARACF is provided.

- **IP Services: Ensure that the FTP user exit routine FTCHKPWD tolerates and additional parameter (Required IF, as of R13)**

  •As of R13, the z/OS FTP server allows logging into FTP with a password phrase. An additional parameter describing the password or password phrase is now passed to the FTCHKPWD user exit.

  •Inspect your FTCHKPWD user exit routine. Modify as needed to tolerate an additional parameter passed to the exit, or process the password parameter.



Comm Server Migration Actions for z/OS R13 from z/OS R11

Migration Actions Pre-First IPL:

- **IP Services: Review VIPARANGE definitions (Recomm, as of R13)**

  •Before R13, any dynamic dynamic VIPA creation request, the first matching VIPARANGE statement is used when creating a DVIPA.

  •As of R13, the most specific VIPARANGE statement match is used for creation, in PROFILE.TCPIP.

  •Update VIPARANGE definitions as needed, so creation is as intended.

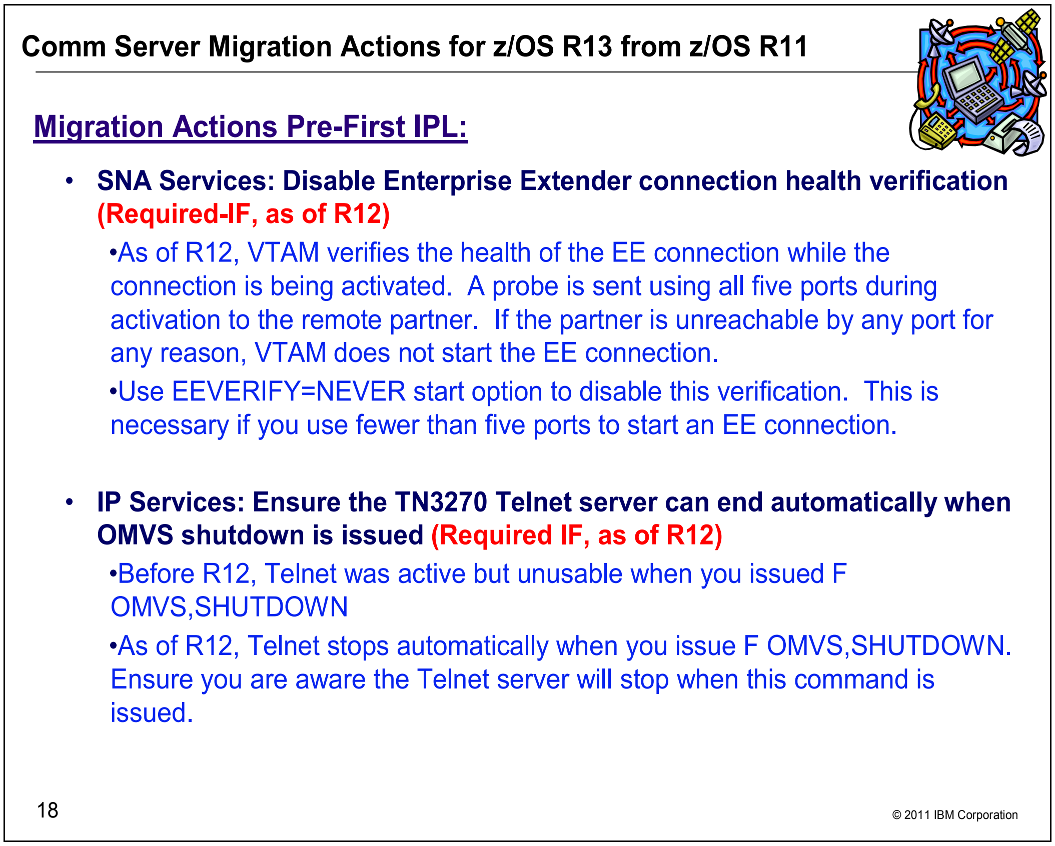
- **SNA Services: Adjust to the relocation of the VTAM internal trace (VIT) table (Required IF, as of R13)**

  •As of R13, VIT table has moved from ECSA to HVCOMMON. VIT data space is eliminated. (Up to 999 pages of ECSA is now available to other applications.)

  •SIZE parm on the TRACE start option and MODIFY TRACE command is used to set the size of the VIT table . Before it was the number of pages in ECSA (e.g. SIZE=999). Now, it’s the number of MB of HVCOMMON (e.g. SIZE=4M).

  •DSPSIZE was the size of the VIT data space, is now not valid. On start list: processing ignores, then continues. On MODIFY TRACE command, the entire command is ignored.

  •Follow instructions in *z/OS R13 Migration.*



Comm Server Migration Actions for z/OS R13 from z/OS R11

Migration Actions Pre-First IPL:

- **SNA Services: Disable Enterprise Extender connection health verification (Required-IF, as of R12)**
  - As of R12, VTAM verifies the health of the EE connection while the connection is being activated. A probe is sent using all five ports during activation to the remote partner. If the partner is unreachable by any port for any reason, VTAM does not start the EE connection.
  - Use EEVERIFY=NEVER start option to disable this verification. This is necessary if you use fewer than five ports to start an EE connection.

- **IP Services: Ensure the TN3270 Telnet server can end automatically when OMVS shutdown is issued (Required IF, as of R12)**
  - Before R12, Telnet was active but unusable when you issued F OMVS,SHUTDOWN
  - As of R12, Telnet stops automatically when you issue F OMVS,SHUTDOWN. Ensure you are aware the Telnet server will stop when this command is issued.

18

© 2011 IBM Corporation

Communications Server Migration Actions Between z/OS V1 R11 and z/OS V1 R13

These migration actions were taken from *z/OS Migration.* Many descriptions and actions have been severely shortened for inclusion in this presentation. For the complete descriptions and actions, refer to *z/OS Migration.*

Communications Server Migration Actions You Can Do Now

IP Services: Define a user ID for the system resolver with an associated OMVS segment (Required-IF, as of R13)

Required if you do not have a user ID defined for the system resolver that has an associated OMVS segment, which provides access to z/OS UNIX services.

Starting with z/OS V1R13, the system resolver uses z/OS UNIX services in the resolver address space. Use of z/OS UNIX services requires the resolver to have an OMVS segment associated with its user ID. If you do not define a user ID for the resolver with an associated OMVS segment, the resolver initialization will fail and the TCP/IP stack initialization will not be able to complete.

Migration action:

1. If you already have a user ID for the system resolver started procedure, and you explicitly defined an OMVS segment for the ID, or an OMVS segment was created automatically through the RACF automated assignment of unique UNIX identities support or through default OMVS segment support, no action is required.
2. If you already have a resolver user ID but it does not have an OMVS segment, you must define an OMVS segment for the resolver user ID.
3. If you do not have a resolver user ID, you must create one that includes an OMVS segment.

There is a sample job provided (EZARACF in data set SEZAINST) that you can use to define a user ID with an OMVS segment and assign it to the system resolver started procedure.

IP Services: Ensure storage availability for ancillary input queue for Enterprise Extender traffic (Recommended, as of R13)

Not required, but recommended if you have the WORKLOADQ parameter specified on the INTERFACE statement and you have concerns about using additional storage.
In z/OS V1R13, the processing of IPAQENET and IPAQENET6 INTERFACE statements is enhanced. Coding the WORKLOADQ parameter on these INTERFACE statements now enables the QDIO inbound workload queueing function for Enterprise Extender (EE) traffic. An additional ancillary input queue (AIQ) is established for inbound traffic for EE if HPR=RTP is specified as a VTAM start option. Each AIQ increases storage utilization by an amount equal to 36K of fixed ECSA, plus potentially the READSTORAGE value (64K multiplied by the number of SBALs) of fixed CSM 4K data space storage. If you have configured QDIO inbound workload queuing, you must ensure that there is sufficient fixed ECSA and 4K CSM dataspace storage for the AIQ for EE traffic.
Each AIQ increases storage utilization by an amount equal to 36K of fixed ECSA plus potentially the READSTORAGE value (64K multiplied by the number of SBALs) of fixed CSM 4K data space storage.
Migration action:

1. Verify if you are using EE; you are using EE if HPR=RTP is defined as a VTAM start option and if an EE XCA Major Node is defined and active. If you are using EE, continue with steps 2-5. If you have HPR=RTP defined as a VTAM start option but do not have an EE XCA Major Node defined and active, continue with steps 2, 3 and 5. Otherwise there is no increase in storage usage and no further action is required.
2. Count the total number of OSA-Express3, and later, interfaces that are coded with the WORKLOADQ parameter on the IPAQENET and IPAQENET6 INTERFACE statements. Make a note of the number.
3. Verify that sufficient ECSA is available. To do this, multiply the total number of OSA-Express3, and later, interfaces that have inbound workload queueing enabled (you determined this number in step 2) by 36K. The resulting number indicates how much new ECSA is required. Use the DISPLAY CSM command to verify that sufficient ECSA is available to enable this function.
4. Verify that sufficient real storage is available. 64-bit real storage is used for the dataspace read buffers. Multiply the total number of OSA-Express3, and later, interfaces that have inbound workload queueing enabled (determined in step 2) by 64K and by 126 (8064K). The maximum number of read buffers per queue is 126. The resulting number is approximately the amount of additional 64-bit real storage that is required for the data space read buffers for all the new EE input queues.
5. If sufficient storage is not available, either increase the available storage or consider defining some of the OSA-Express3, and later, INTERFACE statements with the NOWORKLOADQ parameter.

IP Services: Permit IKE daemon running in FIPS mode to use additional ICSF services (Required-IF, as of R13)

Not required, but recommended if you currently run the IKE daemon in FIPS mode and if you control the access to ICSF resources in the CSFSERV class.
In z/OS V1R13, the Internet Key Exchange (IKE) daemon is enhanced to take advantage of new services that are provided by Integrated Cryptographic Service Facility (ICSF) when the IKE daemon is running in Federal Information Processing Standards (FIPS) mode. The new ICSF services are provided in updates to ICSF PKCS number 11 functions CSFPDVK and CSFPDMK. ICSF now provides the following information to the IKE daemon, each with a single call to ICSF:

- The derivation of the original seed key.
- The phase 1 key set.
- The phase 2 key set

Migration action: The IKE daemon now requires READ access to the CSF1DVK and CSF1DMK resources in CSFSERV when the IKE daemon is configured to run in FIPS mode.
If your security server is RACF, issue the following commands in the order shown. If you use a different security server, determine and perform the equivalent steps.

1. PERMIT CSF1DVK CLASS(CSFSERV) ID(IKED) ACCESS(READ)
2. PERMIT CSF1DMK CLASS(CSFSERV) ID(IKED) ACCESS(READ)
3. SETROPTS RACLIST(CSFSERV) REFRESH

IP Services: Migrate from DNS BIND 9.2.0 (Recommended, but will be Required after z/OS R13)

Not required, but recommended because this action is planned to become a requirement in the release after z/OS R13.
In the release after z/OS V1R13, support for BIND DNS 9.2.0 is planned to be discontinued. If you are using this name server, you are encouraged to find a replacement as soon as you can.
Migration action:

- If you have been using BIND DNS 9.2.0 as a caching-only name server, use the z/OS resolver DNS caching function in z/OS V1R11 instead.

- If you have been using BIND DNS 9.2.0 as a primary or secondary authoritative name server, migrate to BIND DNS on Linux for System z.

Tip: On z/OS V1R11 and later, use IBM Health Checker for z/OS check ZOSMIGV1R11_CS_BIND9 for DNS 9.2.0 to determine if a BIND DNS name server function is in use.

IP Services: Understand and prepare for expanded Intrusion Detection Services (Required-IF, as of R13)

Required if you are using IDS on a stack that is being run as a dual-mode stack (IPv4 and IPv6).

Beginning in z/OS V1R13, Intrusion Detection Services (IDS) is enhanced to monitor IPv6 traffic. This includes scan detection and reporting, attack detection, attack reporting, attack prevention, and traffic regulation. Additional attack detection, reporting, and prevention are also provided for IPv4 traffic.

Migration action:

1. If any of the following IDS attack types are enabled, be aware that both IPv4 and IPv6 traffic will be monitored for attacks of these types:
   a. Malformed attack type
   b. UDP perpetual echo attack type
   c. Flood attack type
   d. ICMP redirect attack type
2. If you use the **trmdstat** command to get a consolidated view of IDS log records, be aware that the default report, provided when the **trmdstat** command is issued with no report option, will be the IDS Summary report. The IDS Summary report provides a summary of all IDS information.
3. If you have a TCP scan rule that applies to all local IP addresses (such as when the LocalHostAddr All is specified explicitly or by default in the policy), then TCP scan events will be detected for both IPv4 and IPv6 packets. If you want the TCP scan rule to continue to only detect scan events for IPv4 packets, modify the rule to specify a local IP address of 0.0.0.0/0.
4. If you have a UDP scan rule that applies to all local IP addresses (such as when the LocalHostAddr All is specified explicitly or by default in the policy), then UDP scan events will be detected for both IPv4 and IPv6 packets. If you want the UDP scan rule to continue to only detect scan events for IPv4 packets, modify the rule to specify a local IP address of 0.0.0.0/0.
5. If you have a TCP traffic regulation (TR) rule that applies to all local IP addresses (such as when the LocalHostAddr All is specified explicitly or by default in the policy), then limits will be enforced for both IPv4 and IPv6 connection requests. If you want the TCP TR rule to continue to only enforce limits for IPv4 connection requests, modify the rule to specify a local IP address of 0.0.0.0/0.
6. If you have a UDP TR rule that applies to all local IP addresses (such as when the LocalHostAddr All is specified explicitly or by default in the policy), then limits will be enforced for both IPv4 and IPv6 packets. If you want the UDP TR rule to continue to only enforce limits for IPv4 packets, modify the rule to specify a local IP address of 0.0.0.0/0.
7. If you use LDAP to configure IDS policy and you are using the default value for attribute ibm-idsLocalHostIPAddress with a TCP scan, UDP scan, TCP TR, or UDP TR rule, events will be detected and limits will be enforced for both IPv4 and IPv6 traffic. If you want these rules to continue to apply to only IPv4 traffic, modify the attribute to specify ibm-idsLocalHostIPAddress:3-0.0.0.0-0.

IP Services: Ensure that the FTP user exit routine FTCHKPWD tolerates an additional parameter (Required-IF, as of R13)

Required if your installation uses the FTCHKPWD user exit routine, and if one of the following conditions is true;

1. *Your exit routine cannot tolerate an additional parameter passed to the exit routine.*
2. *Your exit routine inspects or processes the password parameter in any way.*

Starting in z/OS V1R13, the z/OS FTP server is enhanced to allow logging into FTP with a password phrase. An additional parameter describing the password or password phrase that is used to log into the z/OS FTP server is now passed to the FTCHKPWD user exit. If you have installed an FTCHKPWD exit routine, and your exit routine meets one or both criteria listed above, then you must take action.

Migration action: Inspect your FTCHKPWD user exit routine. Modify as needed to support the additional parameter.

IP Services: Understand change in VIPARANGE security verification processing (Required-IF, as of R13)

Required if you have defined the EZB.BINDDVIPARANGE.sysname.tcpname resource profile, but you have not given READ access to this resource to all applications that create DVIPAs by binding to addresses that are within a VIPARANGE subnet.

Prior to z/OS V1R13, the System Authorization Facility (SAF) resource profile associated with the VIPARANGE statement (EZB.BINDDVIPARANGE.*sysname.tcpname*) was ignored in the following scenario when an application issues a bind:

- The port specified on the bind matches a PORT statement, and
- the IP address of the application's bind is within a VIPARANGE subnet, or the application's bind is an unspecified address and the IP address on the BIND parameter of the PORT statement is within a VIPARANGE subnet.

In this scenario, the PORT statement (including its optional SAF parameter) was used to control access to both the port and to creating the dynamic VIPA (DVIPA).

Beginning in V1R13, the VIPARANGE resource profile is not ignored in this scenario; creation of the IP address is controlled by both the SAF resource profile associated with the VIPARANGE statement and by the PORT statement:

- For a VIPARANGE statement, you can control the creation of the IP address by defining the following SAF resource profiles in the SERVAUTH class:
  - EZB.BINDDVIPARANGE.*sysname.tcpname*
  - EZB.BINDDVIPARANGE.*sysname.tcpname.resname*, if the new SAF parameter is included on the VIPARANGE statement
- The PORT statement controls whether an application can bind to a given port.

Migration action: Be aware that the EZB.BINDDVIPARANGE.*sysname.tcpname* resource profile is always checked if defined; ensure all applications that create DVIPAs by binding to addresses within a VIPARANGE subnet have READ access to this resource.

IP Services: Update IP filter policy to filter IP fragments correctly for RFC 4301 compliance (Required-IF, as of R12)

Required if your IP filter policy selectively matches routed traffic based on TCP port, UDP port, ICMP type and code, ICMPv6 type and code, OSPF type, or MIPv6 type. If you configured the RFC4301Compliance parameter on the IpFilterPolicy statement, it is recommended that you remove the parameter because it is deprecated beginning with z/OS V1R12.

Beginning with z/OS V1R12, all IP security filters must be compliant with RFC 4301. You can no longer use the RFC4301Compliance parameter on the IpFilterPolicy statement to specify whether Policy Agent enforces compliance. The RFC4301Compliance parameter is ignored and Policy Agent enforces the rule that ensures all IP filters are compliant.

IP filter policy support for filtering fragments was improved in z/OS V1R10. Before z/OS V1R10, Communications Server filtered all IP fragments using a policy of first possible filter match and filtered IPv6 fragments as protocol IPv6Frag. Beginning with z/OS V1R10, Communications Server follows rules and restrictions established by RFC 4301 http://www.ietf.org/rfc/rfc4301.txt) to ensure proper classification of fragments. RFC 4301, “Security Architecture for the Internet Protocol”, specifies the base architecture for IPSec-compliant systems, including restrictions on the routing of fragmented packets.

Communications Server does not implement stateful fragment checking, therefore, restrictions were added as required by RFC 4301 to require that IP filter rules for routed traffic do not allow specific ports, types, or codes. The configuration parameter RFC4301Compliance could be used in z/OS V1R10 and z/OS V1R11 to optionally configure whether the RFC 4301 rules should be applied. Beginning with z/OS V1R12, this parameter (RFC4301Compliance on the IpFilterPolicy statement) is deprecated. All IP filter rules must support the RFC 4301 rules and restrictions.

Migration action: Follow the steps documented in *z/OS R13 Migration*.

Tip: Use IBM Health Checker for z/OS check ZOSMIGV1R11_CS_RFC4301 to determine whether IPSec filter rules are compliant with RFC4301. This check is available in z/OS V1R10 with the installation of PTF UA47191 (APAR OA28605).

IP Services: Remove customization of SNMP sysObjectID MIB object in OSNMPD.DATA File (Recommended)

Recommended because the ability to customize the sysObjectID value will not be supported in a future release.

The SNMP agent allows you to provide some initial settings for a small set of MIB objects by using the OSNMPD.DATA file. One of the objects for which an initial value can be provided is sysObjectID.0. The sysObjectID.0 object is the vendor’s authoritative identification of the network management subsystem contained in the entity. That is, it is intended to uniquely identify the SNMP agent. Changing this value is not recommended and the ability to change it will be disabled in a future release. As of z/OS V1R4, warning message EZZ6317I is written to the syslog daemon if the object is set by using the OSNMPD.DATA file. As of z/OS V1R8, message EZZ6317I is also written to the console.

Migration action: Review the statements in your OSNMPD.DATA configuration file. If this file contains a statement for the sysObjectID object, remove the statement from the file.

IP Services: Restore resolver UDP request timeout interval duration (Required-IF, as of R12)

Required if your network requires longer than 5 seconds for resolver queries to be processed by a name server under normal network conditions.

Prior to z/OS V1R12, the default setting for considering UDP requests to a name server to have timed out (as specified by the TCPIP.DATA RESOLVERTIMEOUT statement) was 30 seconds. Starting with z/OS V1R12, the default is 5 seconds. If you want the resolver to continue to use the old value, you must take action.

Migration action:

- Code the RESOLVERTIMEOUT 30 statement in each TCPIP.DATA dataset where the previous UDP timeout duration interval is needed.
- If the resolver is active when all the TCPIP.DATA datasets are updated, issue the MODIFY *resolver*,REFRESH command to cause the resolver to refresh the TCPIP.DATA settings.
- If the resolver is not active, start the resolver.

IP Services: Ensure applications tolerate a larger addrinfo structure (Required-IF, as of R12)

Required if your application has a dependency on the size of the struct addrinfo structures returned by the getaddrinfo() subroutine points.

The addrinfo structure is used by the getaddrinfo() function to hold host address information. In z/OS V1R12, the addrinfo structure (struct addrinfo) is enhanced to comply with RFC 5014. As a result, the struct addrinfo is larger than it was prior to z/OS V1R12. When the res parameter returned by getaddrinfo() is a chain of pointers to struct addrinfo structures, the pointers point to the larger struct addrinfo.

Migration action: Be aware that the res parameter returned by the getaddrinfo() subroutine will point to larger struct addrinfo structures. Change your application as necessary to accommodate larger struct addrinfo structures.

IP Services: Release addrinfo storage after resolver thread task terminates (Recommended, as of R12)

Not required, but recommended to prevent excessive consumption of application virtual storage by resolver functions.

The resolver allocates storage in the application address so contain the addrinfo data returned by the getaddrinfo() function. Prior to z/OS V1R12, the resolver released this storage when the task that issued the getaddrinfo() function terminated. With z/OS V1R12, the resolver releases the addrinfo storage when the process task that owns the task that issued the getaddrinfo() function terminates. Applications with long-running tasks that do not explicitly release addrinfo storage using the freeaddrinfo() function might experience an increase in storage usage.

Migration action: Update application source code to ensure that the freeaddrinfo() function is issued to release all addrinfo storage obtained using the getaddrinfo() function.

IP Services: Update syslogd configuration for archiving rules with shared z/OS UNIX file destinations (Required-IF, as of R12)

Required if you configure multiple syslogd rules that use the same z/OS UNIX file destination, and if you configure automatic archiving for that destination file.

Starting with z/OS V1R11, you can configure the syslogd daemon (syslogd) to perform automatic archiving for z/OS UNIX files. You do this by specifying the -N parameter on the appropriate syslogd rules in the configuration file, along with specific configuration statements for the archive function. Beginning with z/OS V1R12, if you use the same z/OS UNIX file destination on multiple syslogd rules, the automatic archiving function is not supported on any of these rules.

Migration action: If you configure multiple syslogd rules that use the same z/OS UNIX file destination, and configure automatic archiving for that destination file, perform the following steps:

- If you do not require automatic archiving, remove the -N parameter from all rules that share the same z/OS UNIX file destination.
- If you do require automatic archiving, change the destination to a unique z/OS UNIX file for each rule that shares the same z/OS UNIX file destination.

Note: If you do not perform either of these steps, message `FSUM1273` *`jobname`* `AUTOMATIC ARCHIVE NOT USED FOR RULES WITH SHARED DESTINATION` is issued to the operator console, and one or more instances of message `FSUM1272 WARNING: ARCHIVE FUNCTION DISABLED FOR RULES WITH SHARED DESTINATION` *filename* is written to the syslogd error destination. The automatic archiving function is turned off for all syslogd rules that share z/OS UNIX file destinations. Although automatic archiving is turned off, syslogd continues to log messages to the destination file.

SNA Services: Ensure IVTCSM ASSIGN_BUFFER requests do not exceed 500 images for a single CSM buffer (Recommended, as of R13)

No required, but recommended. Your application probably does not request more than 500 images of a CSM buffer; however, you should examine IVTCSM ASSIGN_BUFFER calls to ensure that this is the case.

Beginning with z/OS V1R13, Communications Storage Manager (CSM) will reject the ASSIGN_BUFFER request after 500 image buffers of a single CSM buffer are requested, and CSM will return a new reason code of 26. The new reason code 26 states: "Assign buffer request failed because CSM reached the limit of the maximum number of image buffers of the single CSM buffer."

Migration action: To prevent application failures due to excessive ASSIGN_BUFFER requests:

- Identify any authorized applications that use CSM services and verify that the number of image buffers requested by IVTCSM ASSIGN_BUFFER for a single CSM buffer can never be more than 500.
- If necessary, change the application so the total number of IVTCSM ASSIGN_BUFFER image buffer requests is 500 or less.

SNA Services: Ensure VTAMSG2 in not used in your VTAMLST definitions (Required-IF, as R12)

Required if you have used the name VTAMSG2 in any of your VTAM definition statements.

Starting in z/OS V1R12, VTAMSG2 is a new reserved name of an internal major node built by VTAM. As such, there should be no other use of the name VTAMSG2 in the VTAM definitions that are used by your system.

Migration action: Investigate all your VTAM definitions. If you find the name VTAMSG2 in any of them, change the name to a non-reserved name. Note that if you continue to configure and try to activate VTAMSG2, message `IST607I VARY ACT FOR VTAMSG2 FAILED – INVALID NODE TYPE OR STATE` will be issued.

Communications Server Migration Actions Pre-First IPL

IP Services: Review VIPARANGE definitions (Recommended, as of R13)

Not required, but recommended to ensure DVIPAs are created as intended.

Prior to z/OS V1R13, for any dynamic VIPA (DVIPA) creation request, the first matching VIPARANGE statement is used when creating a DVIPA. Beginning in z/OS V1R13, the most specific VIPARANGE statement match is used when creating a DVIPA.

Migration action: Be aware of the change in VIPARANGE processing. Update VIPARANGE definitions as needed.

IP Services: Update automation that keys on TN3270E Telnet server messages (Req-IF, as of R12)

Required if you have automation checking Telnet server messages.

Prior to z/OS V1R12, all Telnet messages began with the word "TELNET" after the message identifier. The DEBUG, LUNS, and LUNR messages began with "TELNET *jobname*" after the message identifier. Starting with z/OS V1R12, the first word "TELNET" is replaced with the jobname. For the DEBUG, LUNS, and LUNR messages, all other words in the message are shifted to the left. If you have automation which assumes certain positions for these messages, you must make accommodations to accept the new format of these messages.

Migration action: Ensure that no automation expects the first word of a Telnet message to be "TELNET" and ensure that any automation that keys on the DEBUG message EZZ6035I, or the LUNS and LUNR messages EZZ6091I - EZZ6099I, is updated to handle the word shift to the left.

IP Services: Ensure the TN3270E Telnet server can end automatically when an OMVS shutdown command is issued (Required-IF, as of R12)

Required if you issue an F OMVS,SHUTDOWN command when the Telnet application is active.

Prior to z/OS V1R12, if the Telnet application was active when you issued the F OMVS,SHUTDOWN command, Telnet continued to be active but in an unusable state. Starting with z/OS V1R12, Telnet stops automatically when you issue the F OMVS,SHUTDOWN command. Messages are issued in the order they appear below:

```
BPXI055I OMVS SHUTDOWN REQUEST ACCEPTED
EZZ6008I tnproc STOPPING
EZZ6010I tnproc SERVER ENDED FOR PORT 23
EZZ6009I tnproc SERVER STOPPED
```

Migration action: Review your procedures for issuing the F OMVS,SHUTDOWN command and be aware the Telnet server will stop when the command is issued.

IP Services: Disable resolver monitoring of name server responsiveness (Required-IF, as of R12)

Required if you do not want notification of potential problems with name servers in your network.

Prior to z/OS V1R12, the z/OS Communications Server system resolver did not provide notification of situations where a Domain Name System (DNS) name server was not responding to resolver queries sent to the name server.

Starting with z/OS V1R12, the resolver will notify the operator when an unresponsive name server is detected. This function provides critical diagnostic information to the network operator and allows the operator to take steps to avoid performance and availability problems. If you do not want the resolver to provide these notifications, you must take action.

Migration action:

- If you do not already have a resolver setup file, create one.
- Code the UNRESPONSIVETHRESHOLD(0) statement in a resolver setup file. A *percentage_of_queries* value of 0 indicates that resolver should not monitor name server responsiveness. The default percentage is 25% of resolver queries.
- If resolver is active, issue the MODIFY *resolver*,REFRESH,SETUP=*resolver_setup_file* command to cause the resolver to stop monitoring name server responsiveness.
- If the resolver is not active, start the resolver.

IP Services: Disable IP validation checks when defining key exchange policy rules for a dynamic VPN (Required-IF, as of R12)

Required if you want bypass the IP security key exchange validation check that verifies that the identity of the remote peer matches the IP address of the remote peer.

You can configure policy rules for IP security when defining key exchange policy for dynamic virtual private networks (DVPNs). Prior to z/OS V1R12, there was no automatic way to verify that the identity of the remote peer matched the IP address of the remote peer. Beginning with z/OS V1R12, such IP validation checks are enforced by default. If you do not want the IP validation checking enabled on your policy rules for security key exchanges, you must disable it.

Notes:

1. The new default to enable IP validation checking is a benefit; however, one example of why you might want to disable the checking is if your remote security endpoint is expected to be behind a network address translation (NAT) device.
2. A value coded on the parameter to disable the automatic IP validation checking is ignored if the identity of the peer is not an IPv4 or IPv6 address.

Migration action: Specify the value YES on the BypassIpValidation parameter of the KeyExchangePolicy or KeyExchangeAction statements. (The value of NO is the default).

IP Services: Update modified Netstat message catalogs to include timestamp (Req-IF, as of R12)

Required if you are customizing either the netmsg.cat or netmsg6.cat message catalog.

z/OS V1R12 enhances Netstat to verify that its message catalogs are at the same maintenance level as the Netstat program. If the message catalog time stamp does not match the Netstat program time stamp, Netstat uses the default messages. If you are customizing the netmsg.cat or netmsg6.cat message catalogs, you must take action to preserve the time stamp.

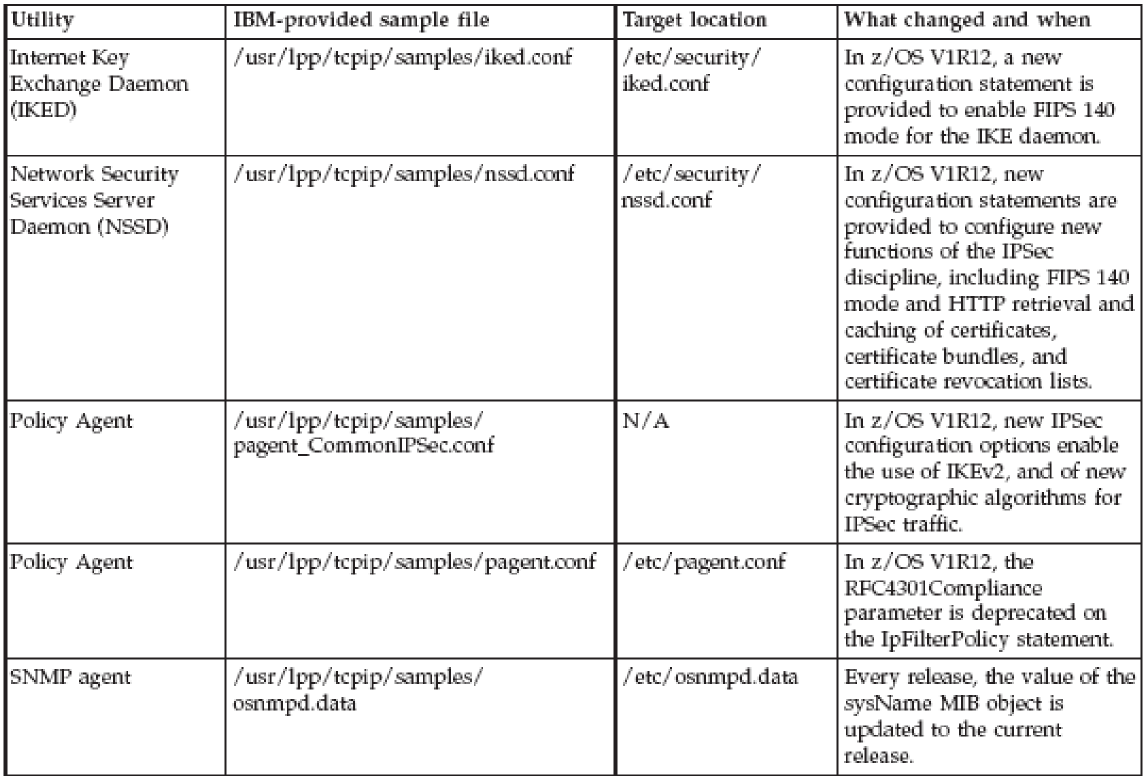
Migration action:

1. Issue the **dspcat** command with the -t -g options against the netmsg.cat (for example, issue **dspcat netmsg.cat -g -t**) to retrieve the necessary output to supply to the **gencat** command to generate your translated message catalog. The z/OS Communications Server netmsg.cat is installed in the z/OS UNIX file system directory /usr/lpp/tcpip/lib/nls/msg/C. The output file displays as follows:
   ```
   The time stamp of catalog netmsg.cat is: 2009 112 00:11 UTC
   $delset 1
   $set 1
   $quote "
   ```
2. Remove the first line of the output file generated by the **dspcat** command and replace it with a $timestamp directive and the time stamp value from the same output file. The new output file displays as follows:
   ```
   $timestamp 2009 112 00:11 UTC
   $delset 1
   $set 1
   $quote "
   ```
3. Make any other appropriate changes to customize your message catalog and issue the **gencat** command to generate the new message catalog.
4. Repeat steps 1-3 for the netmsg6.cat message catalog.

IP Services: Update /etc configuration files (Required-IF)

Required if you have customized a configuration file that IBM has changed.

Some utilities provided by Communications Server require the use of certain configuration files. You are responsible for providing these files if you expect to use the utilities. IBM provides default configuration files as samples in the /usr/lpp/tcpip/samples directory. Before the first use of any of these utilities, you should copy these IBM-provided samples to the /etc directory (in most cases). You can further customize these files to include installation-dependent information. An example is setting up the /etc/osnmpd.data file by copying the sample file from /usr/lpp/tcpip/samples/osnmpd.data to /etc/osnmpd.data and then customizing it for the installation.
If you customized any of the configuration files that have changed, then you must incorporate the customization into the new versions of the configuration files.
Migration action: If you added installation-dependent customization to any of the IBM-provided configuration files listed below, make the same changes in the new versions of the files by copying the IBM-provided samples to the files shown in the table and then customizing the files.



Utility	IBM-provided sample file	Target location	What changed and when
Internet Key Exchange Daemon (IKED)	/usr/lpp/tcpip/samples/iked.conf	/etc/security/ iked.conf	In z/OS V1R12, a new configuration statement is provided to enable FIPS 140 mode for the IKE daemon.
Network Security Services Server Daemon (NSSD)	/usr/lpp/tcpip/samples/nssd.conf	/etc/security/ nssd.conf	In z/OS V1R12, new configuration statements are provided to configure new functions of the IPSec discipline, including FIPS 140 mode and HTTP retrieval and caching of certificates, certificate bundles, and certificate revocation lists.
Policy Agent	/usr/lpp/tcpip/samples/ pagent_CommonIPSec.conf	N/A	In z/OS V1R12, new IPSec configuration options enable the use of IKEv2, and of new cryptographic algorithms for IPSec traffic.
Policy Agent	/usr/lpp/tcpip/samples/pagent.conf	/etc/pagent.conf	In z/OS V1R12, the RFC4301Compliance parameter is deprecated on the IpFilterPolicy statement.
SNMP agent	/usr/lpp/tcpip/samples/ osnmpd.data	/etc/osnmpd.data	Every release, the value of the sysName MIB object is updated to the current release.

SNA Services: Adjust to the relocation of the VTAM internal trace table (Required-IF, as of R13)

Required if one or more of the following conditions are true:

- *If automation of the MODIFY TRACE command exists and the command specifies SIZE in pages or DSPSIZE, and the failure of that command is unacceptable.*
- *If automated applications that parse retired messages not finding any of the newly retired messages is unacceptable.*

Starting with z/OS V1R13, the VTAM internal trace (VIT) table is relocated from ECSA to HVCOMMON and the VIT data space is eliminated. As a result, be aware of the following changes starting in z/OS V1R13:

- The SIZE parameter of the TRACE start option and the MODIFY TRACE command is used to set the size of the VTAM Internal Trace (VIT) table. Prior to z/OS V1R13, the value of the SIZE parameter specified the number of pages of ECSA (for example, SIZE=999). Starting with z/OS V1R13, the value of the SIZE parameter of the TRACE start option and the MODIFY TRACE command specifies the number of megabytes of HVCOMMON (for example, SIZE=4M).
- Starting with z/OS V1R13, the DSPSIZE parameter is not valid.
  - The DSPSIZE parameter was used to set the size of the VIT data space. Prior to z/OS V1R13, the value of the DSPSIZE parameter specified the number of megabytes of data space in 10 megabytes increments (for example, DSPSIZE=5 for 50 megabytes).
  - If DSPSIZE is coded in the VTAM start list, the following informational message is displayed: `IST448I DSPSIZE OPTION IGNORED - NO LONGER SUPPORTED`. Processing then continues disregarding the specification.

- If DSPSIZE is specified on a MODIFY TRACE command, the following informational message is displayed: `IST448I DSPSIZE OPTION IGNORED - NO LONGER SUPPORTED`. The entire MODIFY TRACE command is ignored.
- Numerous messages are no longer issued and are retired.

Valid SIZE values are 4M-2048M inclusive. If you specify a SIZE value that is larger than the default value, z/OS will perform paging on portions of the VIT table. Before you specify a large SIZE value, ensure that you have sufficient real or auxiliary storage to contain the entire VIT. Failure to ensure sufficient storage might result in an auxiliary storage shortage. If an SVC dump is taken that includes common storage, the size of the dump data set also increases. You must also take the increase in the size of the dump data set into consideration.

Migration action: Thoroughly read the complete migration action documented in *z/OS R13 Migration.* It is outlined below:

- Migrate your VTAM start lists. Convert your SIZE value to megabytes, and delete you DSPSIZE specification.
- Migrate your automated MODIFY TRACE commands. If SIZE or DSPSIZE is specified and you want to continue to issue the command, it must be changed.
- Migrate applications that parse retired messages. Migrate any automation that parses for or specifies the character string ISTITDS1, which was the data space name that contained the VIT data space.
- Notice that VIT associated health checks have been removed.

SNA Services: Disable Enterprise Extender connection health verification (Required-IF, as R12)

Required if you used fewer than five ports to start an EE connection in a previous release, you must code the start option EEVERIFY=NEVER before you start VTAM.

Starting in z/OS V1R12, VTAM verifies the health of the Enterprise Extender (EE) connection while the connection is being activated. VTAM sends a Logical Data Link Control (LDLC) probe to the remote partner using all five ports during the activation. If the partner is not reachable by any port for any reason, VTAM does not start the EE connection. VTAM issues the following health verification failure message group if the attempt to connect failed during the activation of the EE connection when VTAM sent the LDLC probe to the remote partner:

```
IST2330I EE HEALTH VERIFICATION FAILED FOR puname AT time
IST1680I type IP ADDRESS ip_address
IST1680I type IP ADDRESS ip_address
IST314I END
```

If you do not want VTAM to verify the health of the EE connection, code the start option EEVERIFY=NEVER before starting VTAM.

Migration action: If you are using less than five ports for all of your EE connections, then you need to specify the EEVERIFY=NEVER start option. If you have specific EE connections, you need to specify EEVERIFY=NEVER on the PU or GROUP statement for that EE connection.

SNA Services: Code MULTPATH start option when using multipath (Required-IF, as R12)

Required if you want to continue using multipath for EE connections.

Prior to z/OS V1R12, multipath support was enabled by a single switch in the TCP/IP profile for both Enterprise Extender (EE) and TCP/IP traffic. Beginning with z/OS V1R12, multipath support is disabled (by default) for EE connections. If you want to continue to use multipath for your EE connections, you must code the start option MULTPATH=TCPVALUE before starting VTAM.

Migration action: Add MULTPATH=TCPVALUE to the VTAM start option file ATCSTR*xx*.

Communications Server Migration Actions Post-First IPL

IP Services: Ensure that preference values associated with IPv6 router advertisement routes are as expected (Required, as of R12)

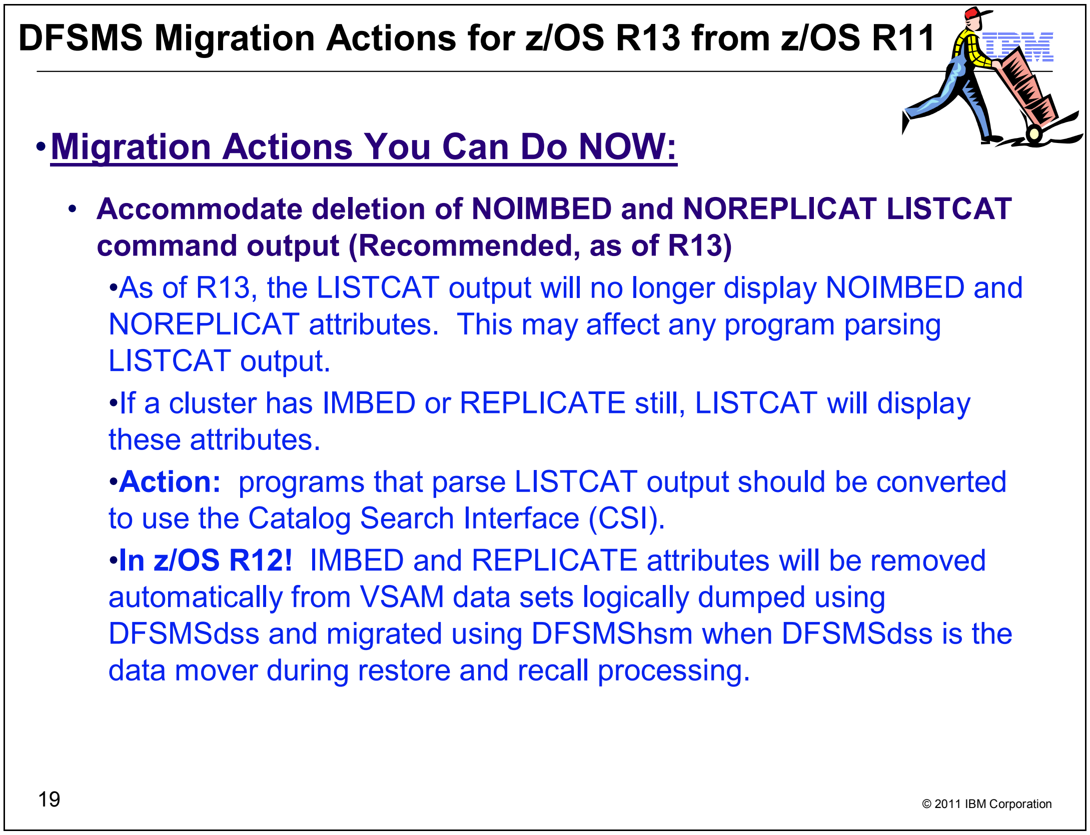
Required if your stack is IPv6 enabled and your adjacent routers are originating IPv6 router advertisement messages that contain preference values other than the default value (medium) for either the default route or an indirect prefix route.

Starting in z/OS V1R12, TCP/IP processes preference values that are provided in IPv6 router advertisement messages. Prior to z/OS V1R12, all router advertisement routes were generated with a medium preference value.

Migration action:

1. Review the reference information for a description of how z/OS Communications Server uses the preference values that are received in IPv6 router advertisement messages.
2. Ensure that the preference values associated with the router advertisement routes generated from information received in router advertisement messages are as expected.

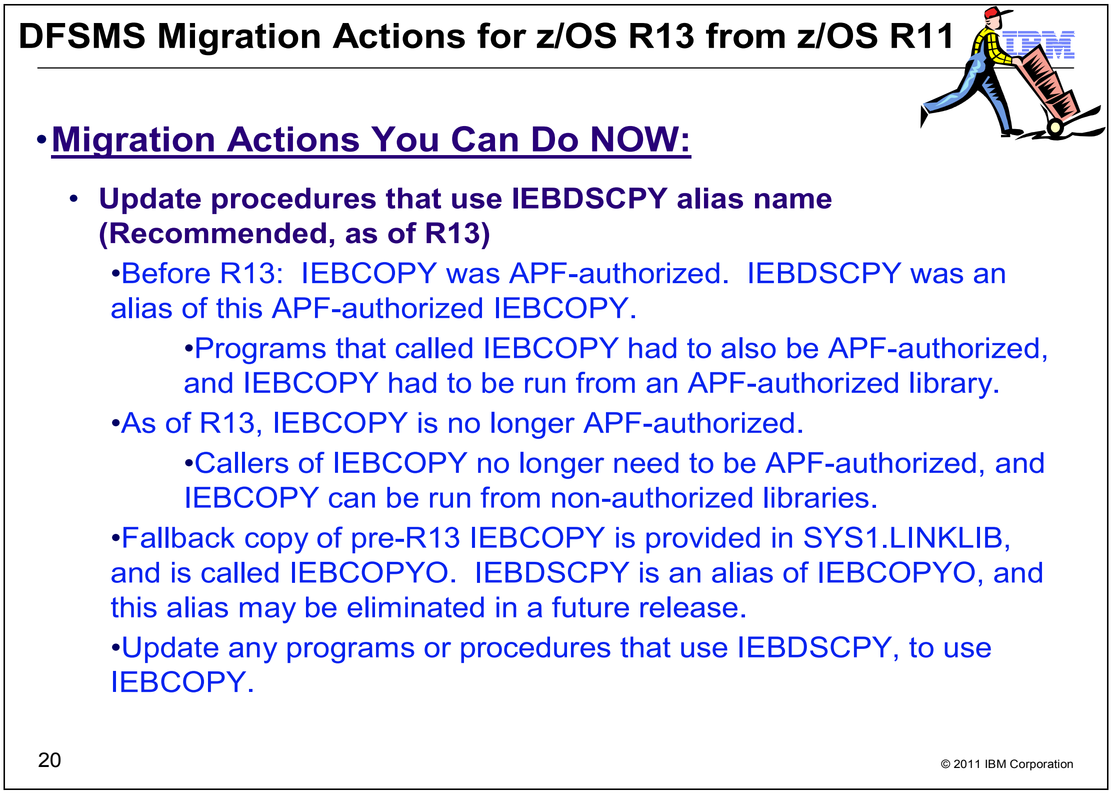
3. If any preference values differ from what you expected, make the necessary configuration changes on the advertising routers.



DFSMS Migration Actions for z/OS R13 from z/OS R11

- **Migration Actions You Can Do NOW:**
  - **Accommodate deletion of NOIMBED and NOREPLICAT LISTCAT command output (Recommended, as of R13)**
    - As of R13, the LISTCAT output will no longer display NOIMBED and NOREPLICAT attributes. This may affect any program parsing LISTCAT output.
    - If a cluster has IMBED or REPLICATE still, LISTCAT will display these attributes.
    - **Action:** programs that parse LISTCAT output should be converted to use the Catalog Search Interface (CSI).
    - **In z/OS R12!** IMBED and REPLICATE attributes will be removed automatically from VSAM data sets logically dumped using DFSMSdss and migrated using DFSMShsm when DFSMSdss is the data mover during restore and recall processing.

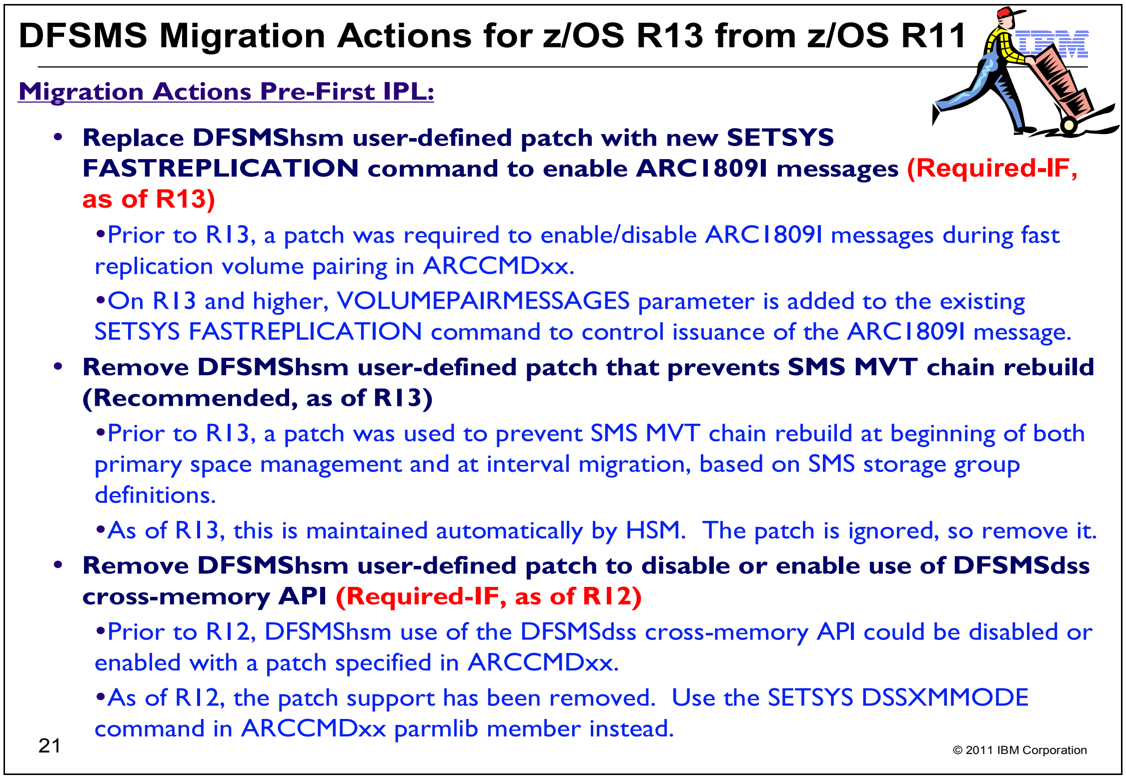
19



DFSMS Migration Actions for z/OS R13 from z/OS R11

- **Migration Actions You Can Do NOW:**
  - **Update procedures that use IEBDSCPY alias name (Recommended, as of R13)**
    - Before R13: IEBCOPY was APF-authorized. IEBDSCPY was an alias of this APF-authorized IEBCOPY.
      - Programs that called IEBCOPY had to also be APF-authorized, and IEBCOPY had to be run from an APF-authorized library.
    - As of R13, IEBCOPY is no longer APF-authorized.
      - Callers of IEBCOPY no longer need to be APF-authorized, and IEBCOPY can be run from non-authorized libraries.
    - Fallback copy of pre-R13 IEBCOPY is provided in SYS1.LINKLIB, and is called IEBCOPYO. IEBDSCPY is an alias of IEBCOPYO, and this alias may be eliminated in a future release.
    - Update any programs or procedures that use IEBDSCPY, to use IEBCOPY.

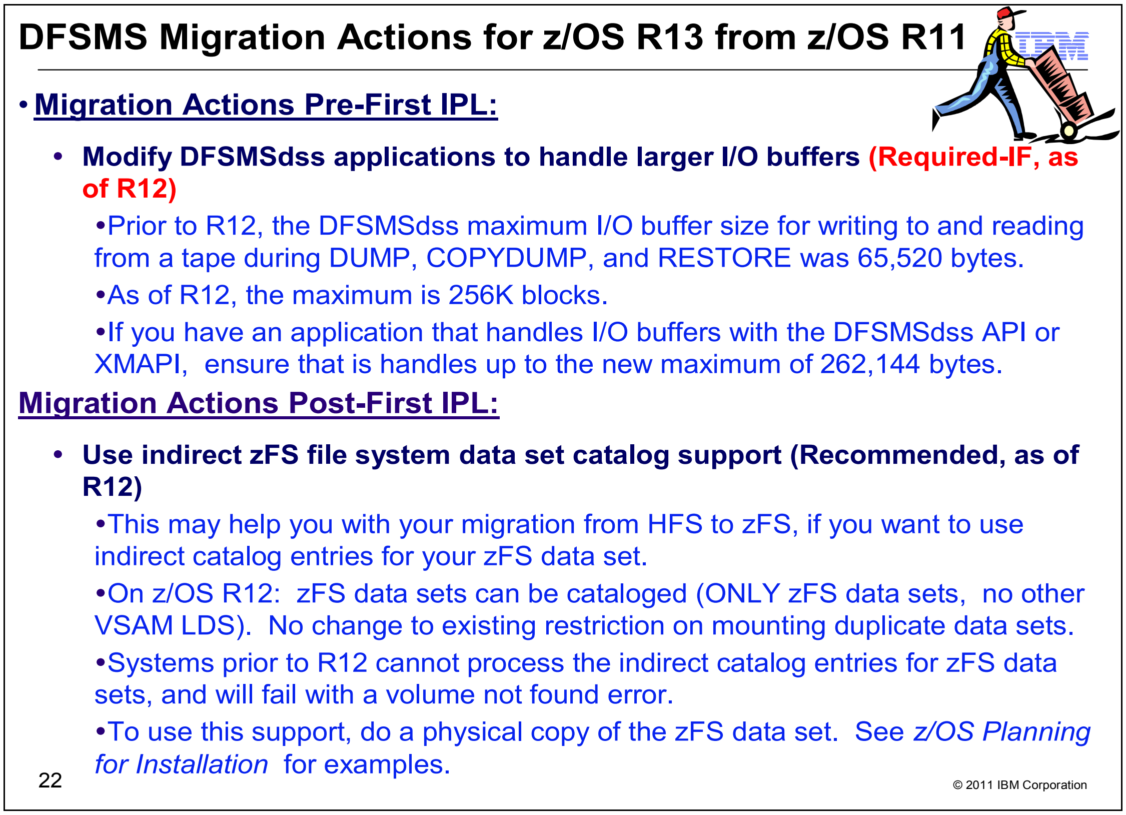
20



DFSMS Migration Actions for z/OS R13 from z/OS R11

Migration Actions Pre-First IPL:

- **Replace DFSMShsm user-defined patch with new SETSYS FASTREPLICATION command to enable ARC1809I messages (Required-IF, as of R13)**
  - •Prior to R13, a patch was required to enable/disable ARC1809I messages during fast replication volume pairing in ARCCMDxx.
  - •On R13 and higher, VOLUMEPAIRMESSAGES parameter is added to the existing SETSYS FASTREPLICATION command to control issuance of the ARC1809I message.
- **Remove DFSMShsm user-defined patch that prevents SMS MVT chain rebuild (Recommended, as of R13)**
  - •Prior to R13, a patch was used to prevent SMS MVT chain rebuild at beginning of both primary space management and at interval migration, based on SMS storage group definitions.
  - •As of R13, this is maintained automatically by HSM. The patch is ignored, so remove it.
- **Remove DFSMShsm user-defined patch to disable or enable use of DFSMSdss cross-memory API (Required-IF, as of R12)**
  - •Prior to R12, DFSMShsm use of the DFSMSdss cross-memory API could be disabled or enabled with a patch specified in ARCCMDxx.
  - •As of R12, the patch support has been removed. Use the SETSYS DSSXMMODE command in ARCCMDxx parmlib member instead.



DFSMS Migration Actions for z/OS R13 from z/OS R11

- **Migration Actions Pre-First IPL:**

- **Modify DFSMSdss applications to handle larger I/O buffers (Required-IF, as of R12)**
  - •Prior to R12, the DFSMSdss maximum I/O buffer size for writing to and reading from a tape during DUMP, COPYDUMP, and RESTORE was 65,520 bytes.
  - •As of R12, the maximum is 256K blocks.
  - •If you have an application that handles I/O buffers with the DFSMSdss API or XMAPI, ensure that is handles up to the new maximum of 262,144 bytes.

Migration Actions Post-First IPL:

- **Use indirect zFS file system data set catalog support (Recommended, as of R12)**
  - •This may help you with your migration from HFS to zFS, if you want to use indirect catalog entries for your zFS data set.
  - •On z/OS R12: zFS data sets can be cataloged (ONLY zFS data sets, no other VSAM LDS). No change to existing restriction on mounting duplicate data sets.
  - •Systems prior to R12 cannot process the indirect catalog entries for zFS data sets, and will fail with a volume not found error.
  - •To use this support, do a physical copy of the zFS data set. See *z/OS Planning for Installation* for examples.

DFSMS Migration Actions Between z/OS V1 R11 and z/OS V1 R13

These migration actions were taken from z/OS *Migration.* Some descriptions and actions have been shortened for inclusion in this presentation. For the complete descriptions and actions, refer to *z/OS Migration.*

DFSMS Migration Actions You Can Do Now

DFSMSdfp: Back up SMS control data sets (Recommended)

Recommended to ensure data integrity.
Migration action: Do the following on your pre-z/OS V1R13 systems:

1. Back up SMS control data sets according to established procedures in the event that fallback is required. The control data set format is VSAM linear.
2. Install all coexistence PTFs defined for z/OS V1R13.

In addition, if you modified and activated a higher-level policy on a pre-z/OS V1R13 system, do the following to ensure that the ACDS can be accessed on z/OS V1R13:

1. On the pre-z/OS V1R13 system, save the active ACDS as an SCDS with the SETSMS SAVESCDS command.
2. On z/OS V1R13, update, translate, validate, and activate the saved SMS policy.

DFSMSdfp: Accommodate deletion of NOIMBED and NOREPLICAT LISTCAT command attributes (Recommended, as of R13)

Not required, but recommended if you use programs that parse LISTCAT results.
Before z/OS V1R13, output from the IDCAMS LISTCAT command displayed the NOIMBED and NOREPLICAT attributes. Starting with z/OS V1R13, these attributes are no longer included in the LISTCAT command output. This might affect programs that parse LISTCAT results. In 2001, support for creating data sets with IMBED or REPLICATE attributes on the AMS DEFINE command was removed. Starting with z/OS V1R13, the LISTCAT output no longer displays the default NOIMBED and NOREPLICAT attributes. This information is no longer needed. This might affect programs that parse LISTCAT results.
Note that for any cluster defined prior to 2001 with IMBED or REPLICATE attributes, those attributes are displayed in IDCAMS LISTCAT command output. Before this z/OS V1R13 change, when you issued a LISTCAT for a data set, at the end, you would see attribute characteristics. They would look something like this example:

```
ATTRIBUTES
KEYLEN-----------------15 AVGLRECL--------------80
RKP--------------------0 MAXLRECL--------------80
STRIPE-COUNT-----------1
SHROPTNS(4,3) RECOVERY UNIQUE NOERASE INDEXED NOWRITECHK NOIMBED NOREPLICAT
UNORDERED NOREUSE NONSPANNED EXTENDED EXT-ADDR
```

Starting with z/OS VR13, when you issue a LISTCAT for a data set, the result will look something like this example:

```
ATTRIBUTES
KEYLEN-----------------15 AVGLRECL--------------80
RKP--------------------0 MAXLRECL--------------80
STRIPE-COUNT-----------1
SHROPTNS(4,3) RECOVERY UNIQUE NOERASE INDEXED NOWRITECHK UNORDERED NOREUSE
NONSPANNED EXTENDED EXT-ADDR
```

Migration action: It is recommended that you convert the programs that parse LISTCAT results to use Catalog Search Interface (CSI).

DFSMSdfp: Modify exit routines to support 31-bit UCB addresses (Required-IF, as of R12)

Required, if an exit routine does not support 31-bit UCB, it will need to be upgraded to support 31-bit UCB addresses.
Before z/OS V1R12, DADSM captured the unit control block (UCB) address passed to it in its interfaces. Starting in z/OS V1R12, DADSM will no longer capture the UCB address passed to it and will use the actual 31-bit UCB address. The UCB address will either be above or below the 16 MB line. This address in turn will be passed into the DADSM pre- and post- installation exit routines (IGGPRE00, IGGPOST0) in the current 4-byte IEXUCB field. Therefore, the address in IEXUCB may now contain a 31-bit UCB address. Exit routines that do not support 31-bit UCB addresses will need to be upgraded to support a 31-bit UCB address.
Migration action: Update the exit to support a 31-bit UCB address.

DFSMSdfp and DFSMSdss: Redefine existing VSAM data sets that contain the IMBED, REPLICATE, and KEYRANGE attributes (Recommended)

Recommended to avoid degraded performance and wasted DASD space.

No supported release of z/OS honors the IMBED, REPLICATE, and KEYRANGE attributes for new VSAM data sets. In fact, using these attributes can waste DASD space and often degrades performance. Servicing these VSAM data sets has become increasingly difficult. In some cases, unplanned outages have occurred. For these reasons, IBM recommends that you stop using IMBED and REPLICATE, and that you minimize or eliminate your use of KEYRANGE.

IMBED and REPLICATE were intended as performance improvements and have been rendered obsolete by newer, cached DASD devices. Striped data sets provide much better performance than KEYRANGE and should be viewed as a candidate for any existing KEYRANGE data sets.

Starting in z/OS V1R12, DFSMSdss provides some assistance in identifying and converting data sets with KEYRANGES, IMBED or REPLICATE attributes:

- Data set dump processing and restore processing issue a message ADR508I (*ttt*)-*mmmmm*(*yy*), THE FOLLOWING DATA SETS REQUIRE SOME ACTION TO BE TAKEN when data sets with those attributes are encountered.
- Logical restore processing automatically converts indexed VSAM data sets with the IMBED or REPLICATE attributes, but not the KEYRANGES attribute, to key-sequenced data sets (KSDS) that do not have those attributes. It also issues message ADR507I (*ttt*)-*mmmmm*(*yy*), DATA SET *dsn* WAS RESTORED WITHOUT THE IMBED OR REPLICATE ATTRIBUTES indicating that the data set has been converted.

Migration action:

1. Determine which VSAM data sets and ICF catalogs were defined with the IMBED, REPLICATE, or KEYRANGE attribute. Data set dump processing and restore processing issue message ADR508I (*ttt*)-*mmmmm*(*yy*), THE FOLLOWING DATA SETS REQUIRE SOME ACTION TO BE TAKEN to indicate data sets with the attributes. To further help you identify the data sets, you can get a tool that reads existing VSAM data sets and ICF catalogs, and reports which ones have these attributes. The tool is available from the software server (ftp.software.ibm.com) in the s390/mvs/tools directory as IMBDSHIP.JCL.TRSD. Download the file in binary format and unterse it on your z/OS system using AMATERSE or TRSMAIN. Instructions for using the tool are included in the downloaded JCL.
   Notes:
   - The tool only checks data sets that are on DASD.
   - “AMATERSE” and “TRSMAIN” are names for a service aid that compresses and decompresses data exchanged with IBM. “AMATERSE” is the preferred program name since its integration into z/OS V1R9. “TRSMAIN” is the original program name and is now shipped as an alias entry point to AMATERSE. For more information about AMATERSE, including several differences with TRSMAIN, see *z/OS MVS Diagnosis: Tools and Service Aids*.
2. Schedule a time for the affected VSAM data sets and ICF catalogs to be unavailable, and redefine them. For VSAM data sets you can use JCL similar to the following:

```
//* EXPORT A KSDS
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//INDD DD DSN=EXAMPLE.KSDS,DISP=OLD
//OUTDD DD DSN=EXAMPLE.KSDS.EXPORTED,DISP=(NEW,CATLG),
// SPACE=(CYL,(1,1)),UNIT=SYSDA
//SYSIN DD *
      EXPORT EXAMPLE.KSDS -
      INFILE(INDD) -
      OUTFILE(OUTDD) -
      TEMPORARY
//* NOW IMPORT THE EXPORTED COPY
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//INDD DD DSN=EXAMPLE.KSDS.EXPORTED,DISP=SHR
//SYSIN DD *
      IMPORT -
      INFILE(INDD) -
      OUTDATASET(EXAMPLE.KSDS)
```

For ICF catalogs, see informational APAR II13354 for step-by-step instructions on using IDCAMS EXPORT/IMPORT with ICF catalogs..

Tip: Use check CATALOG_IMBED_REPLICATE on z/OS V1R11 to detect IMBED and REPLICATE attributes in your master catalog and any connected user catalogs.

The recommendation to migrate from IMBED, REPLICATE, and KEYRANGE was originally made in the z/OS V1R6 timeframe. In Software Announcement 204-180 (RFA39951), dated August 10, 2004, IBM announced its intent to withdraw support for VSAM IMBED, REPLICATE, and KEYRANGE attributes in a future release. Based on customer feedback, IBM no longer plans to remove this support from z/OS in the foreseeable future. IBM still recommends that you stop using these attributes and plans to remove IMBED and REPLICATE attributes during logical DFSMSdss restore operations and DFSMShsm recall operations as announced in Software Announcement 207-175 4 (RFA45594), dated August 7, 2007. The DFSMSdss function to aid in conversion was added in z/OS V1R12.

DFSMSrmm: Replace CIM providers and CIM classes (Required-IF, as of R12)

Required if you have not yet updated your code to handle the current key formats and are still using the backward-compatible CIM provider (provided as rmmcim19.tar.Z compressed tar archive).

In z/OS V1R10, the keys used for the DFSMSrmm CIM classes were changed. Prior to z/OS V1R12, you had the option of using a backward-compatible CIM provider, rather than updating your code to handle the new key formats. Beginning with z/OS V1R12, the backward-compatible CIM provider is no longer supported. If you have not done so already, you must unregister the previous CIM providers and CIM classes, and register the currently supported CIM providers and CIM classes.

Migration action:

1. Update your code to handle the key formats used in z/OS V1R10 and later. For the current formats, see the table in *z/OS R13 Migration.*
2. Using the rmmutil.sh tool, unregister all the z/OS V1R9 CIM providers and unload all the z/OS V1R9 CIM classes.
3. Using the same rmmutil.sh tool, register the complete set of z/OS V1R13 CIM providers and load the z/OS V1R13 CIM classes.

DFSMS Migration Actions Pre-First IPL

DFSMSdfp: Ensure that the Language Environment run-time library is available for DLLs (Required-IF)

Required if your installation builds or references DLLs.

Language Environment provides common services and language-specific routines in a single run-time environment. You can use Language Environment to build and use dynamic link libraries (DLLs) for applications.

Migration action: If your installation builds or references DLLs, either you must set up the system link list to refer to the Language Environment run-time libraries (SCEERUN and SCEERUN2), or each job that creates or uses a DLL must include a STEPLIB DD statement referencing these libraries.

DFSMSdfp: Ensure your SYS1.IMAGELIB is customized for your printing environment (Required-IF)

Required if you are not using your old SYS1.IMAGELIB, you are installing with ServerPac or SystemPac, and you are using line mode printers such as the 3800 or 3900.

If you use line mode printers such as the IBM 3800 or the IBM 3900 running in line mode (not page mode), you must install library character sets, graphic character modification modules, and character arrangement tables in SYS1.IMAGELIB. This migration action does not apply if you are using IBM 3900 printers that are driven by PSF.

Migration action:

1. Run the LCSBLD1 job from the samplib data set to create character sets, graphic character modification modules, and character arrangement tables in SYS1.IMAGELIB.
2. Copy customized or locally-written FCBs and UCS images from your old system's SYS1.IMAGELIB data set to the new system's SYS1.IMAGELIB data set.

DFSMSdfp: Update operator procedures and system automation for new DADSM pre- and post-processing dynamic exits (Required-IF, as of R13)

Required if DADSM installation exits IGGPRE00 or IGGPOST0 are in use.

Before z/OS V1R13, exits for the DADSM pre- and post-processing functions were loaded by DFSMSdfp, as installation exits during initialization, as modules IGGPRE00 and IGGPOST0. Starting with z/OS V1R13, z/OS dynamic exits services is used to define a pre-processing dynamic exit, IGGPRE00_EXIT, and a post-processing dynamic exit, IGGPOST0_EXIT, and associate IGGPRE00 and IGGPOST0 modules as exit routines to these respective dynamic exits. All DADSM functions (create, extend, scratch, partial release, and rename) share these common dynamic exits and will be called where the previous installation exits of IGGPRE00 and IGGPOST0 were called using the same existing interfaces. This change requires changes to DFSMSdfp operating procedures and system automation (if any).

Migration action:

- If you use the IGGPRE00 or the IGGPOST0 installation exits, you do not need to change them in any way; just install them as you always have. DFSMSdfp will automatically exploit the dynamic exit services and use your IGGPRE00 or IGGPOST0 installation exit as exit routines to the new IGGPRE00_EXIT and IGGPOST0_EXIT dynamic exits. You do not need to change the load module names for IGGPRE00 or IGGPOST0, however, you may change the names if desired. If you do change the names, update the PROG*xx* parmlib member or issue the **SETPROG** command to get the modules loaded because DFSMSdfp will not load them as exit routines to the dynamic exits.
- You can now have multiple exit routines associated with each of the IGGPRE00_EXIT and IGGPOST0_EXIT dynamic exits for the DADSM pre- and post-processing exits. Other programs can use the CSVDYNEX macro to associate their exit routines to these dynamic exits and can add and delete exit routines from any dynamic exit routine as required. They also can be added and deleted with the PROG*xx* member of parmlib and with the SETPROG ADD operator command. All exit routines will be called when the DADSM pre- and post-dynamic exits are called from each DADSM function. The execution of one exit routine may then change the behavior of a subsequent one. The order in which the exit routines are called by the system could be in any order.
- The IGGPRE00 and IGGPOST0 module addresses in the CVAF table (CVFDPR31, CVFDPOR31) will continue to be set. Therefore, other programs that continue to use this interface will be unaffected. Since dynamic exit services would not be used in this case, no other exit routine associated with the dynamic exits will be called. These programs should be changed to use dynamic exit services, CSVDYNEX.

DFSMSdfp: Update procedures that use IEBDSCPY alias name to access IEBCOPY

Not required, but recommended if you don't want to be connected to the pre-z/OS V1R13 version of IEBCOPY.

Before z/OS V1R13, the IEBCOPY utility was an APF-authorized program that had to run from an APF-authorized library. If another program called IEBCOPY, that program also had to be APF-authorized. Starting in z/OS V1R13, the IEBCOPY utility is no longer APF-authorized and can be run from non-authorized libraries; callers of IEBCOPY no longer need to be APF-authorized. In addition to this authorization change, other performance improvements have been made to IEBCOPY as well.

IEBCOPY is a data set library management utility that is used to perform many critical system build and maintenance activities. If this utility does not work correctly, then key library data sets could be rendered as unusable and the z/OS user would be left without a fallback method. A fallback copy of the z/OS V1R12 level of IEBCOPY is provided, named IEBCOPYO, for use if you experience problems with the updated version of IEBCOPY in z/OS V1R13. The old IEBCOPYO utility with its alias of IEBDSCPY is installed into SYS1.LINKLIB with authorization code one, AC(1).

Since the code in IEBCOPY is now different from the code in IEBCOPYO, IBM recommends that you do not copy and replace IEBCOPY with IEBCOPYO because of maintenance issues. PTFs will update IEBCOPY and IEBCOPYO load modules appropriately as each load module requires service. The IEBDSCPY alias name of IEBCOPY that exists in earlier versions of DFSMS is no longer an alias name in the z/OS DFSMS V1R13 IEBCOPY load module. The IEBDSCPY alias name continues to exist as the alias to the IEBCOPYO load module in z/OS V1R13, but will be eliminated as an alias of the IEBCOPYO load module in future releases of DFSMS. If you currently access the IEBCOPY utility by using the IEBDSCPY alias name, you need to make the necessary change to use the IEBCOPY primary name.

It is important to note with the changes introduced in z/OS V1R13 IEBCOPY, the requirement for SMP/E to be APF-authorized remains. That is, these IEBCOPY changes have no effect on the requirement that SMP/E is APF-authorized.

Migration action:

- Programs or procedures that use the IEBDSCPY alias name to access IEBCOPY should be changed to use the IEBCOPY primary name. Programs that continue to use the alias name will connect to the old, pre-z/OS V1R13 version, currently called IEBCOPYO.
- IEBCOPY can now be called by programs that are not APF-authorized.

DFSMSdfp: Evaluate applications and modify for EAV enhancements (Required-IF, as of R12)

Required if application programs allocated the non-VSAM data sets listed below with the extended attributes option from z/OS V1R11.

Starting with z/OS V1R12, additional non-VSAM data set types in the extended addressing space (EAS) are supported. This includes support for sequential (BASIC or LARGE), partitioned (PDS or PDSE), Catalogs, and BDAM data sets in EAS. Before z/OS V1R12, any program trying to open one of these types of non-VSAM data sets that have been allocated with extended attribute DSCBs (format 8 DSCB from a z/OS V1R12 system) failed with abend IEC144I 313-0C. Now on a z/OS V1R12 system, applications can open these non-VSAM data sets allocated with extended attribute DSCBs with standard BSAM, BPAM and QSAM access.

In many cases, application programs will function as usual. However, application programs that reference the data set extents found in the DEB or access the DSCB or its extents must change to either support format 8 DSCBs and 28-bit cylinder addressing or to abend when they encounter such data sets.

Migration action:

- Look for applications that allocated the following types of non-VSAM data sets in EAS with the extended attributes option from z/OS V1R11:
  - Sequential (BASIC or LARGE)
  - Partitioned (PDS or PDSE)
  - Catalogs
  - BDAM

  Look for applications that allocate these non-VSAM data sets in EAS with the EATTR=OPT data set keyword on JCL or in SMS data class to specify that the data set can have extended attribute DSCBs (format 8/9 DSCBs) and can optionally have extents in the EAS. In z/OS V1R11, the EATTR keyword would have no affect for these non-VSAM data sets until the data set type is enabled in the system for EAS. However, in z/OS V1R12 these data set types are enabled for EAS.
- Affected applications must either support format 8 DSCBs and 28–bit addressing or to issue an abend code to fail processing when they encounter such data sets, because the system will no longer automatically abend with IEC144I 313-0C.

DFSMSdfp: Accommodate new DCBE macro option (Required-IF, as of R12)

Required if you use the XTIOT, UCB nocapture and DSAB-above-the-line options of dynamic allocation, or if any of the callers of your program might have used those options to allocate files which your program will open .

Before z/OS V1R12, the DCB access methods did not support the nocapture UCB option of a dynamic allocation. Starting in z/OS V1R12, the DCB access methods will support a new DCBE option, LOC=ANY, to signify that the application program supports the XTIOT, UCB nocapture and DSAB-above-the-line options of dynamic allocation. The new DCBE option also signifies that the application program has no dependancy on any of XTIOT, UCB address or DSAB-above-the-line options of dynamic allocation.

Migration action:

1. Review, and modify if needed, the installation exit routines that receive a TIOT entry or UCB address directly or indirectly. These include:
   a. DADSM pre- and post-processing exits (IGGPRE00 and IGGPOST0).
   b. Tape management exits (IFG019LA (label anomaly), IFG019VM (volume mount), IFG019FV (file validation), IFG019FS (file start on volume) and IFG055FE (file end on volume)). They receive UCB addresses from TEPMUCB in IFGTEP, from DEBUCBA and from TIOEFSRT. All three sources will allow a 31-bit UCB address. TEPMUCB sometimes will be an uncaptured 31-bit address. If the new DEB31UCB bit is on, the UCB address and modeset byte will be different as described in IEZDEB.
   c. NSL tape exits (NSLOHDRI, NSLEHDRI, NSLOHDRO, NSLEHDRO, NSLETRLI, NSLETRLO, NSLCTRLO, IEFXVNSL and NSLREPOS). OPEN, EOV and CLOSE will always turn DEB31UCB on in the work area to signify that DXDEBUCB is a 31-bit UCB address and the modeset byte is moved as described above. Even though the UCB address field will be four bytes, it typically will still contain a three-byte address.
   d. Volume label editor routines (IFG0193C and IFG0553C). Same work area changes as described for the NSL routines.
   e. DCB OPEN installation exit (IFG0EX0B).
   f. The IGXMSGEX installation exit for the MSGDISP macro already supports its caller passing a 31-bit UCB address but there might be more cases in which this occurs.

   g. The data management ABEND installation exit (IFG0199I) passes a UCB address in field OAIXUCBA. It might now be 31-bit.
   h. IECIEUCB as mapped by the IECIEPRM macro for the ISO/ANSI Version 3 and Version 4 installation exits (IFG0193G) contains the tape UCB address. In the past this has always been a 24-bit addresss. Now it might be a 31-bit address.

   The above exits are documented in *z/OS DFSMS Installation Exits*. IEFDB401, a dynamic allocation input validation exit, is documented in *z/OS MVS Installation Exits*.
2. Verify that the installation exit routines will not be affected adversely.
3. Set the NON_VSAM_XTIOT option to YES or NO in the DEVSUP*xx* parmlib member.
4. Change the program to set LOC=ANY in the DCBE macro. The default is LOC=BELOW.

Note: The new DCBELOCANY flag can only be used in z/OS V1R12 or later. If a program has the DCBELOCANY flag defined, and is compiled or assembled using a z/OS V1R11 level of the DFSMS macros, then a compile or assemble error will occur. See APAR OA33409 for more information.

DFSMSdss: Build the IPLable stand-alone DFSMSdss image (Required-IF)

Required if you intend to use the Stand-Alone Services provided by DFSMSdss.

If you intend to use the Stand-Alone Services provided by DFSMSdss, you must use the DFSMSdss BUILDSA function to create the Stand-Alone Services IPL-capable core image. Starting with z/OS V1R12, DFSMSdss now uses BSAM instead of EXCP to read from and write to DFSMSdss dump data sets during DUMP, COPYDUMP, and RESTORE operations. To migrate to this support, you must rebuild the IPL-able core image for the Stand-Alone Services program. If this migration action is not performed, users of the DSS standalone restore will not be able to restore backups on tape created with greater than 65520 byte blocks. Message `ADRY3530I SEQUENCE ERROR ON RESTORE TAPE` is issued and the operation is terminated. Backups created with 65520 byte blocks will restore as they did in z/OS V1R11

Migration action:

1. Prepare for Stand-Alone Services by creating a Stand-Alone Services IPL-able core image with the BUILDSA command. With the BUILDSA command you can specify the device (card reader, tape drive, or DASD volume) from which Stand-Alone Services will be IPLed. You can also specify the operator console to be used for Stand-Alone Services. The BUILDSA function builds the IPL-able core image under the current operating system and determines a record size based on whether the IPL is from card, tape, or DASD.
2. Use RACF or another external security system to protect the SYS1.ADR.SAIPLD.Vvolser data set and the Stand-Alone Services modules.
3. If you haven't done so already, make a backup copy of your system that can be restored by this function. For information about backing up volumes, see *z/OS DFSMSdss Storage Administration Guide*.

Note: Message ADRY3530I SEQUENCE ERROR ON RESTORE TAPE might be issued with operation terminated if a user tries to restore a back up that was created with a block size greater than 65520 bytes, using the DSS stand-alone restore program from z/OS V1R10 or z/OS V1R11.

DFSMSdss: Recompile and link-edit exit routines or applications that change options in the ADRUFO block (Required-IF, as of R12)

Required if you change options in the ADRUFO block with either the exit routine or application.

Starting with z/OS V1R12, if you change options in the ADRUFO block with either:

- the options installation exit routine, ADRUIXIT, or
- an application that invokes DFSMSdss with the API or XMAPI and then uses the Presenting ADRUFO Record exit (Eioption exit 13),

you must recompile and link-edit the exit routine or application using macro libraries provided with z/OS V1R13.

Migration action: Recompile and link-edit the exit routine or application using macro libraries provided with release z/OS V1R13.

DFSMSdss: Modify applications to handle larger I/O buffers (Required-IF, as of R12)

Required if the application invokes DFSMSdss with the API or XMAPI.

Starting with z/OS V1R12, DFSMSdss uses BSAM instead of EXCP to read from and write to DFSMSdss dump data sets during DUMP, COPYDUMP, and RESTORE operations. This allows DFSMSdss to support 256K blocks when writing to and reading from a tape. Before z/OS V1R12, the maximum was 65,520 bytes. If you have an application that invokes DFSMSdss with the API or XMAPI, you should ensure that the application handles I/O buffers up to the new maximum of 262,144 bytes. The affected exit options are EIOP03 and EIOP06. For these exits, the storage (buffer) pointed to by EIRECPTR may now be greater than 65,520 bytes, up to a maximum of 262,144 bytes.

Tapes with greater than 65,520 bytes BLKSIZE can be read by z/OS V1R11 systems with the PTFs for APAR OA30822 installed. If you try to restore from a tape that has greater than 64K blocksize, but do not have the PTFs for OA30822 installed on your system, the restore will fail. Most likely, it will fail with an ADR370E INVALID SEQUENCE NUMBER error message.

Migration action: Ensure that the application can handle I/O buffers that are up to 262,144 bytes. If the application cannot handle I/O buffers that are up to 262,144 bytes, you can:

- Specify PARM='USEEXCP=YES' in OPTPTR of the API or the EXEC PARM with PGM=ADRDSSU (for example, EXEC PGM=ADRDSSU,PARM='USEEXCP=YES'). You can also specify PARM='USEEXCP=YES' in OPTPTR of the API that invokes ADRXMAIA, or specify it on the EXEC PARM with PGM=ADRXMAIA (for example, EXEC PGM=ADRXMAIA,PARM='USEEXCP=YES').
- Set a block size to an acceptable value either with the JCL DD statement or when dynamically allocating the DD for your application during processing of the Presenting ADRUFO Record exit (Eioption exit 13).
- Code BLKSZLIM and set the BLKSZLIM value to 65520. Note that DSS only supports BLKSZLIM of 65520 and above; for a backup to be compatible with releases earlier than z/OS V1R10, the backup must have no larger than 65,520 byte blocks. BLKSZLIM can be set in the following places:
  - BLKSZLIM keyword on the DD statement or dynamic allocation. The BLKSZLIM keyword on a DD statement keyword is described in *z/OS MVS JCL Reference*.
  - Block size limit in data class. Set by storage administrator. Available even if the data set is not SMS-managed. The block size in the data class is described in *z/OS DFSMS Implementing System-Managed Storage*.
  - System default set in TAPEBLKSZLIM keyword in DEVSUP*xx* parmlib member in SYS1.PARMLIB. Also available in DFA. The TAPEBLKSZLIM parameter is described in *z/OS MVS Initialization and Tuning Reference*.
- If the application inspects the DTVBLKSZ field in the volume record of physical dumps or the DTHBLKSZ field in the data set header of logical dumps, you may need to make further changes. When a physical backup is created on tape with a block size that is greater than 65,520 bytes, the DTVBLKSZ field is X'FFFE'. The block size is stored in an extended volume record following the volume record, DTSDEVOL. When a logical backup is created on tape with a block size greater than 65,520 bytes, the DTHBLKSZ field is X'FFFE'. The block size is stored in a new extended tape header record, S, following the data set header record.

Note: Tapes with greater than 65,520 bytes BLKSIZE can be read by z/OS V1R11 and z/OS V1R10 systems, with the PTFs for APAR OA30822 installed.

DFSMShsm: Accommodate the changed default of PDA trace during DFSMShsm startup (Required-IF, as of R13)

Required if you do not want the new default.

Before z/OS V1R13, PDA trace was disabled (PDA=NO) during DFSMShsm startup unless PDA=YES was *manually* added to the DFSMShsm startup procedure. Starting with z/OS V1R13, PDA trace is enabled (PDA=YES) by default during DFSMShsm startup. After DFSMShsm is started, the SETSYS PDA setting controls PDA tracing.

Migration action: To enable PDA trace during DFSMShsm startup, no action is required. To disable PDA trace during DFSMShsm startup, specify PDA=NO in the DFSMShsm startup procedure before starting DFSMShsm.

DFSMShsm: Accommodate the changed SETSYS FASTREPLICATION command DATASETRECOVERY parameter default (Required-IF, as of R13)

Required if you want to use fast replication for data set recovery and you do not currently specify a fast replication data set recovery preference in the ARCCMDxx parmlib member of SYS1.PARMLIB

Before z/OS V1R13, the **SETSYS FASTREPLICATION** command DATASETRECOVERY parameter default was PREFERRED and fast replication data set recovery was performed whenever possible. The standard copy method was used when fast replication could not be. Starting with z/OS V1R13, the **SETSYS FASTREPLICATION** command DATASETRECOVERY parameter default is changed to NONE and the standard copy method is used to perform data set
recovery. Fast replication is not used by default.

Migration action:

1. Determine which fast replication data set recovery preference you want to use.
2. Add a fast replication data set recovery preference to the ARCCMD*xx* parmlib member of SYS1.PARMLIB.
   - **a.** If a fast replication data set recovery preference is specified and matches the preference you want to use, no action is required.
   - b. If no fast replication data set recovery preference is specified, add **SETSYS FASTREPLICATION (DATASETRECOVERY (PREFERRED))** to continue using the same preference as before z/OS V1R13.

Alternatively, if your preference is to require fast replication data set recovery, you can specify **SETSYS FASTREPLICATION (DATASETRECOVERY (REQUIRED))**.

Notes:

1. If you previously specified **SETSYS FASTREPLICATION (DATASETRECOVERY (NONE))** in the ARCCMD*xx* parmlib member of SYS1.PARMLIB, you can optionally remove it and use the default.
2. The **FRRECOV** command ALLOWPPRCP options PRESERVEMIRRORREQUIRED, PRESERVEMIRRORPREFERRED, PRESERVEMIRRORNO, and YES cannot be used if fast replication data set recovery is not allowed. That is, if the z/OS V1R13 **SETSYS FASTREPLICATION** command ATASETRECOVERY parameter default is used or if **SETSYS FASTREPLICATION.**

DFSMShsm: Replace user-defined patch with new SETSYS FASTREPLICATION command to enable ARC1809I messages (Required-IF, as of R13)

Required if the ARCCMDxx parmlib member of SYS1.PARMLIB contains: PATCH .FRGCB.+9 BITS(.1......).

Before z/OS V1R13, a patch was required to enable (and subsequently disable) ARC1809I messages during fast replication volume pairing. Starting with z/OS V1R13, the VOLUMEPAIRMESSAGES parameter is added to the existing **SETSYS FASTREPLICATION** command to control the issuance of the ARC1809I messages. Remove the patch and replace it with the new parameter.

Migration action:

1. Remove PATCH .FRGCB.+9 BITS(.1......) from the ARCCMD*xx* parmlib member of SYS1.PARMLIB.
2. Add **SETSYS FASTREPLICATION(VOLUMEPAIRMESSAGES(YES))** to the ARCCMD*xx* parmlib member of SYS1.PARMLIB.

DFSMShsm: Review messages changed from I (informational) to E (eventual action) type (Required-IF, as of R13)

Required if your applications depend on message ARC0036I, ARC0503I, or ARC0704I or if your applications are triggered by message type.

Before z/OS V1R13, messages ARC0036, ARC0503, and ARC0704 were *informational* messages (ending in the letter “I”). Starting with z/OS V1R13, messages ARC0036, ARC0503, and ARC0704 are reclassified as *eventual action* messages (ending in the letter “E”). The meaning of the messages is unchanged. The type is changed only. Each of these messages indicate a significant error has occurred and are more easily identifiable as eventual action messages.

- Message ARC0036E indicates that an I/O error occurred when writing to the PDA output data set.
- Message ARC0503E indicates a dynamic allocation error.
- Message ARC0704E indicates a VTOC copy data set processing error during volume backup, dump, or recovery.

These changes can affect applications that depend on message ARC0036I, ARC0503I, or ARC0704I and applications triggered by message type.

Migration action: Update applications that:

- depend on ARC0036I to work with ARC0036E.
- depend on ARC0503I to work with ARC0503E.
- depend on ARC0704I to work with ARC0704E.
- are triggered by message type.

DFSMShsm: Remove patch that prevents SMS MVT chain rebuild (Recommended, as of R13)

Not required, but recommended because starting from z/OS V1R13, the patch command no longer performs a useful function other than what is already handled automatically by DFSMShsm. Additionally, it been discovered in the past that leaving unnecessary useless patches in DFSMShsm's startup procedures can cause confusion and potential problems.

Before z/OS V1R13, the flag at MCVT+C8 X'80' was used to prevent the SMS MVT chain rebuild at the beginning of primary space management and at the beginning of interval migration, based on the current SMS storage group definitions. If the SMS storage groups had not been changed and no new SMS volumes had been added since the last time primary space management or interval migration was run, you could prevent the SMS MVT chain from being needlessly rebuilt or refreshed by entering the following PATCH command: **PATCH .MCVT.+C8 BITS(1.......)**
Starting with z/OS V1R13, this flag is maintained automatically by DFSMShsm. DFSMShsm checks for the occurrence of an ENF 15 event at the beginning of primary space management, at the beginning of interval migration, and at the beginning of on-demand migration. When an ENF 15 event is reported, DFSMShsm performs an SMS MVT chain rebuild or refresh. After a successful SMS MVT chain rebuild, DFSMShsm does not rebuild or refresh the MVT chain until the next SMS configuration change (that is, the next ENF 15 event).

Starting with z/OS V1R13, DFSMShsm does not take into account the **PATCH .MCVT.+C8 BITS(1.......)** command. Nevertheless, it is recommended to remove this patch command from the ARCCMD*xx* parmlib member of SYS1.PARMLIB to avoid any misunderstanding.
Migration action: Remove **PATCH .MCVT.+C8 BITS(1.......)** from the ARCCMD*xx* parmlib member of SYS1.PARMLIB. Remove this patch when all systems are at z/OS V1R13 if a common DFSMShsm's startup procedure is shared with downlevel systems.

DFSMShsm: Update operator procedure in the Multicluster CDS environment (Required-IF, as of R13)

Required if you use the DFSMShsm Multicluster CDS environment.
In the DFSMShsm Multicluster control data sets environment, the same number of clusters must be specified to all DFSMShsm hosts that are sharing a multicluster MCDS or BCDS, or both. If a DFSMShsm host is started with a different number of clusters than the other currently active hosts, the CDS will be logically corrupted and may require extensive recovery actions.
In z/OS V1R12 (and in z/OS V1R11, z/OS V1R10, and z/OS V1R9 when the PTF for OA29346 is installed), DFSMShsm is modified to detect a conflict between the number of CDS clusters specified to a host in the startup proclib JCL and the number recorded in the MHCR record.
Migration action: A new WTOR message, ARC0264A, is issued asking if the change to the cluster number is intended. If the reply is Y, the DFSMShsm host is allowed to start with the new cluster count. If the reply is N, the DFSMShsm host is not allowed to start, which results in error message ARC0130I RC20.

```
ARC0264A {MCDS|BCDS} CLUSTERS CHANGED FROM m TO d. IF NOT INTENDED,
STARTUP WILL RESULT IN CDS CORRUPTION. INTENDED? (Y OR N)
ARC0130I CONTROL DATA SET DEFINITION RULES FOR THE {MCDS|BCDS} WERE NOT FOLLOWED,
RETURN CODE=20
```

The WTOR message is issued at the first DFSMShsm host initialization right after switching into the multicluster CDS environment.
Note: Regardless of the single or multiple DFSMShsm host environment, the initialization of the first host after the number of CDS clusters has been updated by the reorganization will result in an ARC0264A WTOR message asking the operator to confirm the new cluster count.

DFSMShsm: Remove user-defined patch that disables or enables use of the DFSMSdss cross memory API (Required-IF, as of R12)

Required if the ARCCMDxx parmlib member of SYS1.PARMLIB contains: PATCH .MCVT.+433.
In z/OS V1R10 and later, DFSMShsm's use of the DFSMSdss cross memory API can be disabled or enabled by including a patch or the **SETSYS DSSXMMODE (Y | N)** command in the ARCCMD*xx* parmlib member of SYS1.PARMLIB. Starting with z/OS V1R12 , support is added to the existing **SETSYS DSSXMMODE** command to permit the **Y** and **N** setting to individually control backup, CDS backup, dump, migration, full-volume recovery, and data set recovery. Because of this, disabling or enabling DFSMSdss cross memory mode should be done exclusively with the **SETSYS DSSXMMODE** command. Instances of patching the MCVT at offset +433 should be removed from ARCCMD*xx.*
Migration action: Remove PATCH .MCVT.+433 from the ARCCMD*xx* parmlib member of SYS1.PARMLIB. Add the corresponding SETSYS DSSXMMODE command in the ARCCMD*xx* parmlib member of SYS1.PARMLIB.

DFSMShsm: Configure your security system to permit started procedures using new address space identifier (Required-IF, as of R12)

Required if DFSMSdss cross memory support is used for full-volume recovery from dump.
Before z/OS V1R12, when using the DFSMSdss cross memory API, the address space identifier for full-volume and data set recovery from dump was ARC*n*REST. Starting in z/OS V1R12, when using the DFSMSdss cross memory API, the address space identifier for full-volume recovery from dump is changed to ARC*n*RST*y* where *n* is the DFSMShsm host ID and *y* is the instance of the DFSMSdss started task (a number 1 - 4). Data set recovery from dump will still use ARC*n*REST.
Migration action: Before attempting full-volume recovery from dump, configure your security system to permit started procedures using the new address space identifier for DFSMSdss cross memory support for full-volume recovery from dump.

DFSMShsm: Update applications that depend on QUERY COPYPOOL output (Req-IF, as of R12)

Required if you will be using any FlashCopy technique that creates a full-volume FlashCopy relationship or if you will be specifying SETSYS FASTREPLICATION(FCRELATION(FULL)) and your applications depend on QUERY COPYPOOL output or message ARC1820I.

Before z/OS V1R12, the DFSMShsm QUERY COPYPOOL command output did not display FlashCopy "background copy percent-complete" information. Starting in z/OS V1R12, the QUERY COPYPOOL command output, in message ARC1820I, will display applicable "background copy percent-complete" (PCT-COMP) information for full-volume FlashCopy pairs with an incomplete background copy. Percent-complete information (a percentage) is available for full-volume FlashCopy pairs with an incomplete background copy only. A full-volume FlashCopy relationship is established when the FlashCopy technique (such as fast reverse restore or incremental) designates it, or when SETSYS FASTREPLICATION(FCRELATION(FULL)) has been specified. This change can affect applications that depend on QUERY COPYPOOL output.
Migration action: Update applications that depend on QUERY COPYPOOL output.

DFSMShsm: Update applications that depend on LIST command output (Required-IF, as of R12)

Required if your applications depend on the RACF IND field value in the output with the OUTDATASET or SYSOUT destination, or if your applications depend on LIST COPYPOOL(cpname) output.
The DFSMShsm LIST command output changed in z/OS V1R11 and in z/OS V1R12.

- Starting in z/OS V1R11, the LIST DSNAME(*dsname*) BCDS and LIST LEVEL(*hlq*) BCDS output no longer display the RACF IND field when OUTDATASET or SYSOUT (default) is specified as the destination for the output. The RACF IND field is displayed when TERMINAL is specified as the destination for the output.
- Starting in z/OS V1R12, the LIST COPYPOOL(*cpname*) output includes: a new FASTREPLICATION state (FCFRRINCOMPLETE), fast reverse restore status field (FCFRR=), and recovery complete status field (RECOVERYINCOMPLETE=). This new output is displayed when OUTDATASET, SYSOUT, or TERMINAL is specified as the destination for the output.

These changes can affect applications that depend on LIST output.
Migration action:

1. Remove any dependency on the RACF IND field on the LIST DSNAME(*dsname*) BCDS or LIST LEVEL(*hlq*) BCDS output when using OUTDATASET or SYSOUT.
2. Update applications that depend on LIST COPYPOOL(*cpname*) output to handle the new FASTREPLICATION state and new fields.

DFSMS Migration Actions Post-First IPL

DFSMSdfp: Accommodate 64-bit and AR mode rules enforcement in DFSMS macros (Required-IF, as of R13)

Required if you can code, including exits, that invokes DFSMS macros.
Before z/OS V1R13, many DFSMS macros that did not support 64-bit or AR mode did not react to being invoked in 64-bit or AR mode, and generated code that might have been invalid in 64-bit or AR mode. Starting with z/OS V1R13, these macros are changed to issue an assembly-time message and suppress expansion if they are invoked in 64-bit or AR mode. If you have code, including exits, that invokes DFSMS macros, you should review your code and modify it as appropriate to accommodate the new enforcement of 64-bit and AR mode rules.
Migration action: Modify your source code so that no DFSMS macros are invoked after the SYSSTATE macro specifies 64-bit or AR mode. (The one exception is the TRKADDR macro, which supports 64-bit mode.) When you assemble your code, if a return code of 8 or greater is returned by the High Level Assembler, check for assembler messages that indicate that a macro has been invoked after AMODE 64 or AR mode was specified on the SYSSTATE macro.
Change the source code as appropriate:

- If the macro invocation will be executing in a supported environment (31-bit or 24-bit and not AR mode), then precede that invocation with SYSSTATE AMODE64=NO,ASCENV=P.
- If your tests show that the macro expansion does work when invoked in 64-bit or AR mode, then you can consider coding SYSSTATE with AMODE64=NO and ASCENV=P even though it does not match the execution environment. This type of macro invocation is not supported by IBM unless the documentation for that macro says otherwise.

DFSMSdfp: Run OAM configuration database migration job (Required-IF, as of R13)

Required if you use OAM.
When migrating to z/OS V1R13, you must run the OAM configuration database migration job (CBRSMR1D). CBRSMR1D creates the File System Delete Table in the OAM Configuration Database. You must run CBRSMR1D even if you do not plan to use OAM file system support.
Migration action:

1. Update and run the OAM configuration database migration job (CBRSMR1D) provided in SAMPLIB.

2. Run OAM DB2 BIND and GRANT jobs. To determine which BIND and GRANT jobs you need to run, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*.

DFSMSdfp: Run OAM DB2 BIND jobs (Required-IF)

Required if you use OAM object support.

When migrating to any new release of z/OS, you must run OAM DB2 BIND jobs if you are using OAM for object support. The BIND jobs update DB2 with new OAM DB2 code.

Migration action: Run the BIND jobs appropriate to your installation:

1. Update and execute the samplib job CBRPBIND (OAM DB2 Bind Package Job).
2. 2. Do one of the following:
   a. If your installation starts OAM, uses the file system sublevel or optical or tape devices, or uses the OAM storage management component (OSMC), do the following:
      i. Update and execute samplib job CBRABIND (OAM DB2 Application Plan Bind for LCS and OSR).
      ii. Update and execute samplib job CBRHBIND (OAM DB2 Application Plan Bind for OSMC).
   b. If your installation does not start OAM, use the file system sublevel or optical or tape devices, or use OSMC, update and execute samplib job CBRIBIND (OAM DB2 Application Plan Bind for OSR only).
3. For more information, see the topic “Migrating, Installing, and Customizing OAM” in z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support.

Note: If you choose to edit a previous version of an OAM BIND job, you must incorporate any new changes as described in the header of each samplib OAM BIND job.

DFSMSdfp: Use indirect zFS file system data set catalog support (Recommended, as of R12)

Not required, but recommended to make deployment easier for zFS file system data sets.

Starting in z/OS V1R12, zFS file systems may be cataloged using a system symbol. This allows zFS file system data sets to be indirectly cataloged the same way as non-VSAM data sets.

Systems prior to z/OS V1R12 cannot process the indirectly-cataloged zFS file system data sets and will fail with volume not found errors. This support is limited to zFS file system data sets only. That is, all VSAM linear data sets are not included in this support; only data sets formatted as zFS file systems are included in this support.

Migration action: Use the reference information below to use this new support.

Reference information:

- For setting up the indirect catalog entry, see "Define Cluster" in *z/OS DFSMS Access Method Services for Catalogs*.
- For information about the steps required to establish an indirect catalog entry for zFS file system data sets and what the IDCAMS LISTCAT output will produce, see *z/OS DFSMS Access Method Services for Catalogs*.
- For cloning processes to use this support, see *z/OS Planning for Installation*.

DFSMSdss: Accommodate Catalog Search Interface default change (Required-IF, as of R13, and back to R11 with OA32120)

Required if you want to use CSI during Catalog filtering for logical data set processing.

In z/OS V1R11, DFSMSdss logical data set COPY, DUMP, and RELEASE operation used the Catalog Search Interface (CSI) by default to find cataloged data sets based on the generic filter criteria on the INCLUDE keyword when no input volumes are specified. Prior to z/OS V1R11, you could make use of CSI functionality on z/OS V1R10, z/OS V1R9, and z/OS V1R8 systems by installing the PTF for APAR OA25644 and patching the offset X'54' into the ADRPATCH module to X'11'.

In z/OS V1R13 (and with the PTF for APAR OA32120 installed on z/OS V1R12 and z/OS V1R11), DFSMSdss no longer uses the CSI for Catalog filtering during logical data set processing as the default; DFSMSdss uses generic catalog locates in this scenario.

Migration action:

- To use the CSI during Catalog filtering for the DFSMSdss logical data set COPY, DUMP, and RELEASE operation, the DFSMSdss patch byte at offset X'54' must be set to X'11' to enable the functionality.
- If the DFSMSdss patch byte at offset X'54' is set to any value other than X'00' and X'11' to use generic catalog locates instead of CSI (as done in earlier releases), you do not need to set it because this previous method of finding cataloged data sets is now in effect by default.

Note: If you are intentionally using the CSI default by setting the DFSMSdss PATCH byte at offset X'54' to X'11', then you don't need to take any action to expect the functionality be effective. However, if you left the DFSMSdss PATCH byte at offset X'54' as X'00' and want to continue using the CSI during Logical Data Set processing, you need to set that Patch byte to X'11'.

DFSMShsm: Stop using the HOLD command to quiesce activity prior to control data set backup (Recommended, as of R13)

Not required, but recommended because DFSMShsm will automatically release the ARCCAT resource when a CDS backup is starts.

Before z/OS V1R13, you might have manually or programmatically held DFSMShsm activity using the **HOLD** command prior to starting a control data set (CDS) backup. Starting with z/OS V1R13, the ARCCAT resource is released by all functions running on z/OS V1R13 DFSMShsm hosts, and the functions are quiesced when CDS backup starts. Manually or programmatically holding DFSMShsm activity is no longer necessary.

Cross coupling facility (XCF) services are required to communicate the start of a CDS backup to all DFSMShsm hosts. XCF services must be available and configured properly.

The following are restrictions of taking this migration action.

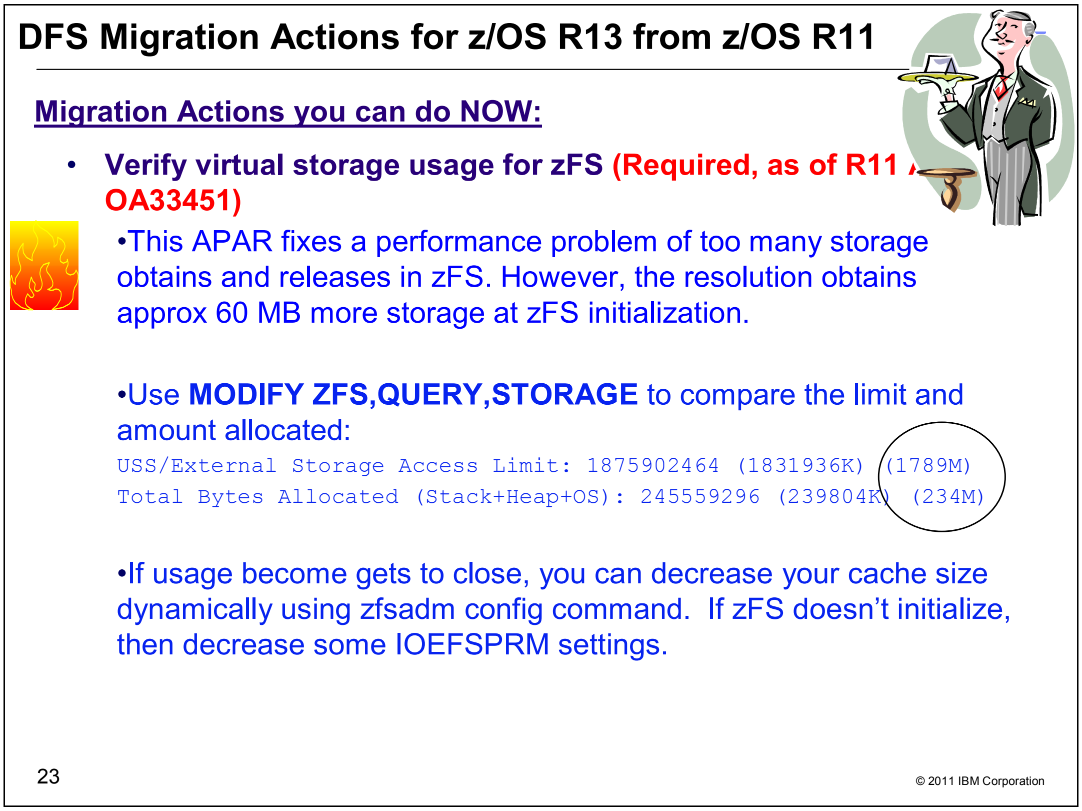
- In a record-level sharing (RLS) CDS environment, all DFSMShsm hosts in the HSMPlex must be z/OS V1R13 or later hosts.
- In a non-RLS CDS environment, this migration action can be taken on z/OS V1R13 DFSMShsm hosts without changing hosts running on prior releases of z/OS.
- Some DFSMShsm environment configuration do not require XCF services. Specifically, a non-RLS CDS non-multiple address space DFSMShsm (MASH) configuration typically does not require XCF services. However, XCF services are required and must be available and configured in all DFSMShsm RLS CDS and MASH configurations.

If you continue to issue a HOLD command to quiesce DFSMShsm activity before a CDS backup and a corresponding RELEASE command to resume activity after CDS backup is complete, the only impact is that you will not see the performance benefit intended by this enhancement.

Migration action: On all z/OS V1R13 DFSMShsm hosts:

1. Remove the procedures, processes, or programs that issue the **HOLD** command to quiesce DFSMShsm activity prior to starting CDS backup.

Remove the corresponding procedures, processes, or programs that issue the **RELEASE** command to resume DFSMShsm activity after CDS backup completes.



DFS Migration Actions for z/OS R13 from z/OS R11

Migration Actions you can do NOW:

- **Verify virtual storage usage for zFS (Required, as of R11 A OA33451)**

•This APAR fixes a performance problem of too many storage obtains and releases in zFS. However, the resolution obtains approx 60 MB more storage at zFS initialization.

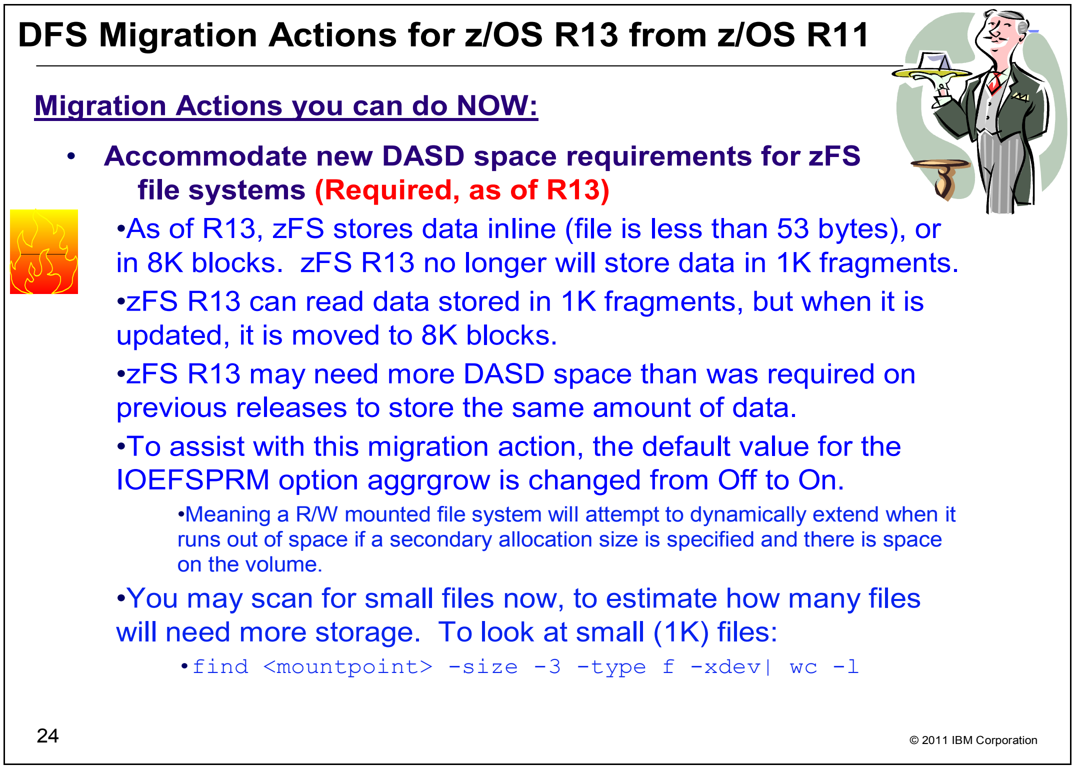
•Use **MODIFY ZFS,QUERY,STORAGE** to compare the limit and amount allocated:

```
USS/External Storage Access Limit: 1875902464 (1831936K) (1789M)
Total Bytes Allocated (Stack+Heap+OS): 245559296 (239804K) (234M)
```

•If usage become gets to close, you can decrease your cache size dynamically using zfsadm config command. If zFS doesn't initialize, then decrease some IOEFSPRM settings.

23

© 2011 IBM Corporation



DFS Migration Actions for z/OS R13 from z/OS R11

Migration Actions you can do NOW:

- **Accommodate new DASD space requirements for zFS file systems (Required, as of R13)**

•As of R13, zFS stores data inline (file is less than 53 bytes), or in 8K blocks. zFS R13 no longer will store data in 1K fragments.

•zFS R13 can read data stored in 1K fragments, but when it is updated, it is moved to 8K blocks.

•zFS R13 may need more DASD space than was required on previous releases to store the same amount of data.

•To assist with this migration action, the default value for the IOEFSPRM option aggrgrow is changed from Off to On.

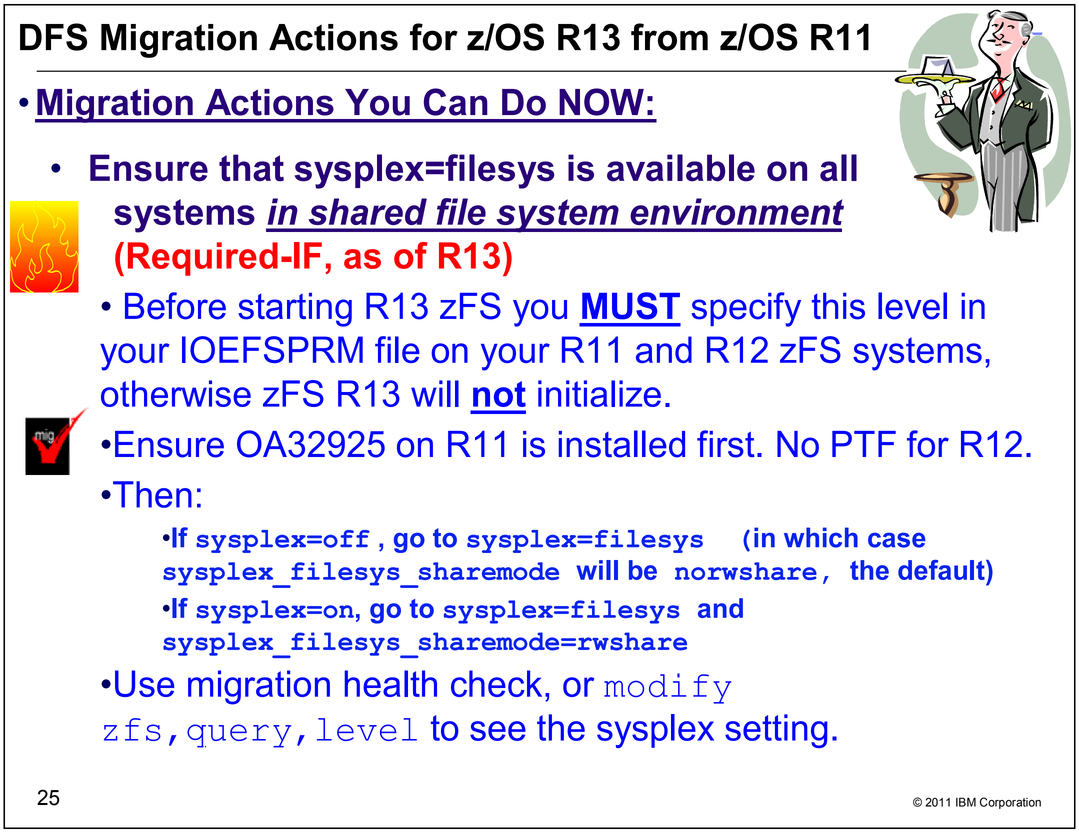
•Meaning a R/W mounted file system will attempt to dynamically extend when it runs out of space if a secondary allocation size is specified and there is space on the volume.

•You may scan for small files now, to estimate how many files will need more storage. To look at small (1K) files:

```
•find <mountpoint> -size -3 -type f -xdev| wc -l
```

24

© 2011 IBM Corporation

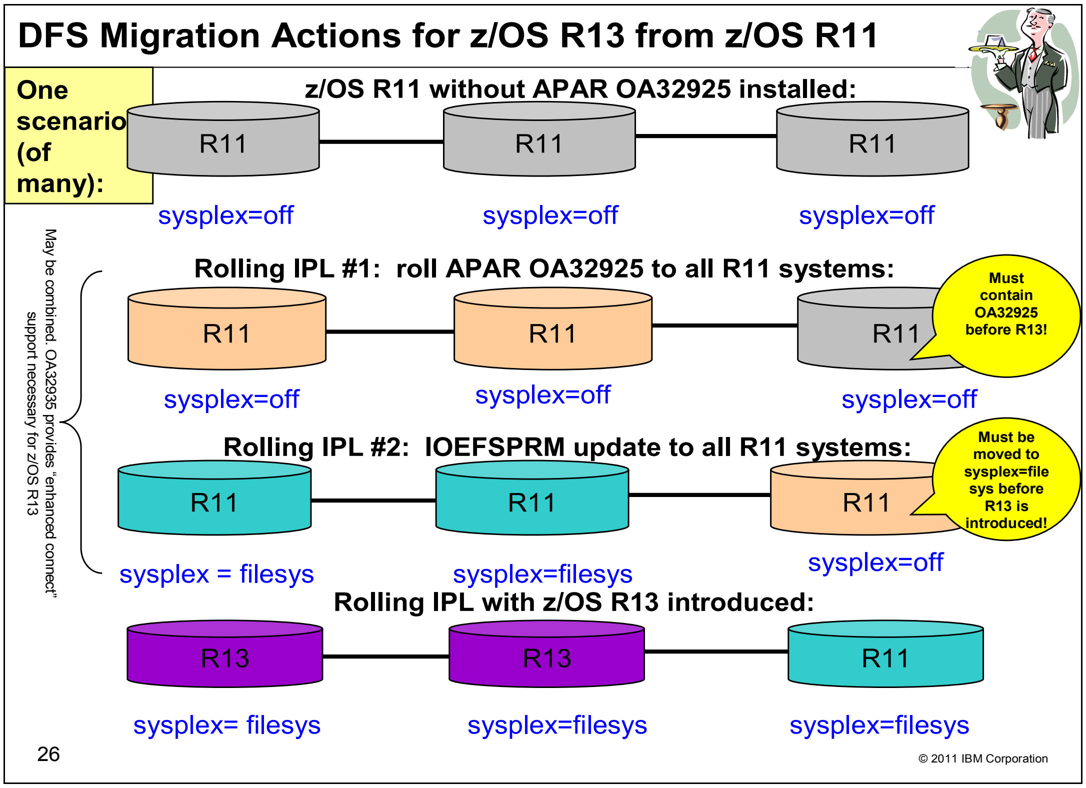


DFS Migration Actions for z/OS R13 from z/OS R11

- **Migration Actions You Can Do NOW:**
  - **Ensure that sysplex=filesys is available on all systems *in shared file system environment* (Required-IF, as of R13)**
    - Before starting R13 zFS you **MUST** specify this level in your IOEFSPRM file on your R11 and R12 zFS systems, otherwise zFS R13 will **not** initialize.
    - Ensure OA32925 on R11 is installed first. No PTF for R12.
    - Then:
      - **If `sysplex=off` , go to `sysplex=filesys` (in which case `sysplex_filesys_sharemode` will be `norwshare`, the default)**
      - **If `sysplex=on`, go to `sysplex=filesys` and `sysplex_filesys_sharemode=rwshare`**
    - Use migration health check, or `modify zfs,query,level` to see the sysplex setting.

25

© 2011 IBM Corporation



DFS Migration Actions for z/OS R13 from z/OS R11

One scenario (of many):

z/OS R11 without APAR OA32925 installed:

R11 — R11 — R11

sysplex=off sysplex=off sysplex=off

May be combined. OA32935 provides "enhanced connect" support necessary for z/OS R13

Rolling IPL #1: roll APAR OA32925 to all R11 systems:

R11 — R11 — R11

sysplex=off sysplex=off sysplex=off

Must contain OA32925 before R13!

Rolling IPL #2: IOEFSPRM update to all R11 systems:

R11 — R11 — R11

sysplex = filesys sysplex=filesys sysplex=off

Must be moved to sysplex=file sys before R13 is introduced!

Rolling IPL with z/OS R13 introduced:

R13 — R13 — R11

sysplex= filesys sysplex=filesys sysplex=filesys

26

© 2011 IBM Corporation

DFS Migration Actions Between z/OS V1 R11 and z/OS V1 R13

These migration actions were taken from *z/OS Migration*. Some descriptions and actions have been shortened for inclusion in this presentation. For the complete descriptions and actions, refer to *z/OS Migration.*

DFS Migration Actions You Can Do Now

zFS: Accommodate new DASD space requirements (Required-IF, as of R13)

Required if you will be using zFS V1R13 to create new zFS file systems or update data in existing file systems, where the file system contains many small files. This action is also required if you have not specified the zFS aggrgrow option in your IOEFSPRM configuration options file.

zFS always reads and writes data in 8K blocks. However, in z/OS V1R13, zFS stores data either inline or in 8K blocks. (Inline data is a file that is smaller than 53 bytes and is stored in the file's metadata.) Unlike in previous releases, zFS R13 no longer stores data in 1K fragments. zFS R13 can read data stored in fragments; however, when the data is updated, it is moved into 8K blocks. Previously, zFS could store data in 1K fragments (contained in an 8K block). This meant that multiple small files could be stored in a single 8K block. Because data is no longer stored in fragments, zFS R13 might need more DASD storage than was required in previous releases to store the same amount of data. More storage may also be needed if zFS R13 is in a mixed-release sysplex and becomes the zFS owning system of a file system.

- **Scenario 1:** If every file in the file system is 1K or less, zFS R13 could require up to four times the DASD storage as was needed in previous releases.
- **Scenario 2:** Because HFS uses 4K blocks to store data and zFS uses 8K blocks, if every file in the file system were 4K or less, zFS R13 could require up to twice as much DASD space to store these files.
- **Scenario 3:** If the file system contains 1000 files that are 1K in size, zFS in R13 could take a maximum of 10 cylinders more than zFS in previous releases.

Typically, however, any increase in the DASD storage used by zFS R13 will be negligible. For example, the z/OS V1R13 version root file system copied using zFS R13 takes approximately 2% more space than the same file system copied using zFS R11. Note that zFS R13 packs multiple ACLs and symbolic links into an 8K block which previous releases did not do. To minimize the chance of application failure due to running out of DASD storage in newly mounted file systems, the default value for the IOEFSPRM option aggrgrow is changed from **Off** to **On**.

Migration action: Perform the following steps, as appropriate for your installation.

For all zFS file systems:

1. If you have not specified the zFS aggrgrow option in your IOEFSPRM configuration options file, recognize that the default is changing in z/OS V1R13 from aggrgrow=off to aggrgrow=on. This means that by default, a zFS read-write mounted file system that is mounted on z/OS V1R13 will attempt to dynamically extend when it runs out of space if a secondary allocation size is specified and there is space on the volume(s).
2. If you do not want that default change and you want it to act as in prior releases, specify aggrgrow=off in your IOEFSPRM configuration options file so that it takes effect on the next IPL. You can dynamically change the aggrgrow option to off with the **zfsadm config -aggrgrow off** command. You can see your current value for aggrgow with the **zfsadm configquery –aggrgrow** command.

For new zFS file systems:

1. Increase the estimated size of a new zFS file system, if you know that many files in the file system will be small.
2. Mount zFS read-write file systems and allow them to dynamically extend; if more DASD space is needed, applications will not fail because the file systems are out of storage. To do so, mount the file systems with the AGGRGROW mount option or use the default aggrgrow=on IOEFSPRM configuration option. The data set must have a non-zero secondary allocation size and there must be space on the volume to allow dynamic extension.

For existing zFS file systems:

1. Use the scan for small files utility (zfsspace) to determine if an existing file system needs more DASD storage. For a mounted zFS file system, the utility shows the number of small files (1K or less), if a secondary allocation is specified, and if aggrgrow=on is specified.You can determine how many files you have in a file system that are less than or equal to 1K in size by using the following shell command: **find <mountpoint> -size -3 -type f -xdev | wc –l** The zfsspace utility can be downloaded from ftp://public.dhe.ibm.com/s390/zos/tools/zfsspace/zfsspace.txt.
2. If a file system has a secondary allocation size and is mounted with the AGGRGROW mount option, allow it to dynamically extend to minimize the potential failure due to lack of storage. If there are insufficient candidate volumes, also consider adding volumes by using the IDCAMS ALTER command with the

ADDVOLUMES option. Generally, after adding volumes, a remount samemode is required to have them take effect.

3. If a file system is not enabled to dynamically extend, consider explicitly growing the file system using the z/OS UNIX **zfsadm grow** command. This is especially important if the file system contains many small files that will be updated.
4. If you expect a file system to grow larger than 4GB (about 5825 3390 cylinders) and it is not SMS-managed with extended addressability, you will need to copy it to an SMS-managed zFS data set with a data class that includes extended addressability. To do so, use the **pax** command. If a zFS aggregate is to be larger than 4GB, it must be SMS-managed with a data class that includes extended addressability.

zFS: Copy cloned file systems to a compatibility mode aggregate (Required-IF, as of R13)

Required if your installation uses cloned file systems.

z/OS V1R13 is planned to be the last release that zFS will support cloning file systems. In anticipation of this removal of support, you should discontinue using zFS clone functions, such as the **zfsadm clone** and **zfsadm clonesys** commands. You should also discontinue mounting any zFS file system aggregates that contain a cloned (.bak) file system. When support for cloning file systems is withdrawn, only zFS compatibility mode aggregates will be supported.

Migration action:

1. Determine if cloned file systems (.bak) have been created or are in the process of being created on your system.
   - Issue the **modify zfs,query** command and review the contents of the FILE report. The Flg field in the report will indicate the status of the file system aggregate.
2. If your system contains cloned file systems, copy that data to a compatibility mode aggregate.

zFS: Copy data from zFS multi-file system aggregates to zFS compatibility mode aggregates (Required-IF, as of R13)

Required if your installation uses multi-file system aggregates.

z/OS V1R13 is planned to be the last release of zFS support for multi-file system aggregates. If you have data stored in zFS multi-file system aggregates, you should copy the data from the zFS multi-file system aggregates into zFS compatibility mode aggregates. When this support is withdrawn, only zFS compatibility mode aggregates will be supported.

Migration action: Use one of the following methods to determine if you are using zFS multi-file system aggregates:

- Use the IBM Health Checker for z/OS check, ZOSMIGV1R11_ZFS_RM_MULTIFS or check ZOSMIGREC_ZFS_RM_MULTIFS.
- Scan your zFS IOEFSPRM configuration options file for define_aggr statements.
- Scan your /etc/rc file for any **zfsadm attach** commands.
- Issue the **zfsadm aggrinfo** command to determine if an aggregate is a multi-file system aggregate; in the command response, COMP indicates compatibility mode and MULT indicates multi-file system.

If you are using zFS multi-file system aggregates, copy the data from each of those file systems into its own zFS compatibility mode aggregate.

zFS: Ensure that sysplex=filesys is available on all systems in a shared file system environment (Required-IF, as of R13)

Required if you have a shared file system environment with more than one system in that environment.

In z/OS V1R13, zFS only runs in sysplex=filesys mode. This requires that all sysplex members in the shared file system environment must run sysplex=filesys, including any z/OS V1R11 and z/OS V1R12 systems. Specifying zFS sysplex=filesys in a shared file system environment causes zFS to run sysplex-aware on a file system basis. This is the preferred mode for zFS in a shared file system environment.

Install the PTF for APAR OA32925 on z/OS V1R11. If a problem occurs when zFS is running sysplex=filesys on a z/OS V1R11 or z/OS V1R12 system, you can perform the following steps:

- Remove the sysplex specification from each system or specify sysplex=off on each system (this is equivalent to the default).

- Perform a rolling IPL or restart zFS on each system.

This procedure cannot be done after zFS on the z/OS V1R13 system has joined the sysplex. If you try to start zFS on another z/OS V1R13 system after you have changed zFS to sysplex=off on the z/OS V1R11 or z/OS V1R12 system, zFS on z/OS V1R13 will not start. This happens because zFS on z/OS V1R13 requires all other systems be zFS sysplex=filesys.

Also, if you try to bring in zFS z/OS V1R13 when sysplex=filesys is not active on all systems, you will receive message IOEZ00721I Sysplex member *sysname* is not running sysplex=filesys. zFS on this initializing member will terminate., where *sysname* is the sysplex member that is not running sysplex=filesys.

Migration action: Perform the following steps to ensure that sysplex=filesys is available on all zFS z/OS V1R11 and z/OS V1R12 systems in a shared file system environment.

1. Install the PTF for APAR OA32925 (UA55765) on all z/OS V1R11 systems, and make it active on all systems through a rolling IPL. This provides the enhanced connect function required by zFS V1R13.
2. If you are currently running zFS sysplex=off, specify sysplex=filesys and make it active on all systems through a rolling IPL. If you are running sysplex=on, specify sysplex=filesys and sysplex_filesys_sharemode=rwshare and make it active on all systems through a rolling IPL. The health check ZOSMIGV1R13_ZFS_FILESYS verifies that all z/OS V1R11 and z/OS V1R12 systems in the shared file system environment have specified sysplex=filesys before z/OS V1R13 is introduced.

To determine if you are running zFS sysplex=filesys, issue the MODIFY ZFS,QUERY,LEVEL operator command. In a shared file system environment, the last line of the response indicates if zFS is running sysplex=filesys. In the following example, zFS is running sysplex=filesys.

```
f zfs,query,level
IOEZ00639I zFS kernel: z/OS zSeries File System
Version 01.11.00 Service Level OA33895 - HZFS3B0.
Created on Mon Aug 23 14:02:18 EDT 2010.
sysplex(filesys,norwshare) interface(3)
IOEZ00025I zFS kernel: MODIFY command - QUERY,LEVEL completed successfully
```

If you do not perform these steps on z/OS V1R11 or z/OS V1R2 systems, you will receive error messages when you try to bring up zFS on a z/OS V1R13 system.

Tip: Use check ZOSMIGV1R13_ZFS_FILESYS(available with APAR OA35465 on z/OS V1R12 and z/OS V1R11).

zFS: Verify virtual storage usage (Required, as of R11 APAR OA33451)

Applying PTF UA55765 (zFS APAR OA33451) to z/OS V1R11 fixes a performance problem that occurs because of too many storage obtains and releases in zFS. The resolution of the problem involves obtaining a new block of storage at zFS initialization. This storage obtain is for approximately 60 MB.

If your zFS virtual storage usage is close to the limit of the zFS address space, this additional virtual storage request at zFS initialization could cause zFS to fail to initialize and not come up, or zFS may come up but with insufficient remaining storage to handle zFS requests such as mount. In these cases, you would likely see zFS message **IOEZ00662I: ZFS is low on storage.**

Migration action:

1. Verify that PTF UA55765 (zFS APAR OA33451) is installed.
2. Check zFS storage usage by using the operator command **MODIFY ZFS,QUERY,STORAGE**. If you compare the third line of data (USS/External Storage Access Limit) to the fourth line (Total Bytes Allocated (Stack+Heap+OS)), you will be able to see how close zFS is to using its maximum storage. The Total Bytes Allocated should be less than the USS/External Storage Access Limit. For example:

```
MODIFY ZFS,QUERY,STORAGE
IOEZ00438I Starting Query Command STORAGE.
zFS Primary Address Space Storage Usage
---------------------------------------
Total Storage Available to zFS: 1938817024 (1893376K) (1849M)
Non-critical Storage Limit: 1917845504 (1872896K) (1829M)
USS/External Storage Access Limit: 1875902464 (1831936K) (1789M)
Total Bytes Allocated (Stack+Heap+OS): 245559296 (239804K) (234M)
Heap Bytes Allocated: 213411011 (208409K) (203M)
Heap Pieces Allocated: 295003
Heap Allocation Requests: 295610
Heap Free Requests: 607
Heap Usage By Component
--------------------------
```

```
Bytes No. of No. of
Allocated Pieces Allocs Frees Component
---------- ------ ------ ------ ---------
66836 10 10 0 Interface
5112 8 12 4 Media Manager I/O driver
1828 5 5 0 Trace Facility
…
```

You can see that, in this case, the Total Bytes Allocated (234M) is much less than the USS/External Storage Access Limit (1789M). If the Total Bytes Allocated becomes greater than or equal to the USS/External Storage Access Limit, zFS will issue an IOEZ00662I message. If you see that the Total Bytes Allocated approaches the value of the USS/External Storage Access Limit, you should take steps to decrease your cache sizes using the z/OS UNIX zfsadm config command. See the z/OS Distributed File Service zSeries File System Administration (SC24-5989) for more information on the zfsadm command.

3. If zFS has failed to initialize and is not active, you should decrease some of your zFS IOEFSPRM settings, such as dir_cache_size, meta_cache_size, recovery_max_storage, token_cache_size, tran_cache_size, vnode_cache_size (especially if they are significantly larger than the default for these values) and restart zFS. If zFS is active but message IOEZ00662I has been issued, you can attempt to decrease the caches dynamically using the zfsadm config command. (You should also make the corresponding changes in your IOEFSPRM file for the next zFS restart.) Alternatively, you can stop and restart zFS after making cache size changes to your IOEFSPRM file.

As a general practice, it is a good idea to periodically check zFS storage usage by using the operator command **MODIFY ZFS,QUERY,STORAGE**.

DFS Migration Actions Pre-First IPL

DCE/DFS: Disable DFS Client initialization (Required-IF, as of R13)

Required, if you are using the DFS client.

The DFS client (DFSCM) is a physical file system that is started during z/OS UNIX initialization based on a FILESYSTYPE statement in the BPXPRMxx parmlib member. Starting with z/OS V1R13, the DFS client function is removed.

Migration action: If your installation uses the DFS client, you must remove the following statement from the BPXPRM*xx* parmlib member to prevent the client from initializing:

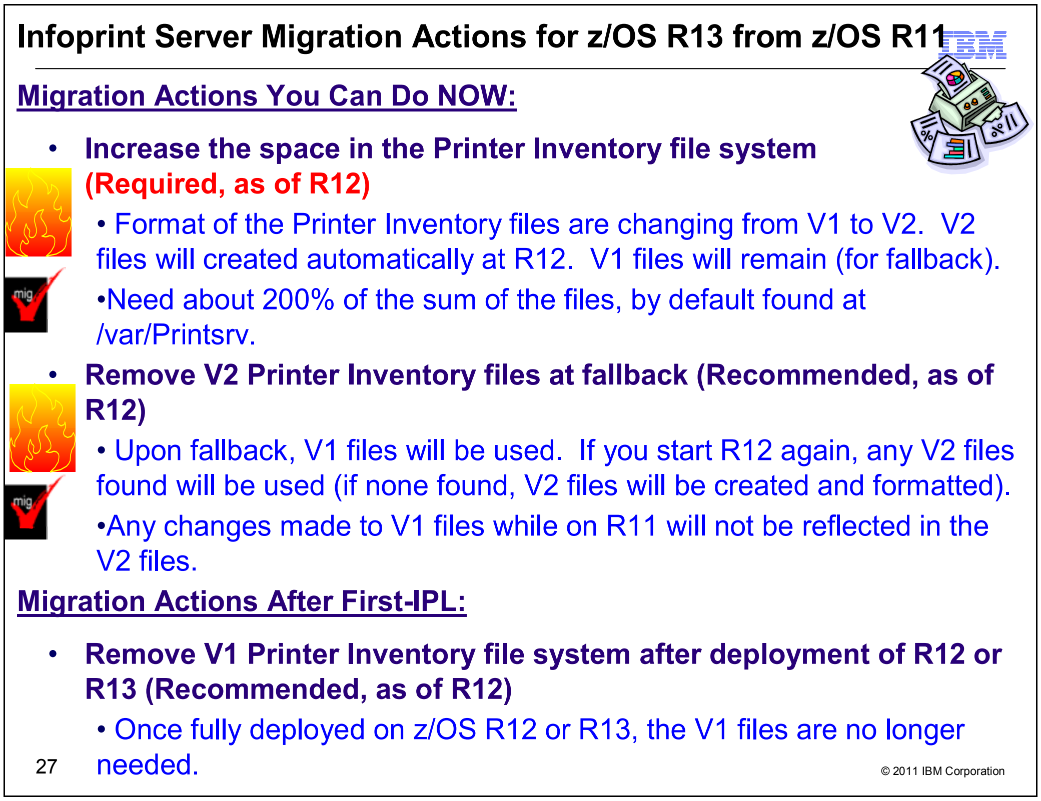
```
FILESYSTYPE TYPE(DFSC)
ENTRYPOINT(IOECMINI)
PARM('ENVAR("_EUV_HOME=/opt/dfslocal/home/dfscm") /
>DD:IOEDFSD 2>&1')
ASNAME(DFSCM)
```

If this migration action is not performed before the first IPL of z/OS V1R13, you will receive the following error message:

```
IOEP12402E: As of z/OS Version 1 Release 13, the DFS client function has been
removed.
```

z/OS UNIX will successfully initialize, but you will need to follow the guidance in the message to remove the entry and restart z/OS UNIX. If you have not already done so, you should use the z/OS UNIX **pax** command to migrate any data in DCE DFS or Episode file systems to other file systems. The recommended general procedure is as follows:

1. Set up a zFS file system to receive the data.
2. Copy your DCE DFS or Episode file system data to the zFS file system, using the z/OS UNIX **pax** command.
3. Set up a z/OS NFS server to allow data access from a remote z/OS UNIX system.



Infoprint Server Migration Actions for z/OS R13 from z/OS R11

Migration Actions You Can Do NOW:

- **Increase the space in the Printer Inventory file system (Required, as of R12)**
  - Format of the Printer Inventory files are changing from V1 to V2. V2 files will created automatically at R12. V1 files will remain (for fallback).
  - Need about 200% of the sum of the files, by default found at /var/Printsrv.
- **Remove V2 Printer Inventory files at fallback (Recommended, as of R12)**
  - Upon fallback, V1 files will be used. If you start R12 again, any V2 files found will be used (if none found, V2 files will be created and formatted).
  - Any changes made to V1 files while on R11 will not be reflected in the V2 files.

Migration Actions After First-IPL:

- **Remove V1 Printer Inventory file system after deployment of R12 or R13 (Recommended, as of R12)**
  - Once fully deployed on z/OS R12 or R13, the V1 files are no longer needed.

27

© 2011 IBM Corporation

Infoprint Server Migration Actions Between z/OS V1 R11 and z/OS V1 R13

Some migration actions from specific elements were selected from *z/OS Migration.* Many descriptions and actions have been shortened for inclusion in this presentation. For the complete descriptions and actions, refer to *z/OS Migration.*

Infoprint Server Migration Actions You Can Do Now

Increase space in the Printer Inventory file system (Required, as of R12)

In z/OS V1R12, the format of the Infoprint Server Printer Inventory files has changed from Version 1 to Version 2 format. When you start Infoprint Server on z/OS V1R13 for the first time, Infoprint Server reformats the Version 1 Printer Inventory files and creates Version 2 Printer Inventory files. The Version 1 Printer Inventory files are not removed so that if you need to fall back to the previous z/OS release, Infoprint Server can use the Version 1 Printer Inventory files. Therefore, the Printer Inventory file system requires more space in z/OS V1R13 than in previous releases. You might need to increase space in the Infoprint Server Printer Inventory file system.

On a z/OS V1R11 system, Infoprint Server cannot read or export the Version 2 Printer Inventory. If you want to use the Version 2 Printer Inventory on z/OS V1R11, use the **pidu** command to export the Version 2 Printer Inventory while running on z/OS V1R13 and then use the **pidu** command to import the exported copy to z/OS V1R11.

Migration action:

1. Do one of these:
   a. Run the IBM Health Check ZOSMIGV1R12_INFOPRINT_INVSIZE available with APAR OA32093..
   b. Run the **df** command to display the current utilization of the Printer Inventory file system. Printer Inventory files are located in the Infoprint Server base directory. The default base directory name is /var/Printsrv. You might have changed the base directory name in the **base-directory** attribute in the aopd.conf configuration file. The aopd.conf default location is /etc/Printsrv/aopd.conf. However, you might have specified a different location in environment variable AOPCONF. The free space required is 200% of the sum of the Version 1 Printer Inventory and historical Printer Inventory files (master.db, jestoken.db, pwjestoken.db, hinv/hinv.db, and logdb/log.db). If the “Capacity” is greater than 33%, increase the size of the file system.

2. To increase the size of the file system, you can use the z/OS UNIX **zfsadm grow** (zFS) or **confighfs** (HFS) command.

Set the **aggrfull** (zFS) or FSFULL (HFS) file system option so that warning messages are issued if the Infoprint Server base directory (/var/Printsrv) is getting full.

Remove Version 2 Printer Inventory files at fallback (Recommended, as of R12)

Not *required, but recommended if you want Infoprint Server to reformat the Version 1 Printer Inventory files after a fallback.*
In z/OS V1R12, the format of the Infoprint Server Printer Inventory files has changed from Version 1 to Version 2 format. When you start Infoprint Server on z/OS V1R13 for the first time, Infoprint Server reformats the Version 1 Printer Inventory files and creates Version 2 Printer Inventory files. Both Version 1 and Version 2 Printer Inventory files exist in the Infoprint Server base directory. Infoprint Server on z/OS V1R13 uses the Version 2 Printer Inventory files. If you fall back to z/OS V1R11, Infoprint Server uses the Version 1 Printer Inventory files.

If you start Infoprint Server on z/OS V1R13 a second time after falling back to a previous z/OS release, Infoprint Server uses the existing Version 2 Printer Inventory files that it created the first time you started Infoprint Server on z/OS V1R13. It does not reformat the Version 1 Printer Inventory files again. If you want Infoprint Server to reformat the Version 1 Printer Inventory files again, remove the Version 2 Printer Inventory files before you start Infoprint Server on z/OS V1R13. Because the Version 2 Printer Inventory files no longer exist, Infoprint Server reformats the Version 1 Printer Inventory files and creates a new set of Version 2 Printer Inventory files.

In most cases, you should remove the Version 2 Printer Inventory files if they exist. If you do not remove the Version 2 Printer Inventory files, any changes that the administrator made to the Version 1 Printer Inventory on z/OS V1R11 are not in the Version 2 Printer Inventory. In addition, Infoprint Central on z/OS V1R13 cannot display historical information for jobs that Infoprint Server processed on z/OS V1R11.
Migration action:

1. Run the IBM Health Checker for z/OS check INFOPRINT_V2DB_CHECK.
2. If Version 2 Printer Inventory files exist after falling back to z/OS V1R11 , remove them from the Infoprint Server base directory. Be careful not to remove any Version 2 files while running z/OS V1R13 because Infoprint Server on z/OS V1R13 requires Version 2 Printer Inventory files. Version 2 files have the extension "v2db". The default base directory is /var/Printsrv. You might have changed the base directory name in the **base-directory** attribute in the aopd.conf configuration file. The aopd.conf default location is /etc/Printsrv/aopd.conf. However, you might have specified a different location in environment variable AOPCONF.
   Example: These z/OS UNIX commands switch to an effective UID of 0, remove all files with the "v2db" extension from directory /var/Printsrv, and switchback to the original UID:
   ```
   su
   rm -f $(find /var/Printsrv/ -name "*.v2db")
   exit
   ```

Note: To remove Printer Inventory files, you must have an effective UID of 0 or be a member of the RACF AOPADMIN group.

Upgrade Java support for IPP Server (Required-IF, as of R12)

Required if you use IPP Server and specify the JAVA_HOME environment variable. You are using IPP Server if the start-daemons={ippd} attribute is specified in the Infoprint Server configuration file. The configuration file's default location is /etc/Printsrv/aopd.conf. However, you might have specified a different location in environment variable AOPCONF.
In z/OS V1R12, the Internet Printing Protocol (IPP) Server component of Infoprint Server requires Java V6.0. If the JAVA_HOME environment variable specifies the location of an earlier version of Java, you must update the JAVA_HOME environment variable.
Migration action:

1. Install IBM 31-bit SDK for z/OS, Java 2 Technology Edition, V6 (5655-R31).
2. If you use the IPP Server, edit the **aopstart** EXEC to update the directory path specified in the JAVA_HOME environment variable. IPP Server requires the 31-bit version of Java V6.0.
3. If you use the z/OS HTTP Server, update the setting of JAVA_HOME in the z/OS HTTP Server environment variables file httpd.envvars.

Note: If you installed Java V6.0 in the default Java directories, you do not need to specify the JAVA_HOME environment variable. If JAVA_HOME is not specified, IPP Server looks for Java files in the /usr/lpp/java/J6.0 directory.

Infoprint Server Migration Actions Pre-First IPL

Remount the Printer Inventory and copy files that were customized (Required)

When migrating to z/OS V1R13 Infoprint Server, you must bring forward the customized data from your previous system.
Migration action: Follow the instructions in *z/OS R13 Migration* for bringing forward the required files.

Update or remove the region size in the AOPSTART startup procedure (Required-IF, as of R13)

Required if you use the AOPSTART procedure that IBM provides and want to continue to use the default region size used in releases prior to z/OS R13.
Starting with z/OS V1R13, the Infoprint Server startup procedure AOPSTART specifies a region size of 512 megabytes. Before z/OS V1R13, AOPSTART did not specify a region size, so the default region size defined for your installation was used. Because the default region size might not be sufficient to use all the functions that Infoprint Server provides, it is a good practice to specify a region size on the startup procedure. However, if you want to continue to use the default region size or a region size other than 512 megabytes, edit the AOPSTART procedure to remove the region specification. If you have customized the AOPSTART procedure, you can continue to use the customized version.
Migration action:

1. Modify the EXEC statement in the AOPSTART procedure to remove or alter the REGION parameter:

```
//AOPSTART EXEC PGM=AOPBATCH,PARM='//usr/lpp/Printsrv/bin/aopstart',
// REGION=512M,
// TIME=NOLIMIT
```

2. Save the AOPSTART procedure.

Tips:

- The AOPSTART procedure is distributed in SYS1.IBM.PROCLIB. However, during installation it might have been copied to another data set in the started task PROCLIB concatenation.
- Specify a region size of at least 256 MB if you start the Infoprint Server Transform Manager to run data stream transforms. (This tip also applies to releases before z/OS V1R13.)
- Specify a region size of at least 200 MB if you start the Infoprint Server IPP Server to receive print requests from IPP-enabled clients. (This tip also applies to releases before z/OS V1R13.)
- User exits, such as IEFUSI, can modify the region size of an address space. Do not alter the region size of address spaces in the OMVS subsystem category.

Upgrade XML for Infoprint Central (Required-IF, as of R12)

Required if you use Infoprint Central. You are using Infoprint Central if the start-daemons={ssid} attribute is specified in the Infoprint Server configuration file. The file's default location is /etc/Printsrv/ aopd.conf. However, you might have specified a different location in environment variable AOPCONF.
In z/OS V1R12, the Infoprint Central component of Infoprint Server, which you can use to work with IP PrintWay extended mode print jobs and printers, requires the IBM XML Toolkit for z/OS V1.10 (5655-J51) product.
Migration action:

1. Install IBM XML Toolkit for z/OS V1.10 (5655-J51).
2. Specify the XML Toolkit for z/OS V1.10 libraries in the LIBPATH environment variable in your z/OS IBM HTTP Server environment variables file (default location is /etc/httpd.envvars). After z/OS V1R12 is installed, Infoprint Central requires the XML Toolkit for z/OS V1.10 libraries:
   - LIBPATH: change /usr/lpp/ixm/IBM/xml4c-5_6/lib to /usr/lpp/ixm/IBM/xml4c-5_7/lib
   - LIBPATH: change /usr/lpp/ixm/IBM/xslt4c-1_10/lib to /usr/lpp/ixm/IBM/xslt4c-1_11/lib
   - ICU_DATA: You can remove this variable because XML no longer uses this variable.
3. Restart the z/OS IBM HTTP Server to pick up the changes to the environment variables file.

Infoprint Server Migration Actions Post-First IPL

Remove Version 1 Printer Inventory files after deploying z/OS V1R13 (Recommended, as of R12)

Not required, but recommended to free up space in the Infoprint Server base directory.
In z/OS V1R12, the format of the Infoprint Server Printer Inventory files has changed from Version 1 to Version 2 format. When you start Infoprint Server in z/OS V1R13, Infoprint Server reformats the Version 1 Printer Inventory

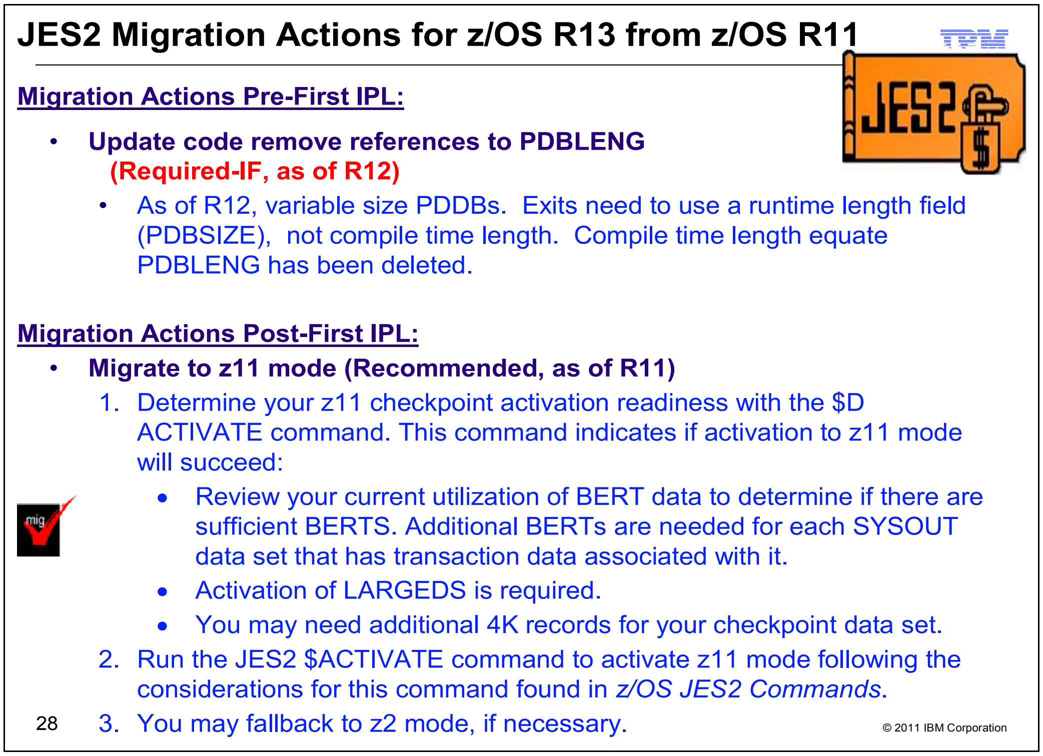
files and creates Version 2 Printer Inventory files. The Version 1 Printer Inventory files are not removed so that if you need to fall back to the previous z/OS release, Infoprint Server can use the Version 1 Printer Inventory files. After you have fully deployed z/OS V1R13 and are sure that you will not need to fall back to the previous z/OS release, you can remove the Version 1 Printer Inventory files to free up space in the Infoprint Server base directory. If you need to fall back to a previous z/OS release after removing the Version 1 Printer Inventory files, use the **pidu** command to export the Version 2 Printer Inventory on the z/OS V1R13 system and import the exported copy to the z/OS V1R11 system.

Migration action: Remove the Printer Inventory Version 1 database files in the Infoprint Server base directory. Version 1 files have a “db” extension. The default base directory is /var/Printsrv. You might have changed the base directory name in the **base-directory** attribute in the aopd.conf configuration file. The aopd.conf default location is /etc/Printsrv/aopd.conf. However, you might have specified a different location in environment variable AOPCONF.

Example: These z/OS UNIX commands switch to an effective UID of 0, remove all files with the “db” extension from directory /var/Printsrv, and switch back to the original UID:

```
su
rm -f $(find /var/Printsrv/ -name "*.db")
exit
```

Note: To remove Printer Inventory files, you must have an effective UID of 0 or be a member of the RACF AOPADMIN group.



JES2 Migration Actions for z/OS R13 from z/OS R11

Migration Actions Pre-First IPL:

- **Update code remove references to PDBLENG (Required-IF, as of R12)**
  - As of R12, variable size PDDBs. Exits need to use a runtime length field (PDBSIZE), not compile time length. Compile time length equate PDBLENG has been deleted.

Migration Actions Post-First IPL:

- **Migrate to z11 mode (Recommended, as of R11)**
  1. Determine your z11 checkpoint activation readiness with the $D ACTIVATE command. This command indicates if activation to z11 mode will succeed:
     - Review your current utilization of BERT data to determine if there are sufficient BERTS. Additional BERTs are needed for each SYSOUT data set that has transaction data associated with it.
     - Activation of LARGEDS is required.
     - You may need additional 4K records for your checkpoint data set.
  2. Run the JES2 $ACTIVATE command to activate z11 mode following the considerations for this command found in *z/OS JES2 Commands*.
  3. You may fallback to z2 mode, if necessary.

28

© 2011 IBM Corporation

JES2 Actions Between z/OS V1 R11 and z/OS V1 R13

These migration actions were taken from *z/OS Migration.* Some descriptions and actions have been shortened for inclusion in this presentation. For the complete descriptions and actions, refer to *z/OS Migration.*

JES2 Migration Actions You Can Do Now

Update code to remove references to PDBLENG (Required-IF, as of R12)

Required if installation exits use PDBLENG equate.

Starting with z/OS V1R12, JES2 supports variable size PDDBs, though the PDDBs generated in this release remain a fixed size. Installation exits that examine PDDBs or step through PDDBs using the compile time length of the PDDB need to be updated to use a run time length field. To facilitate locating an exit code that is assuming a fixed PDDB size, the compile time length equate PDBLENG has been deleted. Code that used this compile time length should be updated to use the run time field PDBSIZE to determine the size of the PDDB. The field PDBSIZE has correctly contained the length of the PDDB since z/OS V1R7 (the field existed in earlier releases but was not consistently set).

Migration action: Before installing z/OS V1R13 JES2, review installation exits for references to the field PDBLENG. If any references are found, the code needs to be updated to use the run time PDDB length field PDBSIZE.

Ensure calls to JES Property Information Services SSI can handle multiple members (Required-IF, as of R12)

Required if you are using the Initiator information function of the JES Property Information Services SSI.

In z/OS V1R11 JES2, the Initiator information function of the JES Property Information Services SSI (SSI 82) returned information for the local member only, even if multiple members matched the value that was specified on the member filter. In z/OS V1R12 JES2, if information for multiple members is requested by specifying wildcards on the member filter, the Initiator information function will return information for all members that match the filter request.

Migration action: Before installing z/OS V1R13 JES2, ensure that all calls to the Initiator information function of the JES Property Information Services SSI (SSI 82) that request information for multiple members can correctly handle information being returned for multiple members.

JES2 Migration Actions Pre-First IPL

<none>

JES2 Migration Actions Post-First IPL

Migrate to z11 mode (Recommended, as of R11)

Not required, but recommended to activate the full-function level of z/OS V1R11 JES2 processing.

If you wish to take advantage of the full-function level of z/OS V1R11 JES2, you must be in z11 mode. Activating z11 mode upgrades the JES2 checkpoint and enables JES2 functionality that is introduced in z/OS V1R11, including JOE data area extensions supported by BERTs. For more information on the JES2 functionality introduced in z/OS V1R11, see the reference links below.

Migration action:

- After migrating to z/OS V1R11 JES2, or later, on all systems in your MAS, determine your z11 checkpoint activation readiness:
  1. Use the **$D ACTIVATE** command. This command indicates if activation to z11 mode will succeed.
  2. Review your current utilization of BERT data to determine if there are sufficient BERTS, as detailed in “Check BERT utilization.”
  3. If you issue the **$ACTIVATE,LEVEL=z11** command, activation of LARGEDS support is required.
  4. An additional *nnn* 4K records for CKPT1 is required for z11 mode.
- Run the **JES2 $ACTIVATE** command to verify non-configuration changes that must be accommodated before going to z11, and to activate z11 mode following the considerations for this command found in *z/OS JES2 Commands*.

Note: The SPOOLDEF LARGEDS=FAIL (default value) in JES2PARM parmlib member is not supported in z11 mode. In z11 mode, on a COLD start, JES2 defaults to LARGEDS=ALLOWED. However, you cannot issue the $ACTIVATE,LEVEL=z11 command in the environment of SPOOLDEF LARGEDS=FAIL.

By default, JES2 restarts in the same mode (z2 or z11) as other members of the MAS (if any are active) or the mode the last active JES2 member was in when it came down. To restart JES2 in z2 mode, specify UNACT on PARM=. On a cold start JES2 starts in z11 mode unless overridden by OPTSDEF COLD_START_MODE.

Tip: Use IBM Health Checker for z/OS health check JES2_Z11_Upgrade_CK_JES2. This check determines if the system is ready to upgrade the JES2 checkpoint to z11 mode.

Check BERT utilization

Before issuing the $ACTIVATE,LEVEL=z11 command, review the current utilization of BERT data to determine whether there are sufficient BERTs. Additional BERTs are needed for each SYSOUT data set that has transaction data associated with it. These SYSOUT data sets can be seen using SDSF by setting APPC ON and examining SYSOUT data sets on the H and O panels; SYSOUT data sets with transaction data have nontraditional JES2 job IDs. Consider increasing the number of BERTs to correspond to two times the maximum number of transaction SYSOUT data sets on the system. BERT utilization should be monitored after the $ACTIVATE to z11 mode to ensure there are sufficient BERTs for the jobs and SYSOUT in the MAS. There are several ways to determine your current BERT usage:

- The $D CKPTSPACE,BERTUSE command displays a table of the types of control blocks in BERTs and how many BERTs are used by each control block type. The example below shows the output of the command:

  ```
  $HASP852 CKPTSPACE CURRENT BERT UTILIZATION
  $HASP852 TYPE COUNT CB COUNT
  $HASP852 -------- --------- ---------
  $HASP852 INTERNAL 11 1,
  $HASP852 JQE 211 108,
  $HASP852 CAT 114 38,
  $HASP852 WSCQ 1 1,
  $HASP852 DJBQ 0 0,
  $HASP852 JOE 0 0,
  $HASP852 FREE 763 0
  ```

  In the example, there are 108 JQEs that have a total of 211 BERTs associated with them. This example is for a system in z2 mode and does not have any BERTs associated with JOEs.
- The $D ACTIVATE command displays the number of BERTs that are needed for activation to z11 mode. This is the number of BERTs that will be associated with JOEs after the $ACTIVATE. The example below shows the output of the $D ACTIVATE command.

  ```
  $HASP895 $DACTIVATE
  $HASP895 JES2 CHECKPOINT MODE IS CURRENTLY Z2
  $HASP895 THE CURRENT CHECKPOINT:
  ```

```
$HASP895 -- CONTAINS 1100 BERTS AND BERT UTILIZATION IS 30
$HASP895 PERCENT.
$HASP895 -- CONTAINS 158 4K RECORDS.
$HASP895 z11 CHECKPOINT MODE ACTIVATION WILL:
$HASP895 -- EXPAND CHECKPOINT SIZE TO 165 4K RECORDS.
$HASP895 -- REQUIRE 22 ADDITIONAL BERTS AND UTILIZATION
$HASP895 WOULD REACH 32 PERCENT.
$HASP895 z11 ACTIVATION WILL SUCCEED IF ISSUED FROM THIS MEMBER.
```

In the example, there are 22 additional BERTs that will be used after the $ACTIVATE to z11 mode, for transaction data associated with JOEs.

Note: When the SPOOLDEF LARGEDS=FAIL (default value) is in effect in your JES2PARM parmlib member, the following message will be issued by the $ACTIVATE command:

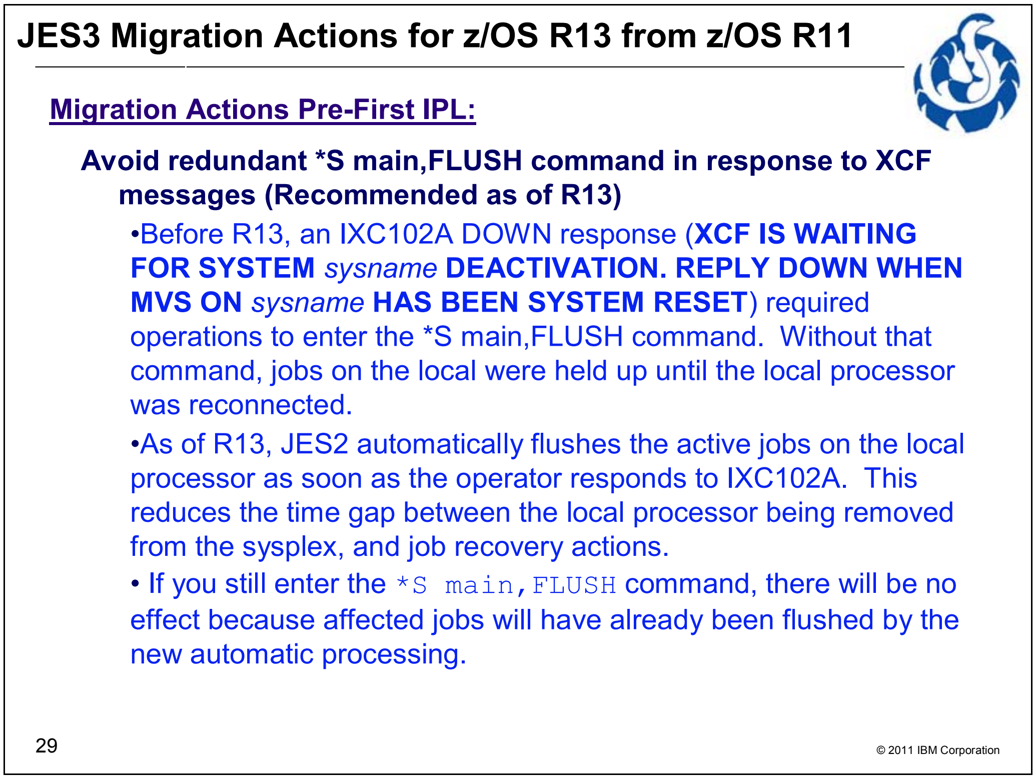
```
$HASP895 z11 ACTIVATION WILL FAIL IF ISSUED FROM THIS MEMBER.
$HASP895 THE FOLLOWING ISSUES PREVENT ACTIVATION:
$HASP895 -- LARGEDS SUPPORT MUST BE ACTIVATED.
```

- A general history of BERT usage can be obtained by using the $JD HISTORY(BERT) command or by using the SDSF RM panel. This displays the usage of BERTs after the system was IPLed. The example below shows the output of the $JD HISTORY(BERT) command:

```
$HASP9130 D HISTORY
$HASP9131 JES2 BERT USAGE HISTORY
DATE TIME LIMIT USAGE LOW HIGH AVERAGE
-------- -------- -------- -------- -------- -------- --------
2009.086 16:00:00 1100 337 337 337 337
2009.086 15:50:09 1100 337 125 337 192
```

Reference information:

- For a list of the enhancements introduced in z/OS V1R11 for z11 mode, see *z/OS Introduction and Release Guide*.
- For $ACTIVATE, $D ACTIVATE, $D CKPTSPACE and $JDHISTORY command details, see *z/OS JES2 Commands*.



JES3 Migration Actions for z/OS R13 from z/OS R11

Migration Actions Pre-First IPL:

Avoid redundant *S main,FLUSH command in response to XCF messages (Recommended as of R13)

- •Before R13, an IXC102A DOWN response (**XCF IS WAITING FOR SYSTEM** *sysname* **DEACTIVATION. REPLY DOWN WHEN MVS ON** *sysname* **HAS BEEN SYSTEM RESET**) required operations to enter the *S main,FLUSH command. Without that command, jobs on the local were held up until the local processor was reconnected.
- •As of R13, JES2 automatically flushes the active jobs on the local processor as soon as the operator responds to IXC102A. This reduces the time gap between the local processor being removed from the sysplex, and job recovery actions.
- • If you still enter the `*S main,FLUSH` command, there will be no effect because affected jobs will have already been flushed by the new automatic processing.

29

© 2011 IBM Corporation

JES3 Migration Actions Between z/OS V1 R11 and z/OS V1 R13

These migration actions were taken from *z/OS Migration.* Many descriptions and actions have been severely shortened for inclusion in this presentation. For the complete descriptions and actions, refer to *z/OS Migration.*

JES3 Migration Actions You Can Do Now

Modify code that depends on the format of suppressed split messages in the DLOG (Required-IF, as of R13)

Required if your installation has a code dependency on the format of messages in the JES3 DLOG.
The JES3 DLOG facility was introduced for tracking all message activity in a sysplex. It uses an MCS extended console to receive the messages and reformats them in the JES3 format. When these messages are longer than can be formatted into a single line, they are split into two lines. Prior to z/OS V1R13 JES3, longer messages having a receive ID were formatted differently if they were suppressed by the message processing facility (MPF). Beginning with z/OS V1R13 JES3, all suppressed messages with a receive ID are split in the same manner. The following DLOG excerpt shows how a suppressed message (XXX100I) was split prior to z/OS V1R13 JES3:

```
10250 1143599 SY1 R= XXX33913 ICH70001I IBMUSER LAST ACCESS AT 11:39:46 ON MONDAY, NOVEMBER 1,
2010
10250 1144001 SY1 R= XXX33913 IEF403I XXX33913 - STARTED - TIME=11.44.00
MLG 10250 1144004 &SY1 R= XXX33913 +XXX100I 9012345678901234567890123456
MLG 10250 1144004 &SY1 R= XXX3391378901234567890123456789012345678901234567890123456789012345
10250 1144004 SY1 R= XXX33913 IEF404I XXX33913 - ENDED - TIME=11.44.00
```

The following DLOG excerpt shows how a suppressed message (XXX100I) is split prior beginning with z/OS V1R13 JES3:

```
10305 1000486 SY1 R= XXX33913 ICH70001I IBMUSER LAST ACCESS AT 09:57:46 ON MONDAY, NOVEMBER 1,
2010
10305 1000488 SY1 R= XXX33913 IEF403I XXX33913 - STARTED - TIME=10.00.48
MLG 10305 1000494 &SY1 R= XXX33913 +XXX100I 9012345678901234567890123456
MLG 10305 1000494 &SY1 R= XXX33913 78901234567890123456789012345678901234567890123456789012345
10305 1000494 SY1 R= XXX33913 IEF404I XXX33913 - ENDED - TIME=10.00.49
```

Migration action: Ensure that any dependency on DLOG message formats are examined and corrected.

JES3 Migration Actions Pre-First IPL

Avoid redundant *S main,FLUSH command in response to XCF messages (Recommended, as of R13)

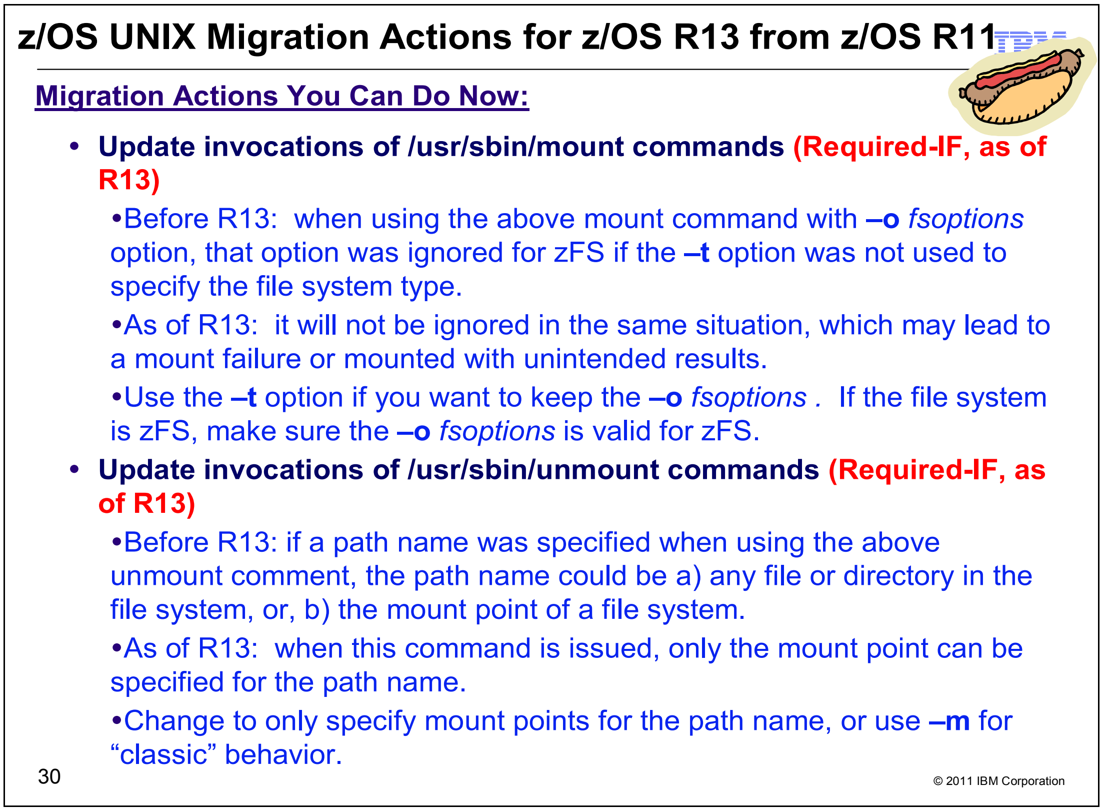
*Not required, but recommended to avoid redundant *S main,FLUSH commands.*
Prior to z/OS V1R13, a DOWN response to message IXC102A issuedfor a JES3 local processor required operations to enter the ***S main,FLUSH c**ommand. Without the ***S main,FLUSH** command, jobs on the local were held up until the local processor was reconnected.
Starting in z/OS V1R13, JES3 flushes the active jobs on the local processor automatically as soon as the operator responds to message IXC102A. This automatic flush eliminates the step of issuing the command and reduces the time gap between the local processor being removed from the sysplex and job recovery actions.
In z/OS V1R13 and later releases, if you run the ***S main,FLUSH** command in response to the XCF messages, the command will have no effect because the affected jobs will have already been flushed by the new automatic processing.
Migration action: If you want to avoid redundant ***S main,FLUSH** commands, remove the ***S main,FLUSH** command from all automated procedures or operating procedures.

Modify code that uses DATLOREC and DATINPTR (IATYDAT) as a programming interface

(Required-IF, as of APAR OA34642)
Required if you use DATLOREC and DATINPTR (IATYDAT) as a programming interface.
When a job's JCL contains in-stream data sets, the in-stream data is stored in multi-record files apart from the JESJCLIN file. These files are located by SYSIN pointer records that are written into a job's JESJCLIN file by input service. Prior to z/OS V1R12 (without APAR OA34642), SYSIN pointer records were included in the JESJCLIN file's record count which is saved in DATLOREC (IATYDAT). In z/OS V1R12 z/OS V1R11, and z/OS V1R10 (with APAR OA34642), the SYSIN pointer records are no longer included in the record count for JESJCLIN files. A new flag, DATCTLRD (IATYDAT), will be on for any record, including SYSIN pointer records, that are not included in a file's record count.
Migration action: Before applying the PTF for APAR OA34642, examine and modify any code that uses DATLOREC and DATINPTR (IATYDAT) as a programming interface. Note that IATYDAT is an internal JES3 control block that resides on spool and that this change affects only JESJCLIN data sets.

JES3 Migration Actions Post-First IPL

<none>

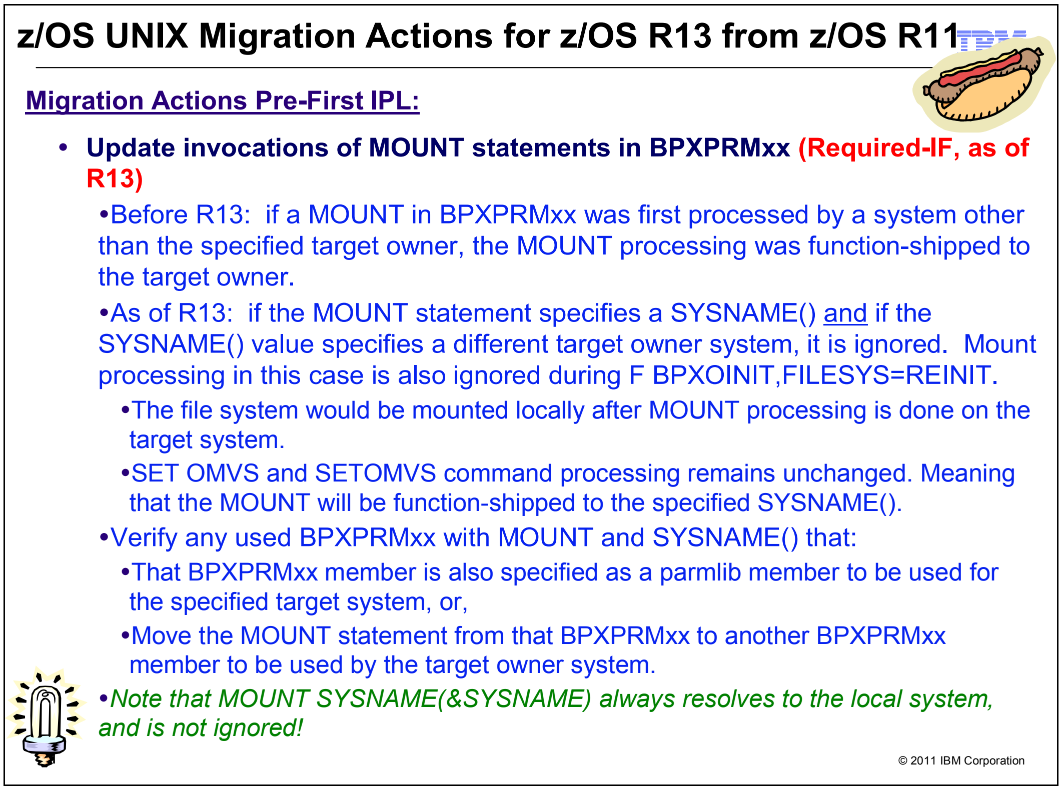


z/OS UNIX Migration Actions for z/OS R13 from z/OS R11

Migration Actions You Can Do Now:

- **Update invocations of /usr/sbin/mount commands (Required-IF, as of R13)**
  - Before R13: when using the above mount command with **–o** *fsoptions* option, that option was ignored for zFS if the **–t** option was not used to specify the file system type.
  - As of R13: it will not be ignored in the same situation, which may lead to a mount failure or mounted with unintended results.
  - Use the **–t** option if you want to keep the **–o** *fsoptions* . If the file system is zFS, make sure the **–o** *fsoptions* is valid for zFS.
- **Update invocations of /usr/sbin/unmount commands (Required-IF, as of R13)**
  - Before R13: if a path name was specified when using the above unmount comment, the path name could be a) any file or directory in the file system, or, b) the mount point of a file system.
  - As of R13: when this command is issued, only the mount point can be specified for the path name.
  - Change to only specify mount points for the path name, or use **–m** for "classic" behavior.

30



z/OS UNIX Migration Actions for z/OS R13 from z/OS R11

Migration Actions Pre-First IPL:

- **Update invocations of MOUNT statements in BPXPRMxx (Required-IF, as of R13)**
  - Before R13: if a MOUNT in BPXPRMxx was first processed by a system other than the specified target owner, the MOUNT processing was function-shipped to the target owner.
  - As of R13: if the MOUNT statement specifies a SYSNAME() and if the SYSNAME() value specifies a different target owner system, it is ignored. Mount processing in this case is also ignored during F BPXOINIT,FILESYS=REINIT.
    - The file system would be mounted locally after MOUNT processing is done on the target system.
    - SET OMVS and SETOMVS command processing remains unchanged. Meaning that the MOUNT will be function-shipped to the specified SYSNAME().
  - Verify any used BPXPRMxx with MOUNT and SYSNAME() that:
    - That BPXPRMxx member is also specified as a parmlib member to be used for the specified target system, or,
    - Move the MOUNT statement from that BPXPRMxx to another BPXPRMxx member to be used by the target owner system.
  - *Note that MOUNT SYSNAME(&SYSNAME) always resolves to the local system, and is not ignored!*

z/OS UNIX Migration Actions for z/OS R13 from z/OS R11

Migration Actions Pre-First IPL:

- **Accommodate changes to support read-only z/OS root for the cron, mail, and uucp utilities (Required-IF, as of R13)**
  - Before R13: certain post-installation customization had to be done every release for the cron, mail, and uucp utilities for the root to be mounted read-only.
  - As of R13: the /usr/lib/cron, /usr/mail, and /usr/spool directories are provided as symbolic links under the /var directory. /usr/lib/uucp directories will be symbolic linked under the /var directory. This positions the root to be mounted read-only “out of the box”, and may remove every release customization.
  - Note that previous documentation had suggested use of /etc, we have decided that /var is more appropriate, and the documentation (and the delivery of z/OS R13) uses /var.
  - Ensure you have enough room under the appropriate /var directory for usage of these utilities: create new file systems, change mount location of individual existing files system, or enlarge existing file systems mounted.

z/OS UNIX Migration Actions for z/OS R13 from z/OS R11

Migration Actions Post-First IPL:

- **Accommodate the new Shell and Utilities version of the** tsocmd **(Required-IF, as of R12)**
  - As of R12, **tsocmd** is included in Shell and Utilities. There are some differences. Tools and Toys download has the R12 level available so you can use a consistent level.
- **Remove MAXSOCKETS values from AF_UNIX in the BPXPRMxx parmlib member (Recommended, as of R12)**
  - Before R12: MAXSOCKETS was required in the NETWORK statement for AF_UNIX, if default of 100 was not acceptable.
  - As of R12: the maximum of 10,000 will be set. It will be ignored if specified on R12. Remove it when not sharing the BPXPRMxx member with pre-R12 systems.
- **Consider skulker invocations due to updated restriction (Required-IF, as of R12)**
  - Before R12: if there were objects whose path name included single quotes or newline character, those objects were not deleted, a warning message was issued, and processing stopped.
  - As of R12: there is no warning message, those objects are ignored, and processing continues.

z/OS UNIX Migration Actions Between z/OS V1 R11 and z/OS V1 R13

Some migration actions from specific elements were selected from *z/OS Migration.* Many descriptions and actions have been shortened for inclusion in this presentation. For the complete descriptions and actions, refer to *z/OS Migration.*

z/OS UNIX System Services Migration Actions You Can Do Now

Update invocations of /usr/sbin/mount commands (Required-IF, as of R13)

*Required because invocations of **/usr/sbin/mount** might fail , or if the mount does not fail, the results might be unpredictable.*

Before z/OS V1R13, when using the **/usr/sbin/mount** command with the **-o** *fsoptions* option, that option was ignored for zFS file systems if the **-t** option was not used to specify the file system type. Starting with z/OS V1R13, the **–o** *fsoptions* option is not ignored in that situation, which might cause the mount to (1) fail or (2) proceed but with possible unintended results.

Migration action: If you have invocations of the **/usr/sbin/mount** command that do not use the **-t** option to specify the file system type but specifies zFS-specific options using the **-o** *fsoptions* option, take the following actions:

1. If you want to keep the **-o** *fsoptions* option, determine the type of the file system and specify it, using the **-t** option.
2. If the file system is zFS, verify that the options string that was specified in **–o** *fsoptions* is valid.

Update invocations of /usr/sbin/unmount commands (Required-IF, as of R13)

*Required because invocations of **/usr/sbin/unmount** might fail if the path name is not the name of the mount point.*

Before z/OS V1R13, if a path name was specified when using the **/usr/sbin/unmount** command to unmount a file system, the path name could be (1) any file or directory in the file system or (2) the mount point of the file system. Starting in z/OS V1R13, when the **/usr/sbin/unmount** command is issued, only the mount point can be specified for the path name. Unmounting based on using any file or directory in a file system can sometimes cause the wrong file system from being unmounted. There can be instances, for example, where the file system has already been unmounted and a subsequent unmount on the same path name would result in accidentally unmounting the wrong file system.

Migration action: If you have invocations of the **/usr/sbin/unmount** command that do not specify a mount point on the *name...* parameter, follow these steps:

1. Look for instances where the specified path names in **/usr/sbin/unmount** invocations are files or directories.
2. Select one of the following actions:
    a. Use the **-m** option in **/usr/sbin/unmount** command if you have confirmed that something other than mount point needs to be specified. Note the file system that contains the file or the directory will be unmounted.
    b. Change the invocation so that only mount points are specified for the path name.

Review programs that invoke the BPX1EXM/BPX4EXM callable service (Required-IF, as of R12, and prior releases with OA30897)

Required if you have a program that fits the criteria described below.

Before z/OS V1R12 (and without the PTFs for APAR OA30897 installed on z/OS V1R11, z/OS V1R10, and z/OS V1R9), an unauthorized caller could pass a parameter greater than 100 characters on the BPX1EXM/BPX4EXM (execmvs) syscall. Beginning with z/OS V1R12 (and with the PTFs for APAR OA30897 installed to z/OS V1R11, z/OS V1R10, and z/OS V1R9), EC6 abends with reason code 0B26C048 will occur if: (1) the caller is not authorized, (2) the parameter is greater than 100 characters, and (3) the program would be executed in an authorized state.

Message BPXP027I goes to the system hardcopy log. See message BPXP027I for the program name and the argument length:

```
BPXP027I JOBNAME jobname, ATTEMPTED TO ISSUE AN EXEC OF THE APF AUTHORIZED MVS
PROGRAM, pgmname, WITH A PARM LENGTH OF xxx
```

Migration action: Review BPX1EXM/BPX4EXM calls from unauthorized callers that might call an APF-authorized program with an argument parameter greater than 100 characters.

- If the program can handle an argument greater than 100 characters, define a profile for the FACILITY class resource name BPX.EXECMVSAPF.*program_name*.

  - **Note:** Individual users do not need to be given access to the profile; the profile just needs to be defined. Also, the *program_name* defines the target APF-authorized program. .
- If the program cannot handle an argument greater than 100 characters, change the BPX1EXM/BPX4EXM calls to pass an argument that is 100 characters or less.

Accommodate new Shell and Utilities version of the tsocmd command (Required-IF, as of R12)

Required if you currently use the Tools and Toys version of the tsocmd command.

Before z/OS V1R12, the **tsocmd** command was obtained from the Tools and Toys section of the z/OS UNIX Web site. Starting with z/OS V1R12, Shell and Utilities support of the **tsocmd** command has been added. The supported version differs from the Tools and Toys version in a number of ways. For example:

- Exit values are consistently sent if the issued TSO/E command fails.
- The tsoin and tsoout environment variables are not supported. Instead, stdin and stdout are supported, as is done for most other Shell and Utilities commands.
- The TSOPROFILE environment variable is supported.

Migration action: Look for current use of the Tools and Toys version of **tsocmd**. If there is no current use of the Tools and Toys version of **tsocmd**, then no actions or changes are required.

If there is current use of the Tools and Toys version of **tsocmd**, determine if the command is located in /bin or in another directory. Also, determine if you want to preserve the Tools and Toys version in addition to the officially shipped version.

1. If you want to preserve the Tools and Toys version, ensure the Tools and Toys version of **tsocmd** is not located in /bin prior to the installation of z/OS V1R13.
2. If you do not want to preserve the Tools and Toys version, and it is located in /bin, then the installation of z/OS V1R13 will automatically replace the Tools and Toys version with the new officially supported version. If the Tools and Toys version is not located in /bin, remove it from its current location. In either case, you will also need to remove the Tools and Toys **tsocmd** load module from the authorized load library (either prefix.TSOCMD.LOADLIB or SYS1.LINKLIB) as described in the **tsocmd** Tools and Toys README documentation available at ftp://ftp.software.ibm.com/s390/zos/tools/tsocmd/tsocmd.readme.txt.

Note: A new version of **tsocmd** that matches the version shipped with z/OS V1R12 will be made available on the Tools and Toys Web site. If you have multiple systems at different releases and you want to have the same version of the tool on all releases, or if you want to try out the new command prior to z/OS V1R12, you can download the new version to the earlier systems and replace the previous Tools and Toys version of **tsocmd** if appropriate. If this action is taken, you will need to clean up the authorized load library as described in Step 2 prior to downloading the new tool.

Remove MAXSOCKETS values from AF_UNIX in the BPXPRMxx parmlib member (Recommended, as of R12)

Not required, but recommended to avoid confusion if users do not know that a value does not need to be specified for the MAXSOCKETS keyword on the NETWORK statement for AF_UNIX for z/OS V1R12.

Before z/OS V1R12, a value had to be specified for the MAXSOCKETS keyword in the NETWORK statement for AF_UNIX in the BPXPRM*xx* parmlib member of SYS1.PARMLIB if the maximum number of AF_UNIX sockets for the system needed to be greater than the default of 100. As of z/OS V1R12, the value does not need to be specified because a maximum value of 10,000 has been set for MAXSOCKETS for AF_UNIX. The MAXSOCKETS keyword is still allowed on the NETWORK statement for AF_UNIX, but will be ignored on a z/OS V1R13 system.

If you share your BPXPRM*xx* parmlib member with lower level systems, only remove your MAXSOCKETS statement from your AF_UNIX domain if you will accept the default MAXSOCKETS value for AF_UNIX for those lower level systems. (The default on z/OS V1R11 and z/OS V1R10 for MAXSOCKETS in the AF_UNIX domain is 100.) Otherwise, remove the MAXSOCKETS statement from the AF_UNIX domain when all systems that share the BPXPRM*xx* parmlib member are at z/OS V1R12 or later.

Migration action: Remove any MAXSOCKETS statement from NETWORK statements for DOMAINNAME(AF_UNIX). In a configuration with a shared BPXPRM*xx* parmlib, MAXSOCKETS should only be removed when all systems are at z/OS V1R12 or higher.

Migrate from HFS file systems to zFS file systems (Recommended)

Not required, recommended because the action is planned to become a requirement in a future release. zFS is the strategic file system for z/OS UNIX and continues to be enhanced to provide superior performance, reliability, and data integrity. IBM has not announced removal of support for the HFS file system.

Before z/OS V1R7, the HFS file system was the primary hierarchical file system. As of z/OS V1R7, you can use any combination of HFS and zFS file systems. Because zFS has higher performance characteristics than HFS and is the strategic file system, you should migrate your HFS file systems to zFS.

The HFS and zFS file system types in mount statements and command operands are now generic file system types that can mean either HFS or zFS. Based on the data set type, the system will determine which is appropriate. But note that you must still specify a type (HFS or zFS and it cannot be defaulted), and if the type you specify is not correct for the file system being mounted, any associated parameter string setting in the mount statement or command is ignored, even though the system sets the type correctly and processes the mount.

Understand the zFS recommendations and limits. For more information, see Minimum and maximum file system sizes in *z/OS Distributed File Service zSeries File System Administration*
Migration action:
1. Before beginning the migration, do the following:

- Establish backout procedures.
- Decide on naming conventions.
- Decide on unavailability.
- Understand any cloning or deployment changes required by zFS systems being linear data sets. Considerations would include any copy utility invocations, BPXPRMxx specifications for symbolics, and placement of zFS file systems on system volumes.

2. Perform the conversion from an HFS to zFS file system.
Tip: Use the BPXWH2Z tool to perform the conversion. It is an ISPF-based tool that migrates HFS file systems to zFS file systems. Using its panel interface, you can alter the space allocation, placement, SMS classes, and data set names. A HELP panel is provided. With this tool, you can:

- Migrate HFS file systems (both mounted and unmounted) to zFS file systems. If the HFS being migrated is mounted, the tool automatically unmounts it and then mounts the new zFS file system on its current mount point.
- Define zFS aggregates by default to be approximately the same size as the HFS. The new allocation size can also be increased or decreased.
- Have the migration run in TSO foreground or UNIX background.

You can use the JCL sample ISPBTCH in SYS1.SAMPLIB to invoke BPXWH2Z as an ISPF batch job. Before you run the job, be sure to read the Notes section. When you run BPXWH2Z on your z/OS V1R13 system, it uses the z/OS V1R13 level of the **pax** command. This level was enhanced in z/OS V1R7 for sparse file support and other characteristics that are of concern when migrating from an HFS to zFS file system. You can manually migrate from an HFS to zFS file system without using the tool. However, you would need to allocate and format the target zFS file systems.
Requirement: The BPXWH2Z tool requires the zFS address space to be operational. Therefore, before attempting to migrate existing HFS to zFS file systems using BPXWH2Z, make sure that the zFS address space has been successfully configured and initialized.
Tip: Consider migrating the sysplex root file system to zFS after you IPL z/OS V1R13. You can dynamically migrate the HFS sysplex root in a shared file system configuration to zFS while the root is in use, without disrupting workloads. Before z/OS V1R10, you could not migrate without disrupting active workloads. Although the shared file system configuration is required, the sysplex can be a single system. For the steps involved in migrating the sysplex root file system from HFS to zFS, see below.
Tip: Use check USS_HFS_DETECTED to verify all file systems mounted. This check issues exception message BPXH068E if any HFS file systems are found.
In z/OS R12! DFSMSdfp will allow a zFS data set to be recataloged with an indirect volume serial or system symbol. This is designed to allow the zFS file systems used for z/OS system software files (called version root file systems) to be cataloged using an indirect volume serial or a system symbol the same way as non-VSAM data sets to make cloning and migration easier.

3. Change policies and scripts, and so forth, to reflect the change from the HFS file system to zFS file system.
Tip: Use the RMF Monitor III option to report on zFS activity.

Migrating the sysplex root file system from HFS to zFS after IPLing z/OS V1R13:
Before you begin the migration:

- Ensure that the following requirements have been met:
  - All systems in the sysplex are at the V1R13 level.

  - The current sysplex root file system PFS, and the new sysplex root file system PFS, are up in all the systems in shared file system configuration.
- Be aware of the following restrictions:
  - The current sysplex root file system must be mounted as a read-only file system.
  - The systems that do not meet the requirements for this migration action cannot join the sysplex during the sysplex root file system migration processing, but they can join the sysplex after the sysplex root migration has completed.
  - The current sysplex root and the new sysplex root must be either HFS or zFS in any combination. If the new sysplex root is zFS, then it must be HFS-compatible.
  - The sysplex root or any directories on it cannot have been exported by the DFS or SMB server.
- Note the following:
  - Remote NFS mounts of the sysplex root or any directories on it are considered active use of the current sysplex root file system.
  - During the migration, the new zFS sysplex root file system must not be HSM-migrated, mounted, or in use.
  - Mount parameters are preserved during the migration or replacement of the sysplex root file system of the same file system type (PFS). They are dropped if the file system type is different.
  - Directories, data, files, and links are not copied from one file system to another.

Perform the migration as follows:
1. Ensure that a file system has been mounted read-only as the current sysplex root file system. When the root is mounted read-only, there are no function-shipping clients as long as physical paths to the DASD are available to each system. To verify that there are no function-shipping clients, issue: D OMVS,F,NAME=root_file_system_name You should see CLIENT=N on each system.

2. Allocate and set up the new zFS sysplex root file system:
   - Create a new zFS file system to be used as the new sysplex root file system. *z/OS Distributed File Service zSeries File System Administration* discusses creating and managing zFS file systems.

   Rules:
     - The UID, GID and the permission bits of the root directory in the new sysplex root file system must be same as the root directory in the current sysplex root file system.
     - If the SECLABEL class is active and the MLFSOBJ option is active, the security label for the new zFS file system must match the assumed security label of the current sysplex root file system.
   - On the new sysplex root file system, set up the active mount points and the symbolic links. The mount points and symbolic links must be the same as the ones on the current sysplex root file system. You can set them up either (1) manually or (2) by using the **pax** shell command to populate the new sysplex root file system using the existing sysplex root as a source. To do it manually, create a mount point in the existing sysplex root (for example, /newroot) and mount the new sysplex root file system in the MODE(RDWR) on that mount point. After mounting the new sysplex root file system, manually issue MKDIRs and ln -s to create the mount point directories and symbolic links similar to the existing sysplex root file system. Note that the new sysplex root file system must contain all active mount points and symbolic links exactly as on the existing sysplex root file system.
   - Use the **pax** shell command to populate the new file system, using the existing sysplex root as a source.

     Example:
     ```
     cd /
     pax -wr -pe -XCM ./ /newroot
     ```
     For more information about using **pax** to copy data from an HFS file system to a zFS file system, see *z/OS Distributed File Service zSeries File System Administration*.
   - Unmount the new zFS file system.

3. Before proceeding with this step, ensure PTF UA49782 for APAR OA29567 is installed on your z/OS V1R10 systems. Then, on any system in the shared file system configuration, issue:
F OMVS,NEWROOT=new.root.file.system.name,COND=<Yes|No>

   YES Proceed conditionally. The system checks for active usage in the current sysplex root file system and reports the active usage in a BPXF245I message. If file activity is found, the command fails with EBUSY return code and JrActivityFound reason code. If file activity is not found, the command continues processing to replace the sysplex root. YES is the default.

NO Proceed unconditionally. The system checks for active usage in the current sysplex root file system and reports the active usage in a BPXF245I message. Replacement of the sysplex root file system will continue.

The migration of the sysplex root file system will begin. During the migration, active connections to files and directories in the current sysplex root file system are broken.

After the migration completes:

- The root CWD('/') is updated on all systems in the sysplex to point to the new sysplex root file system.
- New opens go to the new sysplex root file system. The current sysplex root for the root directory is replaced for all processes in all systems. The current directory for root directory is replaced for any processes using it.
- Old connections in the previous sysplex root file system might get EIO errors.

4. Update the TYPE parameter and name of the sysplex root file system in the BPXPRM*xx* member of SYS1.PARMLIB.

z/OS UNIX System Services Migration Actions Pre-First IPL

Update invocations of MOUNT statements in the BPXPRMxx parmlib member (Required-IF, as of R13)

Required because required file systems might not be mounted.

Before z/OS V1R13, if the MOUNT statement in the BPXPRM*xx* member of SYS1.PARMLIB was first processed by a system other than the specified target owner, the MOUNT processing was function-shipped to the target owner. Starting with z/OS V1R13, if the MOUNT statement specifies a SYSNAME() keyword, the mount request is ignored during file system initialization and F BPXOINIT,FILESYS=REINIT processing if the SYSNAME() value specifies a different target owner system. The file system is then automatically mounted locally after the MOUNT statement is processed on the target system. The MOUNT statements in BPXPRM*xx* continues to be processed as part of SET OMVS and SETOMVS command processed without any changes; that is, the MOUNT processing continues to be function-shipped to the specified SYSNAME() target owner system.

Migration action: If you have a BPXPRM*xx* parmlib member that contains MOUNT statements with a SYSNAME() keyword specifying a specific target name, select one of the following actions:

1. Verify that the BPXPRM*xx* member is also specified as a z/OS UNIX parmlib member for the specified target systems.
2. Move the MOUNT statements from that BPXPRM*xx* member to a BPXPRM*xx* member that is used by the target owner system.

Note that specifying SYSNAME(&SYSNAME) always resolves to the system name of the local system; the MOUNT statement is processed as a result.

Accommodate changes to support read-only z/OS root for the cron, mail, and uucp utilities (Required-IF, as of R13)

Required if either of the following is true: 1) You have performed the post-installation activities to make **uucp**, **cron**, or **mail** supported for a read-only z/OS root. You do not necessarily have to be running with the z/OS root as read only but only have the post-installation customization as described in *z/OS UNIX System Services Planning*. 2) You have used **uucp**, **cron**, or **mail** facilities and have not performed the post-installation customization as described in *z/OS UNIX System Services Planning*.

Before z/OS V1R13, for each new release, certain post-installation activities had to be done for the **cron**, **mail**, and **uucp** utilities in order for the root file system to be mounted read-only. Starting in z/OS V1R13, the /usr/lib/cron, /usr/mail, and /usr/spool directories are provided as symbolic links.

Because the new z/OS V1R13 symbolic links are directed to a system-specific file system, other file systems are not affected by this change. When shared in read-only mode, the z/OS root will contain the /var symbolic links.

If you use any of the utilities (**uucp**, **cron**, or **mail**), then review any files that these utilities produce, such as logs. Now that the symbolic links, by default, point to the/var location, ensure that you have room for these utility-produced files in /var. This might mean: (1) creating a separate file system for files (such as /var/spool, as recommended in *z/OS UNIX System Services Planning*), (2) moving existing separate file systems you already have to be mounted under /var (from/etc, for instance), or (3) enlarging your /var file system if you don't have separately mounted file systems for these utilities.

Migration action: While this migration action should be performed before the first IPL of z/OS V1R13, the changes to use /var for this support can be done at any time. Although previous documentation had shown the use of /etc n examples, after further consideration, we now recommend using /var for these utilities.

1. If you currently use /etc or another directory for post-installation customization for these utilities, decide if you want to continue to use those directories or move to the /var structure that is provided with z/OS V1R13. Moving to the /var structure is recommended because:
   - You can minimize any subsequent post-installation customization, since the symbolic links to /var will be provided for you by IBM. Continued use of non-/var directories may mean post-installation work every time to remove the delivered structure and replace it with your own.
   - Continued use of /etc (or another directory), requires you to manage and maintain the symbolic links required from /var to that directory, which is “double symlinking.” This “double symlinking” might be confusing for those that maintain the system.
2. If you use /var for your post-installation customization, then ensure that the /var file system to be mounted at the z/OS V1R13 level of /var (or subdirectories shown below) contains the following directories or files for the utilities you are using. These will now be referenced by symbolic links in the z/OS V1R13 root file system when cron, mail, and uucp are used:

/var/mail
/var/spool
/var/spool/cron
/var/spool/locks
/var/spool/cron/atjobs
/var/spool/cron/crontabs
/var/spool/uucp
/var/spool/uucppublic
/var/spool/uucp/.Xqtdir
/var/spool/uucp/.Sequence
/var/spool/uucp/.Status
/var/cron
/var/uucp/Systems
/var/uucp/Devices
/var/uucp/Dialers
/var/uucp/Dialcodes
/var/uucp/Permissions
/var/uucp/config

Note that the /var file system provided by ServerPac will contain these directories but not the files, because the files are provided when the utilities are configured and used. If you use to use the ServerPac /var and add your files to that /var file system, that is acceptable. This action is similar to how you would merge an /var file system provided by Server Pac with an existing file system. If you have any user files under /usr/spool, /usr/lib/cron, or /usr/mail in the root file system (that is, they have not been symlinked under /var or another directory to support the read-only root), then those files must be moved to /var/spool, /var/cron, or /var/mail (or another directory besides the /var that you choose).

Discontinue use of invalid REXX variables in z/OS UNIX syscalls (Required-IF, as of R13)

Required if you use invalid REXX variables because they could cause problems that might not be noticed.

Before z/OS V1R13, issuing z/OS UNIX syscalls with invalid REXX variables might not generate an error message. Starting in z/OS V1R13, error messages are issued.

Migration action: If you have REXX programs with multiple syscalls that use the (*rexx-variable*) format of passing variables and do not check RETVAL and or ERRNO upon return, follow these steps.

1. Run these programs.
2. Fix parsing errors as they occur, replacing invalid REXX variables with valid ones. The error number will identify the invalid REXX variable. Note that REXX variable names cannot start with a digit or include arithmetic operators. For example, x*y is an invalid REXX variable.
   a. If the program produces a REXX parsing error, the error will be: `IRX0041I Error running ....... line xx: ......` This is a TSO REXX error message.
   b. If the program produces a z/OS UNIX syscall parsing error, the error code will be -21,-22,..., as described under the section "Returned from the SYSCALL environment" in the USS REXX documentation.

z/OS UNIX System Services Migration Actions Post-First IPL

Consider skulker invocations due to updated restriction (Required-IF, as of R12)

Required if you are concerned about not getting a warning message if objects in the specified directory or subdirectories whose path name includes single quotes or a newline character are found.

Before z/OS V1R12, if the **skulker** command was issued and there were objects in the specified directory (and in any subdirectories if the **-r** option was used) whose path name included single quotes or a newline character, those objects were not deleted, a warning message was issued, and all processing stopped. Starting with z/OS V1R12, a warning message is not issued, those objects are ignored, and processing continues.

Migration action: Use the **find** command to determine if there are any objects in the specified directory or subdirectories whose path name includes single quotes or a newline character. Do not use the **find** command to locate the objects whose path name include single quotes or a newline character as a standard practice before using the **skulker** command if you want the directories removed because issuing the **find** command will modify the access times of directories that are part of that **find** request.

Use the BPX.UNIQUE.USER profile instead of BPX.DEFAULT.USER (Recommended, as of R11)

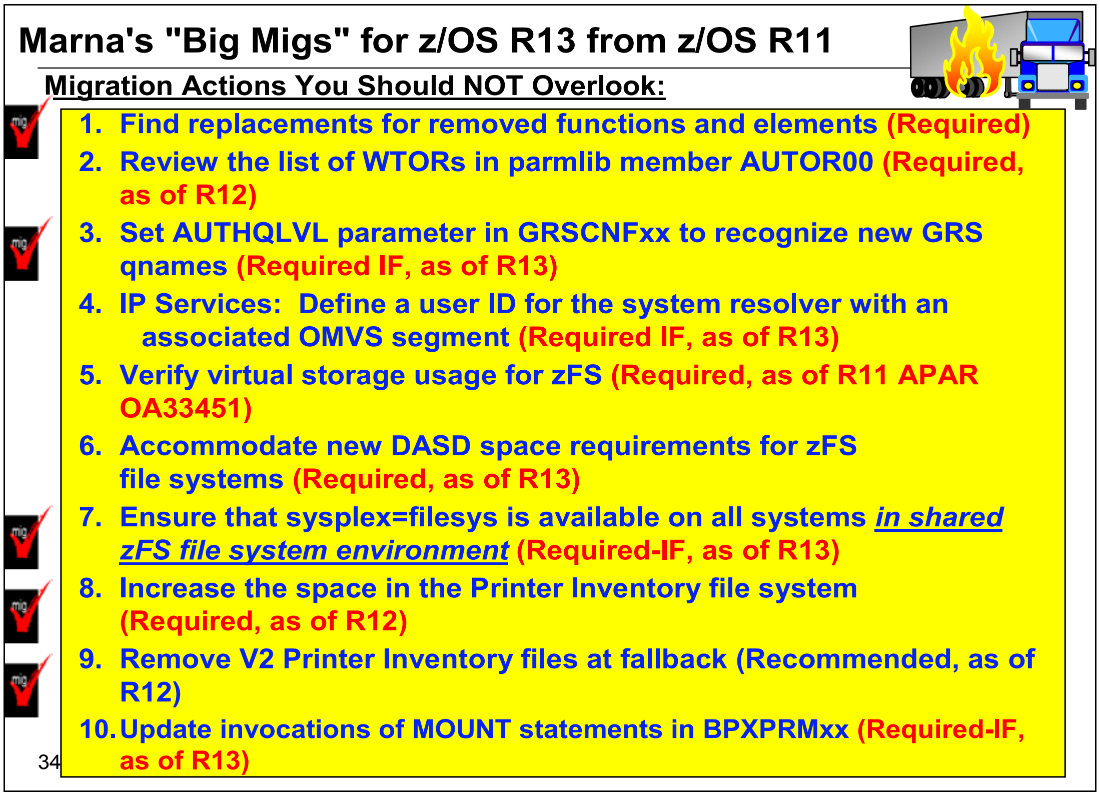
Recommended because in the release following z/OS V1R13, support for the BPX.DEFAULT.USER profile is planned to be removed. The use of BPX.UNIQUE.USER increases security because shared IDs are not used.

Before z/OS V1R11, if the BPX.DEFAULT.USER profile in the FACILITY class was defined, users who accessed z/OS UNIX services who did not have an OMVS user or group segment were assigned the default OMVS segments for the length of the user session. All users of the default OMVS segments shared the same UID and GID. As of z/OS V1R11, if BPX.UNIQUE.USER has been defined, users who access z/OS UNIX services who do not have an OMVS user or group segment are automatically assigned an OMVS segment with a unique UID and GID. The new OMVS segments are added to the user and group profiles in the RACF database.

Migration action: Follow the steps in *z/OS UNIX System Services Planning* to set up the BPX.UNIQUE.USER profile. If BPX.DEFAULT.USER has not been deleted, BPX.UNIQUE.USER takes precedence when default OMVS segments are used.

To remove the BPX.DEFAULT.USER profile, use the following RACF commands:

```
RDELETE FACILITY BPX.DEFAULT.USER
SETROPTS RACLIST(FACILITY) REFRESH
```

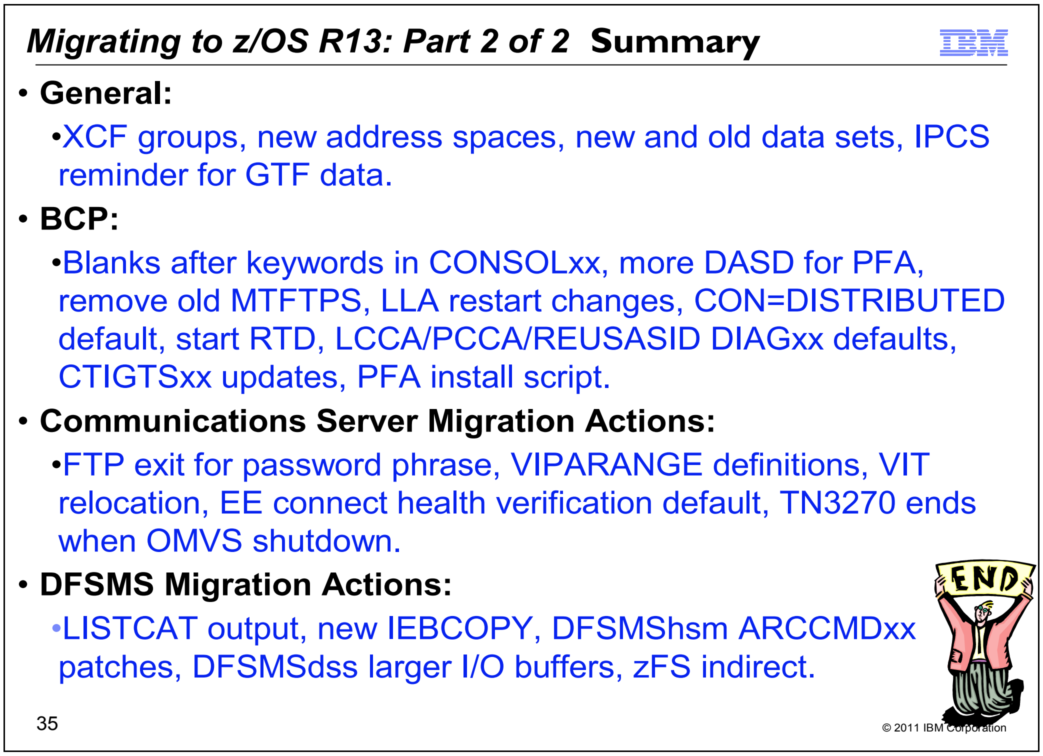


Marna's "Big Migs" for z/OS R13 from z/OS R11

Migration Actions You Should NOT Overlook:

1. Find replacements for removed functions and elements (Required)
2. Review the list of WTORs in parmlib member AUTOR00 (Required, as of R12)
3. Set AUTHQLVL parameter in GRSCNFxx to recognize new GRS qnames (Required IF, as of R13)
4. IP Services: Define a user ID for the system resolver with an associated OMVS segment (Required IF, as of R13)
5. Verify virtual storage usage for zFS (Required, as of R11 APAR OA33451)
6. Accommodate new DASD space requirements for zFS file systems (Required, as of R13)
7. Ensure that sysplex=filesys is available on all systems *in shared zFS file system environment* (Required-IF, as of R13)
8. Increase the space in the Printer Inventory file system (Required, as of R12)
9. Remove V2 Printer Inventory files at fallback (Recommended, as of R12)
10. Update invocations of MOUNT statements in BPXPRMxx (Required-IF, as of R13)

34

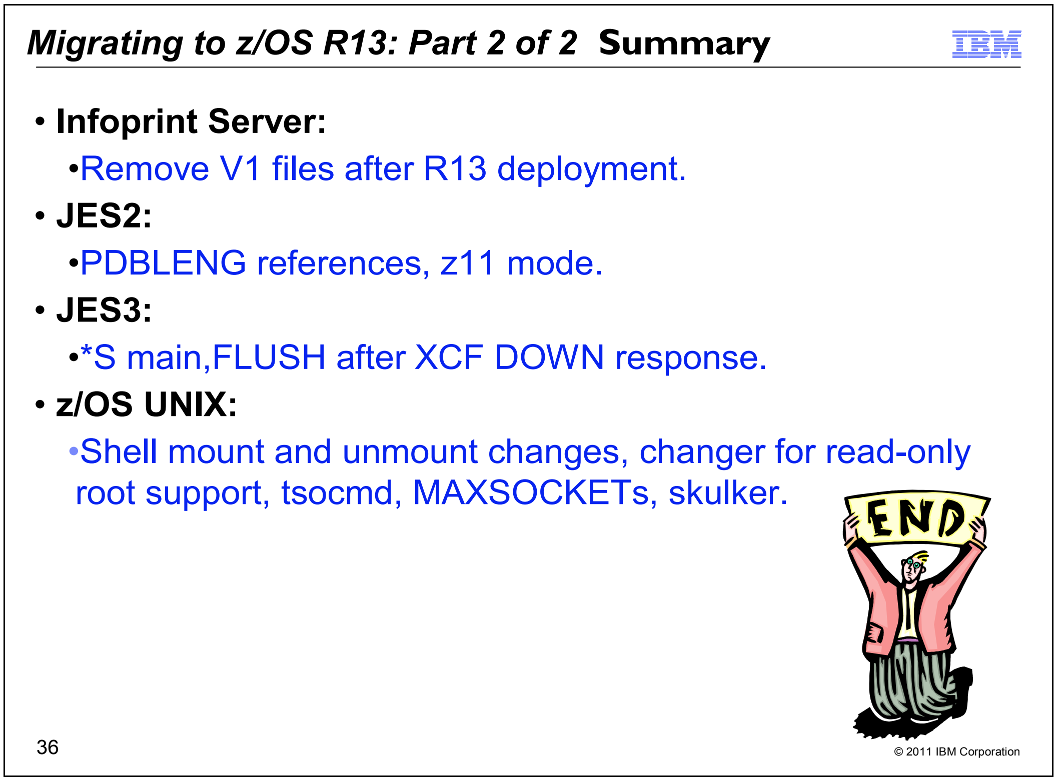


Migrating to z/OS R13: Part 2 of 2 Summary

- **General:**
  - XCF groups, new address spaces, new and old data sets, IPCS reminder for GTF data.
- **BCP:**
  - Blanks after keywords in CONSOLxx, more DASD for PFA, remove old MTFTPS, LLA restart changes, CON=DISTRIBUTED default, start RTD, LCCA/PCCA/REUSASID DIAGxx defaults, CTIGTSxx updates, PFA install script.
- **Communications Server Migration Actions:**
  - FTP exit for password phrase, VIPARANGE definitions, VIT relocation, EE connect health verification default, TN3270 ends when OMVS shutdown.
- **DFSMS Migration Actions:**
  - LISTCAT output, new IEBCOPY, DFSMShsm ARCCMDxx patches, DFSMSdss larger I/O buffers, zFS indirect.

35

© 2011 IBM Corporation



Migrating to z/OS R13: Part 2 of 2 Summary

- **Infoprint Server:**
  - Remove V1 files after R13 deployment.
- **JES2:**
  - PDBLENG references, z11 mode.
- **JES3:**
  - *S main,FLUSH after XCF DOWN response.
- **z/OS UNIX:**
  - Shell mount and unmount changes, changer for read-only root support, tsocmd, MAXSOCKETs, skulker.

36

© 2011 IBM Corporation