Abstract:

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 2.1 from either z/OS 1.12 or z/OS 1.13. Specifics of selected migration actions will be covered. It is strongly recommended that you attend both sessions for a complete migration picture.

The general availability date for z/OS V2R1 was September 30, 2013.
The general availability planned date for z/OS V2R2 is September 2015.

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.

Notice Regarding Specialty Engines (e.g., zIIPs, zAAPs and IFLs):

Any information contained in this document regarding Specialty Engines ("SEs") and SE eligible workloads provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g., zIIPs, zAAPs, and IFLs). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the "Authorized Use Table for IBM Machines" provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html ("AUT").

No other workload processing is authorized for execution on an SE.

IBM offers SEs at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

2 © 2015 IBM Corporation

Migrating to z/OS V2.1: Part 2 of 2 Agenda

- **Scope of presentation**
- **Definition of a "migration action"**
- **Overview of migration actions for z/OS V2R1 from z/OS R13 or R12:**
  - General Migration Actions
  - BCP
  - Communications Server
  - DFSMS
  - for FIPS 140 users
  - Distributed File Service - zFS
  - JES2
  - JES3
  - z/OS UNIX

3 © 2015 IBM Corporation

Scope of Presentation

- This presentation is applicable to z/OS V2R1 migrations from either z/OS R13 or R12.
- Not fully inclusive of all migration actions, but rather gives you an overview of migration actions that are:
  - Very important to understand
  - May be common to many users

- Remember: Use the *z/OS V2R1 Migration* book for a complete list of all migration actions.
  - The latest level is GA32-0889-03, and contains z13 information.
  - New format of the *z/OS Migration* book that is targeted to your specific migration path:
    - Chapter 1: Introduction   *for all users*
    - Chapter 2: General migration actions   *for all users*
    - Chapter 3: Migration from z/OS R13   *for R13 -> V2.1 users*
    - Chapter 4: Migration from z/OS R12   *for R12 -> V2.1 users*

Pick one!

4   © 2015 IBM Corporation

Definition of "Migration Action":

IBM

- 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**

🔥 Means "don't overlook!"	✔ Means some programmatic assistance is available	🧹 Means a cleanup action

5

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 V2R1 level of code to make these changes, and the changes don't require the z/OS V2R1 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 V2R1.
- *Pre-First IPL.* These are migration actions that you perform after you've installed z/OS V2R1 but before the first time you IPL. These actions require the z/OS V2R1 level of code to be installed but don't require it to be active. That is, you need the z/OS V2R1 programs, utilities, and samples in order to perform the migration actions, but the z/OS V2R1 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 V2R1 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 V2R1 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 V2R1. You need a running z/OS V2R1 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 V2R1 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.

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

means that this is a cleanup item or contains a portion that is a cleanup item. It is associated with something that is obsolete. It may cause confusion if someone thinks it does something. It is best to perform this action to avoid any confusion, since it is not needed anymore.

Elements with Migration Actions for z/OS V2R1

These elements have new V2R1 migration actions:

- ➤ BCP
- ➤ BookManager Build
- ▪ CIM
- ➤ Communications Server
- ➤ Crypto Services – ICSF, SSL
- ➤ DFSMS
- ▪ DFSORT
- ➤ Distributed File Service – zFS
- ▪ HCD
- ➤ IBM HTTP Server
- ▪ Infoprint Server
- ▪ ISPF

- ➤ JES2
- ➤ JES3
- ▪ Language Environment
- ▪ Library Server
- ▪ RMF
- ▪ SDSF
- ▪ Security Server (RACF)
- ▪ TSO/E
- ▪ XL C/C++
- ➤ z/OS Font Collection
- ➤ z/OS UNIX

➤ *means that some of that element's migration actions are discussed in these two presentations*

▪ *means that you need to look for that element's migration actions in the z/OS V2R1 Migration book*

6

© 2015 IBM Corporation

Migration Actions for Elements for z/OS V2R1

When migrating from z/OS R12 to z/OS R13, the specified elements in the slide above have required migration actions. Refer to *z/OS V2R1 Migration* for complete information on the required migration actions for all elements. Some migration actions for selected elements follow in this presentation.

General Migration Actions for z/OS V2R1

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

  - **Verify your IFAPRDxx Product ID is correct for z/OS V2 (Required)**
    - z/OS V2 has a new program number.
    - You can have both V1 and V2 features enabled, if sharing.
    - Failure to update IFAPRDxx may result in various initialization failures such as:

    ```
     IFA104I REGISTRATION HAS BEEN DENIED FOR PRODUCT
     WITH OWNER=IBM CORP NAME=z/OS
     FEATURE=TCP/IP BASE
     VERSION=.. ID=5650-ZOS
    ```

    - Use customized IFAPRDxx shipped with your z/OS V2 order, such as:

    ```
                    PRODUCT OWNER('IBM CORP')
                     NAME('z/OS')
                     ID(5650-ZOS)
                     VERSION(*) RELEASE(*) MOD(*)
                     FEATURENAME('SECURITY SERVER')
                     STATE(ENABLED)
    ```

7   © 2015 IBM Corporation

General Migration Actions for z/OS V2R1

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

  - **Accommodate new address spaces (Recommended)**
    - New in V2R1: **PCIE** (PCI Express) **and FPGHWAM** (HW Accelerator Mgr). Started automatically as of zEC12/zBC12 (regardless if zEDC or RoCE facilities are installed). They are persistent, and require no security customization. Expect either:
      - IQP002I PCIE INITIALIZATION COMPLETE, or
      - IQP031I REQUESTED SERVICE IS UNSUPPORTED BY HARDWARE
    - **IBM Health Checker for z/OS** and Generic Tracker, **GTZ**, (see later slides)
    - JES2 Converter/Interpreter **jesxCInn**, where jesx is the subsystem name, and nn is 01-25 for uniqueness. They are persistent, and used for interpretation process during conversion phase *only when* INTERPRET=JES is specified on JOBDEF. Default number of AS is 2, max is 25.   Proper userid must be assigned to each AS, using started proc tables (ICHRIN03) or a STARTED class profile has been defined to match AS names.

    - New in R13: **GPM4CIM** for cross-platform performance management with RMF XP. You may start GPM4CIM(s) with a START command.
    - Runtime Diagnostics address space, **HZR**, will be persistent. When you start it with the START command, it will remain active until stopped with a STOP
8   command. To analyze the system: MODIFY HZR,ANALYZE.   © 2015 IBM Corporation

General Migration Actions For z/OS V2R1

These migration actions were taken from *z/OS V2R1 Migration*. Some descriptions and actions have been shortened for inclusion in this presentation. For the complete descriptions and actions, refer to *z/OS V2R1 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 V2R1 systems during your migration, and allow back out from z/OS V2R1 if necessary. Use the SMP/E 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 V2R1 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 V2R1 systems and specify a Fix Category (FIXCAT) value of **"IBM.Coexistence.z/OS.V2R1"**. 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 V2R1, the size of the z/OS root file system is approximately 3300 3390 cylinders when HFS or 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.

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.

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:
- Enforced since z/OS V1R13, GTF data cannot be properly formatted on a lower level or upper level system.   For example, 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 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 V2R1 /etc and /var subdirectories.

Copy files from your old system to the z/OS V2R1 /etc and /var subdirectories, and then modify the files as necessary to reflect z/OS V2R1 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 V2R1 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 V2R1 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 tables in *z/OS Migration* 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 V2R1 and R13.

Add references to new data sets (Required)

Migration action: For z/OS V2R1, the following data sets and paths were added:
*   SYS1.FONTLIB for the z/OS Font Collection z/OS
*   SYS1.FONTLIBB for the z/OS Font Collection z/OS
*   /usr/lpp/booksrv/plugins/IBM/ for Library Server
*   SYS1.SFNTILIB for the z/OS Font Collection z/OS (use in place of SYS1.FONTOLN)
*   /usr/lpp/fonts/ worldtype/IBM/ for the z/OS Font Collection z/OS
*   SYS1.SFONDLIB for the z/OS Font Collection z/OS
*   /usr/lpp/hzc/ include/IBM/ in z/OS UNIX for zEnterprise Data Compression (zEDC).
*   /usr/lpp/hzc/lib/IBM/ in z/OS UNIX for zEnterprise Data Compression (zEDC)
*   EPH.AEPHPLIB for Library
*   EPH.AEPHTAB for Library Server
*   SYS1.AFNT300 for the z/OS Font Collection z/OS
*   SYS1.AFNTDLIB for the z/OS Font Collection z/OS
*   SYS1.AFNTILIB for the z/OS Font Collection z/OS
*   SYS1.AFNTLIBB for the z/OS Font Collection z/OS
*   SYS1.AFONTLIB for the z/OS Font Collection z/OS
*   SYS1.AFONTHFS for the z/OS Font Collection z/OS
*   SYS1.FONT300 for the z/OS Font Collection

The only data set or path that has been added to z/OS R13 was the following:
*   /usr/lpp/gpm/IBM 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.

The following address spaces are new for z/OS V2R1:
*   PCIE (PCI Express) and FPGHWAM (Hardware Accelerator Manager). They provide the infrastructure for PCI Express I/O and hardware accelerator activities. These address spaces are started automatically during z/OS initialization, if the appropriate z/OS PCIE facilities hardware is installed. They are persistent address spaces. If the PCIE address space is successfully initialized, the message is displayed:
    IQP002I PCIE INITIALIZATION COMPLETE
    If the required hardware is not installed, the following message is written to the hardcopy log:
     IQP031I REQUESTED SERVICE IS UNSUPPORTED BY HARDWARE .
    For information about the PCIE messages, see *z/OS MVS System Messages, Volume 9 (IGF-IWM)*. For information about the FPGHWAM (Hardware Accelerator Manager) messages, see *z/OS MVS System Messages, Volume 5(EDG-GFS).*
    PCIE and FPGHWAM do not require any security customization.

- IBM Health Checker for z/OS. As of z/OS V2R1 the system starts IBM Health Checker for z/OS address space automatically during system initialization.
- JES2 Converter/Interpreter. A new persistent address space is used when the interpretation process is performed for a job during the JES2 conversion phase. The address space is only created when INTERPRET=JES is specified on JOBDEF. The number of address spaces used depends on the CISUB_PER_AS setting on JOBDEF. The number of conversion processes (PCEDEF CNVTNUM=) divided by the number of subtasks per address space (CISUB_PER_AS) gives the number of address spaces created. The default number of created address spaces is 2 and the maximum number is 25. The name of the address spaces are jesxCInn where jesx is the JES2 subsystem name and xx is a number (from 01 to 25) to create uniqueness. This address space accesses the PROCLIB data sets defined in the JES2 start PROC and using the JES2 dynamic PROCLIB service. You must ensure that a proper user ID is assigned to the address space (presumably the same user ID that is assigned to the JES2 address space) using entries in either the started procedures table (ICHRIN03) or that a STARTED class profile has been defined that matches each new address space names. If you prefer, both the started procedures table and STARTED class profile might be in place. This action ensures that the correct user ID is assigned.
- Generic Tracker (GTZ). As of z/OS V2R1, GTZ replaces the Console Tracking Facility with enhanced capabilities to track use of system functions.

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.

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*.

Verify your IFAPRDxx Product ID is correct for z/OS Version 2 (Required)

Because z/OS has changed from Version 1 (V1) to Version 2 (V2), the IFAPRDxx parmlib member Product ID should reflect the new program number for z/OS. The Feature, Release, and Modification values also found in IFAPRDxx should be specified as "*", so those values do not have to change; only the Product ID is affected by this change.

If you are sharing the IFAPRDxx parmlib member between z/OS V1 and V2 systems, keep the V1 Product ID statements (for 5694-A01). You can have both the V1 Product ID and the V2 Product ID statements in the same IFAPRDxx parmlib member. After you have completed your migration to z/OS V2, you can remove the V1 Product ID statement from your IFAPRDxx parmlib member.

Migration action:

- If you use ServerPac, the customized IFAPRDxx parmlib member has been shipped to you in CPAC.PARMLIB. Verify that you are either using that parmlib member, or have copied its contents to a parmlib member you are using.
- If you are using CBPDO, use provided job ZOSREG in the RIMLIB, to place a customized IFAPRD00 member into your parmlib.

Note: Customers must ensure that the policy in IFAPRDxx enables only the version/release at which they are licensed. Use of and enablement of z/OS features is subject to the z/OS license terms and conditions and must be done with the knowledge of your asset manager according to the terms and conditions for z/OS. For additional license terms and conditions, see the "Usage Restriction" section of the *z/OS Licensed Program Specifications* .

Update your check customization for modified IBM Health Checker for z/OS checks (Recommend)

Not required, but recommended to ensure that your checks continue to work as you intend them to work.

Changes that IBM makes to the checks provided by IBM Health Checker for z/OS can affect any updates you might have made.

The checks that are new in z/OS V2R1 are:
- CATALOG_RNLS
- ICSF_COPROCESSOR_STATE_NEGCHANGE
- ICSF_MASTER_KEY_CONSISTENCY
- ICSFMIG_DEPRECATED_SERV_WARNINGS
- IOS_IORATE_MONITOR
- IOS_FABRIC_MONITOR
- RACF_AIM_STAGE
- RACF_CERTIFICATE_EXPIRATION
- RACF_UNIX_ID
- SUP_SYSTEM_SYMBOL_TABLE_SIZE
- SYSTRACE_MODE
- SYSTRACE_BRANCH
- OCE_XTIOT_CHECK
- USS_KERNEL_PVTSTG_THRESHOLD
- USS_KERNEL_STACKS_THRESHOLD
- VLF_MAXVIRT
- ZOSMIGV2R1_CS_GATEWAY
- ZOSMIGV2R1_DEFAULT_UNIX_ID

The checks that were changed by IBM in z/OS V2R1 are:
- ASM_LOCAL_SLOT_USAGE
- ASM_PLPA_COMMON_USAGE
- ASM_PLPA_COMMON_SIZE
- CATALOG_IMBED_REPLICATE
- RACF_**classname**_ACTIVE
- RACF_SENSITIVE_RESOURCES
- SLIP_PER
- VSM_CSA_LARGEST_FREE
- VSM_CSA_THRESHOLD
- VSM_SQA_THRESHOLD
- XCF_CF_STR_SCM_UTILIZATION
- XCF_CF_SCM_UTILIZATION
- XCF_CF_STR_MAXSCM
- XCF_CF_STR_MAXSPACE
- XCF_CF_STR_SCMMAXSIZE
- ZOSMIGV1R11_CS_DNSBIND

The checks that were deleted by IBM in z/OS V2R1 are:
- CEE_USING_LE_PARMLIB
- PFA_FRAMES_AND_SLOTS_USAGE

The following check was new in z/OS V1R13:
- ZOSMIGV1R13_RO_SYMLINKS

The checks that were changed by IBM in z/OS V1R13 are:
- SUP_HiperDispatch
- XCF_SFM_CFSTRHANGTIME

The checks that were deleted by IBM in z/OS V1R13 are:
- CSVTAM_VIT_DSPSIZE
- CSVTAM_VIT_SIZE

Migration action:
1. Look at the updated checks in *IBM Health Checker for z/OS: User's Guide*.
2. Review changes you made for those checks, in HZSPRM*xx* parmlib members, for example.
3. Make any further updates for the checks to ensure that they continue to work as intended.

General Migration Actions Post-First IPL

Migrating to z/OS 2.1: Part 2 of 2 (Migration Details)

BCP Migration Actions for z/OS V2R1

Migration Actions Pre-First IPL:

- **Convert your existing IBM Health Checker for z/OS set-up for automatic start-up (Required IF, as of V2R1)**
  - As of z/OS V2R1, Health Checker starts automatically at IPL!
  - First time user: follow steps in *IBM Health Checker for z/OS User's Guide*. (You may want to set all checks' WTOTYPE to HARDCOPY.)
  - Existing user:
    - Make sure your proc can be found at IPL (default HZSPROC), or specify the name in IEASYSxx HZSPROC=.
    - Remove any existing START HZSPROC (perhaps in COMMNDxx).
      - System rejects subsequent attempts to start Health Checker.
    - Recommended to specify your HZSPRMxx members in IEASYSxx HZS=(aa,bb,....) and use HZSPRM='PREV' on HZSPROC for easy restarts. HZSPROC HZSPRM='aa,bb,…' is still supported.
    - Optionally specify HZSPDATA(dsn) in HZSPRMxx.
  - Of course, WARNUND in IEASYSxx is supported for sharing ☺.

10

© 2015 IBM Corporation

BCP Migration Actions for z/OS V2R1

Migration Actions Pre-First IPL:

- **Move from the console tracking facility to the Generic Tracker (Required IF, as of V2R1)**
  - As of z/OS V2R1, Generic Tracker facility (GTZ) replaces the console tracking facility. Lots of enhancements in the Generic Tracker!
  - Existing users of CNZTRKR service do not need to change.
  - GTZ will auto-start and initialize at IPL;

```
GTZ0105I GENERIC TRACKER IPL-TIME INITIALIZATION STARTING
GTZ0106I GENERIC TRACKER IPL-TIME INITIALIZATION COMPLETE
GTZ0001I GENERIC TRACKER INITIALIZATION STARTING
GTZ0002I GENERIC TRACKER INITIALIZATION COMPLETE
```

  - Userid associated with GTZ needs access to read parmlib concatenation data sets.
  - Although it is started, tracking is disabled.
  - Must replace any existing controls and commands for the console tracking facility, with Generic Tracker controls and commands:
    - Support in IEASYSxx GTZ= points to GTZPRMxx members. SETGTZ and SET GTZ commands available.
    - GTZPRINT utility and GTZQUERY service available.
    - Convert CNIDTRxx to GTZPRMxx. Sample tool provided!
  - Of course, WARNUND in IEASYSxx is supported for sharing ☺.

11

© 2015 IBM Corporation

Migrating to z/OS 2.1: Part 2 of 2 (Migration Details)

BCP Migration Actions for z/OS V2R1

Migration Actions Pre-First IPL:

- **Accommodate increase in nucleus size with the z/OS V1R13 RSM Enablement Offering Web deliverable (Required IF, as of R13 with the Web deliverable, or V2R1)**
  - With the RSM Enablement Offering Web deliverable (on any server), the nucleus will increase in size by approximately 380K above the 16MB line.
  - Available private storage may affect your applications.
  - Review your current available private storage usage above the line. Ensure that this increase will not adversely affect your system.

- **Accommodate increase in ESQA with the z/OS V1R13 RSM Enablement Offering Web deliverable (Required IF, as of R13 with the Web deliverable, or V2R1)**
  - With the RSM Enablement Offering Web deliverable (on any server), there is an increased allocation of 24K (6 pages) in ESQA per CPU per LPAR. "Per CPU" includes general purpose CPs, zIIPs, and zAAPs.
  - Review your ESQA specification in IEASYSxx to ensure an increased allocation will not adversely affect your system. If you need to increase your ESQA specification, also review the effects on your current available private storage usage above the 16MB line..

12   © 2015 IBM Corporation

BCP Migration Actions for z/OS V2R1

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 release after z/OS V2.2.
  - 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, dynamically add consoles, … and more!

- **Review the list of WTORs in parmlib member AUTOR00 (Required)**
  - AUTOR00 is in support of the AutoReply function introduced in z/OS R12.
  - The IBM-shipped parmlib member contains several changes for z/OS R13 and V2R1, and should be reviewed.
  - If the WTORs listed in AUTOR00 are automated by your existing automation product, ensure that the replies in AUTOR00 are appropriate. Update with your own AUTORxx parmlib member.

13   © 2015 IBM Corporation

BCP Migration Actions for z/OS V2R1

Migration Actions Pre-First IPL:

- **Consider the new default value for the LOADxx DYNCPADD keyword (Required IF, as of V2R1)**
  - Before V2R1, LOADxx could enable CPUs to be added over the life of the IPL if the HW supported it (z10 or later). Default was for all CPUs, which was the minimum supported by a z/OS release (ex. 100) or the machine (ex. 80), and which could be asked for explicitly with DYNCPADD ENABLE.
  - As of z/OS V2R1, LOADxx keyword DYNCPADD now supports 1-4 char decimal value nnnn, for how many CPUs can be dynamically added. The default has changed to 16.
  - If LOADxx DYNCPADD is omitted (defaulted), the number of CPUs z/OS can dynamically add over the life of the IPL is 16.
    - If 16 is not sufficient, specify DYNCPADD nnnn. You cannot specify a value higher than the minimum of what z/OS or the HW supports.
  - If you specify DYNCPADD ENABLE, you will get maximum flexibility, but also will result in maximum storage usage and overhead.
    - z/OS V2R1 can support 256 CPUs, earlier releases supported 100.
  - LOADxx DYNCPADD nnnn cannot be shared with a pre-V2R1 system. That is the "nnnn" part does not coexist with pre-V2R1 systems.

14   © 2015 IBM Corporation

BCP Migration Actions for z/OS V2R1

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 ABEND330. 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   © 2015 IBM Corporation

BCP Migration Actions For V2R1

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.

Migrate from SNMP to z/OS BCPii for communication to the HMC or SE for z/OS Capacity Provisioning support (Required-IF, as of V2R1)

Required if you use the Windows-based Capacity Provisioning Control Center (CPCC)function.

As of z/OS V2R1 the Capacity Provisioning Manager no longer supports the System z API for communication with the Support Element (SE) or Hardware Management Console (HMC). The protocol used by System z API is based on IP network connection using SNMP. For z/OS V2R1 it is required to configure the Capacity Provisioning Manager for communication through the z/OS BCP Internal Interface (BCPii) protocol. The SE and HMC support for the System z API remain, and is not affected by this withdrawal of support for Capacity Provisioning. If you are currently using SNMP for the communication, it is required 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 V1R13 or z/OS V1R12 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).

Make accommodations for RACROUTE AUTH check for SLIP Command (Required-IF, as of V2R1)

Required if you are use a SLIP command with Action of REFAFTER or REFBEFOR and are using a security product that is affected by the additional RACROUTE AUTH check.

As of z/OS V2R1, a RACROUTE AUTH check is now performed for a SLIP command that is issued with action of REFAFTER or REFBEFOR. This change might affect your installation, depending on the security product you are using.

Migration action: If you use the z/OS RACF security product, there is no action to take. If you use another security product, contact your vendor to see if there is any support or changes that you need to make.

Remove the REPORTCOMPLETIONS option from the IEAOPTxx member (Recommended, as of V2R1 and with the PTF for APAR OA35428 on z/OS V1R13)

Not required, recommended to avoid the information message, if you ever used the REPORTCOMPLETIONS parameter.

As of z/OS V1R13, the support for the REPORTCOMPLETIONS=YES|NO parameter in IEAOPTxx member of parmlib has been removed. This option was introduced with OA34801 (2010/11/24) and removed with OA35428 July 7, 2011. REPORTCOMPLETIONS=YES was intended to provide a system-wide option for installations with a single large production environment, where all regions were exempted from being managed towards response time goals and if you wanted to obtain response time statistics with different granularity through service and report classes. In all other cases the usage of this parameter was not recommended.

The REPORTCOMPLETIONS=YES option has been superseded by the "Manage Region Using Goal Of Both" option on the classification rule that allows to manage CICS Terminal Owning Regions (TORs) in a very granular way.

Migration action: To clean up the IEAOPTxx parmlib member, remove the REPORTCOMPLETIONS. You can still set the REPORTCOMPLETIONS option, but it is ignored with the following message:

IRA800I OPT MEMBER IEAOPTxx KEYWORD ReportCompletions IGNORED, NO LONGER USED

Move BCPii API calls into your application instead of in BCPii ENF exits (Required-IF, as of z/OS V1R13 and z/OS V1R12, both with APAR OA37035)

Required if you have coded a BCPii API call from your ENF exit.

As stated in various IBM publications, non-SRB ENF exits need to avoid time-consuming processing. Coding an HWIEVENT ENF exit to execute BCPii APIs may result in multiple problems, such as delays with BCPii event notification processing when BCPii services are simultaneously being invoked. Starting with z/OS V1R12 and later, BCPii enforces this restriction, and BCPii API calls made from within a BCPii ENF exit are now rejected with return code HWI_UNSUPPORTED_ENVIRONMENT.

Migration action: If you have coded a BCPii API call from your ENF exit, move the BCPii API call into your application and have the BCPii ENF exit post the application when the event occurs. Your application program may now issue the BCPii API call from the user's thread. For an example of how to code a BCPii ENF exit, see the sample ENF event exit HWIXMCX1 in SYS1.SAMPLIB.

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 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.

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.

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

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.
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 AUTORxx 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.

Move from the console tracking facility to the Generic Tracker (Required-IF, as of V2R1)

Required if you have started or customized the console tracking facility.
Before z/OS V2R1, the console tracking facility was used to track use of system functions; however, it had some limitations. Starting with z/OS V2R1, the console tracking facility is being replaced with a new enhanced Generic Tracker. As of z/OS V2R1, support for the console tracking facility parmlib member, CNIDTRxx, is no longer provided. The Generic Tracker accepts GTZPRMxx parmlib members, and a tool to convert CNIDTRxx to the Generic Tracker GTZPRMxx is also provided.
It is assumed that the retired console tracking facility commands are typically issued manually and for diagnostic purposes only, not in any scripts or automation tools. The callable service CNZTRKR, compiled into executable programs, continues to be supported and the internal routing of any data passed to CNZTRKR and the Generic Tracker will be transparent to the existing CNZTRKR caller.
Migration action:
- Replace the use of any console tracking facility commands. Use COMMNDxx, automation scripts, or manually enter the commands on the console command line with their corresponding Generic Tracker (GTZ) counterparts. You can use the following mapping as a quick reference:
    - Instead of using COMMNDxx to start the console tracking facility, use the new system parameter GTZ (in IEASYSxx) to specify a GTZPRMxx member that specifies the SETGTZ TRACKING=ON command.
    - Instead of the DISPLAY command, consider using utility GTZPRINT or a user-written program with the service GTZQUERY to retrieve, store, and process current tracking data.
    - Instead of SETCON TRACKING={ON|OFF}, use SETGTZ TRACKING={ON|OFF}
    - Instead of SETCON TRACKING=ONWITHABEND, use SETGTZ DEBUG(ACTION=ABEND...)
    - Instead of DISPLAY OPDATA,TRACKING, use DISPLAY GTZ command, the GTZPRINT tool, or the GTZQUERY macro service.
    - Instead of SET CNIDTR=xx, use SET GTZ=xx or system parameter GTZ in IEASYSxx
- Instead of having any SETGTZ commands in COMMNDxx, consider putting them into GTZPRMxx parmlib members. You can use the SET GTZ command or the GTZ system parameter in IEASYSxx to select and execute the content of those GTZPRMxx.
- For any new applications use macro GTZTRACK instead of macro CNZTRKR. Consider converting any existing use of CNZTRKTR to GTZTRACK.
- Convert existing CNIDTRxx parmlib members to GTZPRMxx. See the sample GTZCNIDJ for how the GTZCNIDT conversion tool can help you automate this conversion.

- Associate a user ID with the GTZ address space and grant that user ID permission to access the PARMLIB concatenation. This is required when using new system parameter GTZ to identify GTZPRMxx parmlib members to be processed at GTZ startup. See the description of system parameter GTZ in *MVS Initialization and Tuning Reference* for details.

🔥 Convert your existing IBM Health Checker for z/OS set-up for automatic start-up (Required-IF, as of V2R1)

Required if you are currently using the IBM Health Checker for z/OS and wish to continue to use it as you have customized it. This migration action is strongly recommended for those that have not used the IBM Health Checker for z/OS on each system yet.

Before z/OS V2R1, IBM Health Checker for z/OS users had to perform a set-up procedure and start IBM Health Checker for z/OS manually. As of z/OS V2R1 the system starts IBM Health Checker for z/OS automatically. If you haven't started IBM Health Checker for z/OS before, you will probably see program exceptions. See the *IBM Health Checker for z/OS: User's Guide* for how to handle those exceptions.

Migration action: For first time users of IBM Health Checker for z/OS, follow the steps for Optimizing IBM Health Checker for z/OS in *IBM Health Checker for z/OS: User's Guide*.

For users with existing IBM Health Checker for z/OS set-ups, use the following migration actions to convert systems to the IBM Health Checker for z/OS automatic start-up:

1. Make sure the system knows the name of your hzsproc procedure if you renamed it from the default HZSPROC: The start-up procedure for IBM Health Checker for z/OS is called HZSPROC, by default. If you customized your hzsproc name, you must specify it to the system, using the new HZSPROC system parameter in IEASYSxx.

2. Remove any existing START HZSPROC invocations that start IBM Health Checker for z/OS and rely on the automatic start-up: Because IBM Health Checker for z/OS now starts automatically, you must look for instances of START HZSPROC invocations in your installation. For example, look for START HZSPROC invocations in the COMMNDxx, parmlib member. After the system brings IBM Health Checker for z/OS up automatically, the system rejects any attempt to bring up a second instance. The system issues one of the two following warning messages:
   ```
   HZS0101I – "...HEALTH CHECKER... IS ALREADY ACTIVE"
   ```
   or if the automatically started IPL-time instance is still initializing:
   ```
   HZS0116I – "...HEALTH CHECKER... START PENDING"
   ```

3. Change the way you specify the HZSPRMxx parmlib members you want the system to use. Before z/OS V2R1, users typically specified the HZSPRMxx parmlib members for IBM Health Checker for z/OS in the HZSPROC procedure. Now starting with z/OS V2R1, IBM recommends that you do the following to tell the system which members of HZSPRMxx to use:

➢ Specify the HZSPRMxx parmlib members for your installation in the new HZS system parameter of IEASYSxx. This provides the default for the automatic start of IBM Health Checker for z/OS at IPL-time.

➢ In your hzsproc procedure, default to or define HZSPRM='PREV':
   ```
   //HZSPROC PROC HZSPRM='PREV'
   //HZSSTEP EXEC PGM=HZSINIT,REGION=0K,TIME=NOLIMIT,
   // PARM='SET PARMLIB=&HZSPRM'
   //*HZSPDATA DD DSN=SYS1.&SYSNAME..HZSPDATA,DISP=OLD
   // PEND
   // EXEC HZSPROC
   ```

➢ HZSPRM='PREV' specifies the following:

- For the initial automatic start, the system will use the HZSPRMxx suffixes listed in the HZS system parameter.

- For manual restarts after the initial automatic start, IBM Health Checker for z/OS initially uses the HZSPRMxx parmlib members that were in effect just before the previous Health Checker instance was stopped. This action will in particular include any parmlib members specified through a MODIFY HZSPROC,ADD,PARMLIB or MODIFY HZSPROC,REPLACE,PARMLIB command, while this first instance was running.

   For example, assume HZSPRM=PREV was specified when that first instance was started and system parameter HZS was set to (00,01). Then this first instance would have initially used

HZSPRM00 and HZSPRM01. Now assume a MODIFY HZSPROC,ADD,PARMLIB=(02,03) was specified and then later this first instance is stopped. A manual restart, still with HZSPRM=PREV, will initially use HZSPRM00, HZSPMR01, HZSPRM02, and HZSPRM03, as in the previous instance before it was stopped.

If MODIFY HZSPROC,REPLACE,PARMLIB=(02,03) is used instead, the secondary instance initially only uses HZSPRM02 and HZSPRM03. Specifying HZSPRM='PREV' makes occasional manual restarts (after applying service, for example) easy and consistent.

4. Optionally specify an HZSPDATA data set for persistent data in the HZSPRMxx parmlib member: Before z/OS V2R1, you could only specify the HZSPDATA in the HZSPROC startup procedure. Now you can define your HZSPDATA data set in either the HZSPROC startup procedure or on the HZSPDATA parameter of the HZSPRMxx parmlib member.

Consider the new default value for the LOADxx DYNCPADD keyword (Required-IF, as of V2R1)

Required if the LOADxx DYNCPADD value is omitted (defaulted) and the default number of CPUs (16) z/OS will be able to dynamically add over the life of the IPL is not sufficient. Recommended, if you specify DYNCPADD ENABLE, which provides maximum flexibility but also results in maximum storage usage and overhead.

Before z/OS V2R1, through PARMLIB member LOADxx you could enable that CPUs be added to the configuration over the life of the IPL if the hardware supported such addition. The default was for all CPUs that could be configured to the LPAR, which was the minimum supported by the z/OS release (for example, 100) and the machine (for example, 80), and which could be asked for explicitly by DYNCPADD ENABLE.

In z/OS V2R1, the LOADxx keyword DYNCPADD now supports a 1-4 character decimal value nnnn that indicates how many CPUs z/OS is able to dynamically add over the life of the IPL. The default has changed to 16.

Notes:

- All system z hardware (z10 EC/BC and later hardware) supports dynamic CPU addition.
- When specifying the maximum number of CPs that z/OS can dynamically add with LOADxx DYNCPADD *nnnn*, this LOADxx cannot be shared with pre-V2R1 systems; that is, the *nnnn* parameter of DYNCPADD is not recognized by pre-V2R1 systems.
- Specifying DYNCPADD nnnn or taking the DYNCPADD 16 default allows z/OS to determine the number of CPUs that z/OS must be prepared to be dynamically added for the life of the IPL. Because z/OS can know the maximum CPU id that can be dynamically added for the life of the IPL at IPL, z/OS can obtain CPU array related storage based on the maximum number of CPUs that can be activated for the life of the IPL

Migration action: If the default limit of 16 CPUs that can be dynamically added is not sufficient, then indicate on your LOADxx DYNCPADD the number you desire. The maximum number of CPUs that can be added over the life of the IPL will be capped by the minimum between the highest CPU id hardware and the z/OS release supports. The DYNCPADD setting cannot be changed dynamically; it requires an IPL.

Review the changed messages associated with two digit CP IDs. Update any necessary automation or operator procedures to accommodate the two digits. Before z/OS V2R1, there was only one digit used for CP IDs.

Plan for the increase of the maximum number of supported CPUs to 256 (Required-IF, as of V2R1)

Required for programs using local constants or the z/OS constants for the number of CPUs the z/OS CPU infrastructure supports at the current or a specific level of z/OS

In z/OS V2R1, z/OS CPU infrastructure will support up to a maximum of 256 CPUs (CPU ids 0-255). Earlier releases of z/OS support up to 100 CPUs (CPU ids 0-99). Components or products allocating storage for CPU related arrays or bitmasks might require changes to support the V2R1 CPU infrastructure. Allocating CPU related arrays or bitmasks on a per CPU basis is done using one of the following:

- Run-time fields in the z/OS CVT (mapped by CVT) and ECVT (mapped by IHAECVT) control blocks representing the maximum CPU id a z/OS image can use for the life of the IPL. Products using run time fields will not require changes to support the V2R1 CPU infrastructure.
- Compile-time or assemble-time constants in the z/OS ECVT control block or within the product itself representing the maximum CPU id the z/OS CPU infrastructure supports. Products using compile-time or assemble-time constants will need to recompile at a minimum and may require code changes to support the V2R1 CPU infrastructure.

All products running on z/OS V2R1 must prepare to support all CPUs supported by the z/OS V2R1 CPU infrastructure (up to 256 CPUs with CPU ids 0-255). Products that support the z/OS V2R1 CPU infrastructure will be able to run on earlier z/OS releases whose CPU infrastructure supports a smaller number of CPUs.

Programs that do not support up to the maximum number of CPUs the z/OS infrastructure supports might not be able to work with all CPUs on the z/OS image. The system impacts are program dependent.

Plan for the new default TRACKDIRLOAD in PROGxx (Recommended, as of z/OS V2R1)

Recommended, not required, if you used the old default value of NOTRACKDIRLOAD and you want to retain it, you must explicitly specify NOTRACKDIRLOAD in PROGxx.
Starting with z/OS V2R1 the default in PROGxx has changed from NOTRACKDIRLOAD to TRACKDIRLOAD. With TRACKDIRLOAD, internal system processing can be enhanced. Using TRACKDIRLOAD might cause extra but generally beneficial overall processing.
Migration action: If you want to use the old default value, specify NOTRACKDIRLOAD in PROGxx.

Plan for new entries AXRINIT and AXRRXTSS in the program properties table (Recommended, as of V2R1)

Recommended to avoid an unintended override of the IBM shipped default PPT.
In z/OS V1R13 and earlier there were no entries in the program properties table for AXRINIT and AXRRXTSS to indicate that these programs needed to run privileged; system programmers had to manually add entries for AXRINIT and AXRRXTSS into SCHEDxx parmlib members. In z/OS V2R1, these entries are now included in the IBM supplied default program properties table, and you can remove the SCHEDxx specifications for AXRINIT and AXRRXTSS.
Migration action: Remove the specifications of AXRINIT and AXRRXTSS in SCHEDxx. The recommended action described in DOC APAR OA40519 is no longer needed in z/OS V2R1.

Plan for security changes to EXECIO restricting the REXX exec for allocating an internal reader (Required-IF, as of V2R1)

Required if the invoker of the System REXX exec wants to invoke EXECIO to submit JCL.
In z/OS V1R13 and earlier for a REXX exec that was running under System REXX (TSO=YES), the exec was able to allocate an internal reader and subsequently invoke EXECIO to submit JCL. As of z/OS V2R1, this function is restricted if the security product (RACF or equivalent) indicates that the invoker does not have authority to the entity JCL .
Migration action: Permit access to allow the System REXX exec that uses EXECIO to submit JCL for allocating an internal reader. The System REXX exec runs under the security environment as specified by the SECURITY keyword on the AXREXX invocation; the default is the invoker of the AXREXX macro. The invoker of the System REXX exec must have access to the JCL resource in the TSOAUTH resource class.

Accommodate increase in nucleus size for the z/OS V1R13 RSM Enablement Offering Web deliverable (Required-IF, as of R13 RSM Enablement Offering Web deliverable and V2R1)

Required if the increase in nucleus size will affect the private storage available to applications on your system above the 16MB line. The migration action is relevant on all servers.
With the introduction of the z/OS V1R13 RSM Enablement Offering Web deliverable, the nucleus will increase in size by approximately 380K above the 16MB line. Because of this increase, the available private storage might affect the way your applications run on your system.
Migration action: Review your current available private storage usage above the 16MB line using reports from RMF or an equivalent product. Ensure that an increase of 380K for the nucleus above the 16 MB line will not adversely affect your system. Adjust values accordingly.

Accommodate increase in ESQA with the z/OS V1R13 RSM Enablement Offering Web deliverable (Required-IF, as of R13 RSM Enablement Offering Web deliverable and V2R1)

Required if the increased allocation in ESQA will affect the private storage available to applications on your system. The migration action is relevant on all servers.
With the introduction of the z/OS V1R13 RSM Enablement Offering Web deliverable, there is an increased allocation of 24K (6 pages) in ESQA per CPU per LPAR. This increased allocation in ESQA per CPU includes general purpose CPs, zIIPs, and zAAPs regardless of the status (online, offline, configured, stand-by, and so forth) of the hardware. Because of this increase in ESQA, if you change the ESQA size, the available private storage may affect application execution on your system.
Migration action: Review your ESQA specification in IEASYSxx, to ensure that an ESQA increased allocation of 24K per CPU used on the LPAR will not adversely affect your system. If you need to increase your ESQA specification, you should also review the effects on your current available private storage usage above the 16 MB line using reports from RMF or an equivalent product. Adjust values accordingly.

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 and V2R1:
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*.

BCP Migration Actions Post-First IPL

Accommodate the SETLOAD xx,IEASYM command to update system symbols without initiating an IPL (Required-IF, as of V2R1)

Required if you are currently using the IEASYMUP module provided in SYS1.SAMPLIB or the SYMUPDTE routine to update system symbols.

Before z/OS V2R1, the downloadable SYMUPDTE routine and the IEASYMUP module in SYS1.SAMPLIB were provided as mechanisms to update system symbols without initiating an IPL. Starting with z/OS V2R1, the SETLOAD xx,IEASYM command is available to perform this task. In z/OS V2R1, the IEASYMUP module in SAMPLIB is updated to return with a RC=X'FFF', not having done the requested function. However, unless this IEASYMUP module is rebound, there is no way to prevent the usage of an old copy, or detect an update because of use of an old copy of the tool.

In z/OS V2R1 you should stop using the downloadable SYMUPDTE routine or the IEASYMUP module from samplib in your earlier release. Note that use of SYMUPDTE or IEASYMUP might produce incorrect results when used in conjunction with the SETLOAD xx,IEASYM command. The SYS1.LINKLIB program IEASYMU2 is instead provided via APAR OA42569 as a replacement for the function provided by IEASYMUP / SYMUPDTE on previous z/OS releases.IEASYMU2 is a supported program in z/OS, and has considerations when used with SETLOAD xx, IEASYM.

Notes:

- If you continue to use the SYMUPDTE routine or the IEASYMUP module, or if you use the IEASYMU2 program, system symbols will be updated and the system will continue to function properly until the SETLOAD xx,IEASYM command is used. Once SETLOAD xx,IEASYM has been issued, updates introduced by those other programs will be lost. You may continue to use the IEASYMU2 program again, even after a SETLOAD xx,IEASYM has been issued, but the same considerations will be in effect.
- Updates to the system symbol table through the SETLOAD xx,IEASYM command will occur only on the system issuing the command.
- When a SETLOAD command is issued and fails, messages issued by IEFPRMLB (logical parmlib service) that contain *jobname* and *stepname* will contain the master *jobname* and *stepname* of the last step that ran under master because the SETLOAD command runs under master but does not run as its own step. In this case, the error is related to the SETLOAD processing and NOT the step whose name appears in the message.

Migration action:

- Rebind the IEASYMUP module from the z/OS V2R1 SAMPLIB to disable the code or simply remove it from LINKLIB or your LNKLST library.
- If you have used the downloadable SYMUPDTE routine, remove it from your LINKLIB or your LNKLST library. Begin using SETLOAD xx,IEASYM command instead of these obsolete modules. Or change your JCL to use IEASYMU2 instead of IEASYMUP (and remove any joblib/steplib specification). IEASYMU2 verifies access through the same profile of IEASYMUP.* in the FACILITY class that IEASYMUP did, so there are no security definition changes from using IEASYMUP to IEASYMU2.

Migrate to GRS-managed FICON CTCs (Recommended, as of OA38230)

Not required, but recommended because it might be required in the future, as part of the hardware statement of direction. See the hardware release memorandum RFA z196 for further information. http://www-01.ibm.com/common/ssi/rep_ca/0/897/ENUS110-170/ENUS110-170.PDF . This migration action only applies to installations where GRS directly manages CTCs as specified in GRSCNFxx. There are two applicable configurations:

- *A GRS Ring where the GRS complex is greater than the sysplex, or*
- *A GRS Ring that does not utilize sysplex communications at all.*

If you are running GRS Star, or using embedded XCF communications links rather than GRS-managed CTCs, this migration action is not applicable to you.

Prior to OA38230, GRS could only manage CTCs running in basic mode. Starting with APAR OA38230 (on z/OS V1R11, z/OS V1R12, and z/OS V1R13), GRS can additionally manage FICON CTCs (FCTC), which run in extended mode.

Migrate GRS-managed basic-mode CTCs to GRS-managed FICON CTCs in order to run in extended mode. Note that GRS star continues to be the recommended configuration for global ENQ/DEQ processing because of its better performance and overall reliability, availability, and serviceability (RAS). If the installation requires a GRS ring, IBM

suggests that the GRS complex equals the sysplex, as it provides a more robust transport layer for its communications.

Migration action: There are two separate methods (rolling IPL and complex-wide IPL) to migrate to the updated GRSCNFxx members with the additional FICON CTC, depending on the needs of the installation.   Follow the detailed instructions in *z/OS Migration.*

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.

o   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.

Communications Server Migration Actions for z/OS V2R1

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.

Migration Actions You Can Do Before First IPL:

- **IP Services: Add FTP to your IKJTSO00 AUTHCMD and AUTHPGM NAMES section (Required IF, as of V2R1)** ~~removed as of PI13288~~

  - As of V2R1, the z/OS FTP client supports user exits. The FTP client uses Dynamic Exit services to see if you have installed an FTP client user exit, therefore the z/OS FTP client must be authorized.

  - Update your IKJTSO00 AUTHCMD and AUTHPGM names with FTP. You may use V2R1's SAMPLIB IKJTSO00 as an example. If it is not updated you may see CSVDYNEX DEFINE failures.

17   - Do **not** share this change between z/OS V2.1 and prior systems!   © 2015 IBM Corporation

---

Communications Server Migration Actions for z/OS V2R1

Migration Actions You Can Do NOW:

- **IP Services: Be aware of IP Fragment IDS enhancements (Required IF, as of V2R1)**

  - Prior to V2R1, When IP Fragment IDS attack type is enabled, only IPv4 traffic was monitored.

  - As of V2R1, IP Fragment attack type of the IDS will monitor both the IPv6 and IPv4 traffic for suspicious fragments. It also will check for overlays that change the data in the packet.

- **IP Services: Prepare for the additional of IPv6 support for policy-based routing (Required IF, as of V2R1)**

  - Prior to V2R1, if you are using policy-based routine on a stack running as dual-mode (IPv4 and IPv6) and didn't specify the source and destination IP address, it only applied to IPv4 packets.

  - As of V2R1,that same rule applies to both IPv4 and IPv6.

  - If you want it to continue to apply to only IPv4 packets, change the rule to have either the source or destination IP address be 0.0.0.0/0.

18   © 2015 IBM Corporation

Communications Server Migration Actions for z/OS V2R1

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: Understand the change in the support provided by the DVIPSEC parameter on the IPSEC statement in the TCP/IP profile (Required-IF, as of V2R1

Required if the TCP/IP profile for your stack specifies the IPSECURITY parameter on the IPCONFIG6 statement and also specifies the DVIPSEC parameter on the IPSEC statement.

Before z/OS V2R1, the DVIPSEC parameter on the IPSEC statement in the TCP/IP profile enabled Sysplex-Wide Security Associations (SWSA) for IPv4 on a stack that had IPCONFIG IPSECURITY specified in the TCP/IP profile. Support for SWSA for IPv6 was not provided in these releases. Beginning with z/OS V2R1, SWSA for IPv6 is supported. The DVIPSEC parameter on the IPSEC statement in the TCP/IP profile enables SWSA for IPv6 on a stack that has IPCONFIG6 IPSECURITY specified in the TCP/IP profile.

Migration action:
- If you have both of the following specified in your TCP/IP profile, be aware that SWSA for IPv6 will be enabled on your stack:
- The IPSECURITY parameter on the IPCONFIG6 statement
- The DVIPSEC parameter on the IPSEC statement
- If you have IPv6 TCP traffic that is protected by an IPSec Security Association (SA) with an IPv6 DVIPA endpoint, you can see the following changes:
- When an IPv6 DVIPA is moved during a planned or unplanned DVIPA takeover, new SAs are automatically reestablished with the same security service characteristics as the SAs that existed on the host that owned the DVIPA.
- IPv6 TCP traffic that is protected by an IPSec SA with a sysplex-distributed DVIPA endpoint can be distributed to target hosts.
- Ensure that you configure the appropriate IP security policy on the backup and target hosts.

IP Services: Be aware of IP Fragment attack type of the Intrusion Detection Services (IDS) enhancements to monitor both IPv4 and IPv6 traffic (Required-IF, as of V2R1)
Required if you are using IDS on a stack and the IP Fragment attack type is enabled.
Beginning in z/OS V2R1, IP Fragment attack type of the Intrusion Detection Services (IDS) is enhanced to monitor both IPv4 and IPv6 traffic for suspicious fragments. It is also enhanced further to check for overlays that change the data in the packet. Be aware that in z/OS V2R1, if you have the IP fragment IDS attack enabled, IPv6 traffic will now be monitored. In earlier releases, only IPv4 traffic was monitored.
Migration action: If you are using IDS on a stack and the IP Fragment attack type is enabled, be aware of the following information:
- Both IPv4 and IPv6 traffic are monitored for suspicious fragments.
- The IP Fragment attack type checking is enhanced to check for overlays that change the data in the packet, including changes to the length of the packet.

IP Services: Prepare for the addition of IPv6 support for policy-based routing (Required-IF, as of V2R1)
Required if you are using policy-based routing on a stack that is being run as a dual-mode stack (IPv4 and IPv6).
As of z/OS V2R1 policy-based routing is enhanced to route IPv6 traffic. In earlier releases a policy-based routing rule that did not specify the source and destination IP addresses only applied to IPv4 packets. Starting in z/OS V2R1, that same policy-based routing rule applies to both IPv4 and IPv6 packets.
Migration action: If you have a policy-based routing rule that specifies neither source IP addresses nor destination IP addresses, the rule will apply to both IPv4 and IPv6 packets. If you want the rule to continue to apply to only IPv4 packets, modify the rule to specify either a source or destination IP address of 0.0.0.0/0.

IP Services: Replace any GATEWAY statements in the TCP/IP profile with equivalent BEGINROUTES statements (Recommended, as of V2R1)
Not required, but recommended to take advantage of the latest support offered by BEGINROUTES/ENDROUTES.
As of z/OS V2R1 support for the GATEWAY statement in the TCP/IP profile is eliminated. The BEGINROUTES/ENDROUTES statement block was introduced in z/OS V1R1 Communications Server and replaces the GATEWAY profile statement for configuration of static routes. The GATEWAY statement has been stabilized since then, and has not been updated to support enhancements such as IPv6 and replaceable static routes. Additionally, the GATEWAY syntax is error prone.
Migration action: Replace any GATEWAY statements in your TCP/IP profile with the equivalent BEGINROUTES/ENDROUTES statement block.

In order to create BEGINROUTES statements from your existing specification, you can use IPCS. Run the TCPIPCS PROFILE report on a dump of the TCP/IP address space and static routes are presented as a BEGINROUTES/ENDROUTES statement block, even if you coded them by using a GATEWAY statement. However, you will have to execute the commands to know with certainty what changes to make.

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 queueing, 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: 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.

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.

Communications Server Migration Actions Pre-First IPL
IP Services: Ensure FTP is listed in the AUTHCMD and AUTHPGM NAMES section of your IKJTSOxx member of SYS1.PARMLIB (Required-IF, as of V2R1)

Required if you invoke the z/OS FTP client.

Beginning in z/OS V2R1, the z/OS FTP client supports user exits. The FTP client invokes z/OS Dynamic Exit Services (DES) to determine whether you have installed FTP client user exit EZAFCCMD or EZAFCREP. To invoke DES successfully, the program FTP must be APF authorized. Therefore, if you want to use the z/OS FTP client user exits, you must add FTP to the AUTHCMD and AUTHPGM NAMES section of your IKJTSOxx member of SYS1.PARMLIB.

Note: Before applying the PTF of TCP/IP APAR PI05065, you will receive the following messages during the FTP Server initialization if you do not add the FTP program to the AUTHCMD and AUTHPGM NAMES section of your IKJTSOxx parmlib member.

```
EZA1555I CSVDYNEX DEFINE failed for user exit EZAFCCMD, RETURN CODE x' 08'  REASON CODE x' 00000804'
EZA1555I CSVDYNEX DEFINE failed for user exit EZAFCREP, RETURN CODE x' 08'  REASON CODE x' 00000804'
```

Migration action: Add FTP in the AUTHCMD and AUTHPGM NAMES section of your IKJTSOxx member of SYS1.PARMLIB. In z/OS V2R1, SYS1.SAMPLIB(IKJTSO00) member contains the FTP definitions in both AUTHCMD and AUTHPGM NAMES section.

Note: SYS1.SAMPLIB(IKJTSO00) member was updated with TSO/E APAR OA45541 to remove the FTP definitions in both the AUTHCMD and AUTHPGM NAMES sections, which were added in z/OS V2R1.

Other system (coexistence or fallback) requirements: The IKJTSOxx change cannot be shared between z/OS V2.1 systems and pre-z/OS V2.1 systems. That is, do not share the IKJTSOxx change for FTP between z/OS V2.1 and prior systems.

System impacts: If you make the IKJTSOxx FTP change on your z/OS V1.12 or z/OS V1.13 system you may see various abends or problems. These problems may result in *message IKJ56652I You attempted to run an authorized command or program. This is not supported under the dynamic TSO/E environment*, or other related problems.

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 /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
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.

File Transfer Protocol Daemon (FTPD)	SEZAINST(FTPSDATA)	/etc/ftp.data	In z/OS V2R1, a new configuration statement is provided to specify that a type 119 SMF record of subtype 71 is collected for the FTP daemon configuration information when the FTP daemon starts.

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.

Communications Server Migration Actions Post-First IPL

DFSMSdfp Migration Actions for z/OS V2R1

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

  - **Do not use IEBCOPYO (Required-IF, as of V2R1)**
    - In R13: IEBCOPY was enhanced, and was no longer APF-authorized.
      - IEBCOPYO (the old APF-authorized form) was provided for fallback, if necessary. IEBDSCPY was an alias of IEBCOPYO.
    - As of V2R1, IEBCOPYO has been removed, and IEBDSCOPY is an alias of the non-APF authorized form of IEBCOPY.
    - Don't use IEBCOPYO, use IEBCOPY.
    - If you use IEBDSCPY, understand that you will invoke the non-APF authorized form of IEBCOPY. Make appropriate changes accordingly.

20

© 2015 IBM Corporation

DFSMShsm Migration Actions for z/OS V2R1

Migration Actions Pre-First IPL:

Accommodate default value change for dump VTOC copies (Required-IF, as of V2R1)

- Prior to V2R1, the default value was 2 for VTOCCOPIES of the DEFINE DUMPCLASS command.
- As of V2R1, the new default is 0 when hsm dumps copy pool volumes.
  - The default value of 2 remains when hsm dumps non-copy pool volumes.
- VTOC copies are not needed in a FRBACKUP/FRRECOV environment.
- If you want old behavior for FRBACKUP/FRRECOV environment, specify VTOCCOPIES(2) on DEFINE DUMPCLASS for each dump class used to dump copy pool volumes.

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.

21

© 2015 IBM Corporation

DFSMS Migration Actions for z/OS V2 R1

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: Obtain descriptive text in messages for Open/Close/End of Volume ABENDs (Required-IF, as of z/OS V1R13 with PTF for APAR OA37505)

Required if any of these conditions are true:
- *You have z/OS V1R13 without APAR OA37505 installed, and are using the OCE_ABEND_DESCRIP function in DEVSUPxx for verbose messages.*
- *If you have automation or programs that depend upon the type (single-line WTO as opposed to multi-line WTO) of the IECxxxx message for OPEN, CLOSE, end-of-Volume determinant ABENDs.*

z/OS V1R13 DFSMSdfp introduced a new system option to include descriptive text in the IECxxxx messages for OPEN, CLOSE, end-of-volume determinant ABENDs. The text from the message documentation manuals. This function is enabled by specifying in the MPFLST*xx* PARMLIB member: .MSGOPTION VERBOSE(x), where *x* is either Y or N, with N as the default. If you were using OCE_ABEND_DESCRIP in DEVSUPxx to obtain this

descriptive text before OA37505 was installed, then you need to convert to using MPFLSTxx after OA37505 is installed.

Note: Beginning in z/OS V2R1, or after applying the PTF for APAR OA37505 in z/OS V1R13, the VERBOSE message is displayed only in the JOBLOG. In addition, all OPEN, CLOSE and end-of-volume ABEND messages are issued using a Message Buffer Manager service, whether or not the new VERBOSE message function is enabled. One change this introduces is that the IECxxxx portion of the message always appears as a multi-line WTO message. Automated operation services that parse these messages may be affected.

Remove OCE_ABEND_DESCRIP from DEVSUPxx parmlib member; otherwise you receive the following informational message at IPL time: `IEA253I DEVSUP OCE_ABEND_DESCRIP NO LONGER HAS AFFECT. USE MPFLSTXX MEMBER`

Migration action: If you were using DEVSUP*xx* to obtain the ABEND descriptive text before OA37505 was installed, then you need to convert to using MPFLST*xx* to obtain this information after OA37505 is installed. In the MPFLST*xx* PARMLIB member, specify .MSGOPTION VERBOSE(Y) to enable display of the descriptive text.

Because OPEN, CLOSE and end-of-volume ABEND messages now include the IECxxxx portion of the message as a multi-line WTO message, any automated operation services that parse these messages might need to be adjusted. Examine such automated operation services and adjust them to handle the new format as needed. Here is an example of the output message you receive after you install APAR OA37505:

```
JOB00608 00000090 IEC141I 013-18,IGG0191B,IBMUSER9,S01,SYSUT1,5901,ZR1DT1, 651
651 00000090 SYS1.SAMPLIB(ZZZZZZZZ)
```

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).

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, 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: Do not use IEBCOPYO (Required-IF, as of V2R1)

Required if you used IEBCOPYO directly or through the alias IEBDSCPY

In z/OS V1R13, the IEBCOPY utility was enhanced, and one of the enhancements was that the utility was no longer APF-authorized. For users who needed to fallback from the new IEBCOPY to the previous APF-authorized form, IEBCOPY was retained in z/OS V1R13 under the name IEBCOPYO. The IEBDSCPY alias name for IEBCOPY was also revised to apply to IEBCOPYO for z/OS V1R13.

In z/OS V2R1, the IEBCOPYO form of the utility is eliminated and the IEBDSCPY alias now applies to the standard non-APF-authorized form of IEBCOPY.

Migration action:

- Any programs or jobs that called IEBCOPYO in z/OS V1R13 must be changed to call IEBCOPY in V2R1. It is not expected that you used IEBCOPYO on z/OS V1R13, unless you had a problem with the z/OS V1R13 level of IEBCOPY and had to fall back to the z/OS V1R12 level of IEBCOPY.
- Any programs or jobs that call the IEBDSCPY alias name in z/OS V2R1 will invoke the non-APF-authorized IEBCOPY. Make any appropriate changes if this affects your programs or jobs.

DFSMShsm: Accommodate default value change for dump VTOC copies (Required-IF, as of V2R1)

Required if you want the old default value of VTOCCOPIES(2) in a FRBACKUP/ FRRECOV environment.

Before z/OS V2R1, the system used a default value of 2 for the VTOCCOPIES parameter of the DFSMShsm DEFINE DUMPCLASS command. Starting with z/OS V2R1, if the VTOCCOPIES parameter is not specified, the system uses a default value of 0 when DFSMShsm dumps copy pool volumes. The default value of 2 will continue to be used when DFSMShsm dumps non-copy pool volumes. If the VTOCCOPIES parameter is never specified, the LIST DUMPCLASS output displays a VTOCCOPIES value of *** to indicate that no value is specified. VTOC copies are not needed in a FRBACKUP/FRRECOV environment.

Migration action: If you want to keep the default value of VTOCCOPIES(2) used before z/OS V2R1 in a FRBACKUP/FRRECOV environment, specify VTOCCOPIES(2) on the DEFINE DUMPCLASS command for each dump class used to dump copy pool volumes.

DFSMShsm: Update applications that depend on LIST command output (Required-IF, as of V2R1)

Required if your application depends on the output of the LIST DUMPCLASS command

The output of the LIST DUMPCLASS command now includes the value of the RECOVERRESET parameter. RECOVERRESET is a new parameter of the DEFINE DUMPCLASS command that specifies whether or not the data-set-changed indicator in the VTOC entry is reset for all data sets that are restored during full volume recover processing.

Migration action: Update applications that depend on the output of the LIST DUMPCLASS command to accommodate the new value for RECOVERRESET.

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.

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.

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.

DFSMS Migration Actions Post-First IPL

DFSMSdss: Accommodate new default behavior for full-volume and track restore operations (Required-IF, as of V2R1)

Required if you want DFSMSdss to unconditionally reset the data-set-changed indicator during full-volume and tracks restore operations.

The data-set-changed indicator (DS1DSCHA) in the VTOC indicates whether or not the data set has changed since its last backup. Before z/OS V2R1, during a full-volume restore operation, DFSMSdss unconditionally reset (turned off) the data-set-changed indicator for each data set restored to the target volume. During a tracks restore operation, if any VTOC track was restored, DFSMSdss might reset the data-set-changed indicator for all data sets on the volume. This applies to all VSAM and non-VSAM data sets and all SMS and non-SMS data sets.

With z/OS V2R1, the default behavior for full-volume and tracks restore operations has changed. By default, DFSMSdss now resets the data-set-changed indicator only if the RESET keyword was specified on the DUMP

command. Along with this change, a RESET keyword has been added to the RESTORE FULL and RESTORE TRACKS commands, which allows you to specify whether the data-set-changed indicator is to be reset. In addition, you can use the options installation exit routine, ADRUIXIT, to control the resetting of the data-set-changed indicator.

You can use RESET on the RESTORE command for any FULL or TRACKS dump taken with V2R1, or any previous releases. RESET(YES) and RESET(NO) will work as expected. However, RESET(DUMP) will act as RESET(NO) because there is no way to determine if the RESET keyword was specified at DUMP time on a down-level dump. The new RESET(DUMP) functionality is only available in V2R1. Note that the default behavior for DFSMSdss DUMP is unchanged when DFSMShsm invokes DFSMSdss to perform a full volume restore. By default, DFSMShsm will pass RESET(YES) to DFSMSdss. The other parameters might be specified through the DEFINE DUMPCLASS command.

On a down-level dump, RESET(DUMP) will act as RESET(NO) because there is no way to determine if the RESET keyword has been specified at DUMP time on a down-level dump. The new RESET(DUMP) function is only available in V2R1.
Migration action: To obtain the behavior in previous releases to allow DFSMSdss to unconditionally reset the data-set-changed indicator during full-volume and tracks restore operations, use RESET(YES) with the RESTORE FULL and RESTORE TRACKS commands.

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 object support.
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. If you use OAM object support, you must run CBRSMR1D even if you do not plan to use OAM file system support. 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.

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.

Migration Actions for z/OS V2R1   IBM

If you use FIPS 140, these migration actions may be applicable to you:

- **IP Services: Ensure ICSF is active before starting the NSS daemon in FIPS 140 mode** (V2R1)
- **IP Services: Ensure ICSF is active before starting the IKE daemon in FIPS 140 mode** (V2R1)
- **IP Services: Ensure ICSF is active before starting the Policy Agent when AT-TLS groups are configured in FIPS 140 mode** (V2R1)
- **System SSL: Ensure ICSF is available when running System SSL in FIPS 140-2 mode** (V2R1)
- **IP Services: Permit IKE daemon running in FIPS mode to use additional ICSF services** (V1R13)

Something in common: **As of z/OS V2R1, FIPS 140 support now requires ICSF services.**

23

© 201...

Migration actions for FIPS 140 users in z/OS V2R1

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*.

IP Services: Ensure ICSF is active before starting the NSS daemon in FIPS 140 mode (Required-IF, as of V2R1)

Required if the NSS daemon is configured in FIPS 140 mode.
As of z/OS V2R1 FIPS 140 support now requires ICSF services. If the NSS daemon is configured in FIPS 140 mode, the daemon will fail to activate if ICSF is not active. Ensure ICSF is started before starting the NSS daemon if it is configured in FIPS 140 mode.
Migration action: If the NSS daemon is configured in FIPS 140 mode, ensure ICSF is active prior to starting the NSS daemon.

IP Services: Ensure ICSF is active before starting the IKE daemon in FIPS 140 mode (Required-IF, as of V2R1)

Required if the IKE daemon is configured in FIPS 140 mode
As of z/OS V2R1 FIPS 140 support now requires ICSF services. If the NSS daemon is configured in FIPS 140 mode, the daemon will fail to activate if ICSF is not active. Ensure ICSF is started before starting the NSS daemon if it is configured in FIPS 140 mode.
Migration action: If the IKE daemon is configured in FIPS 140 mode, ensure ICSF is active prior to starting the IKE daemon.

IP Services: Ensure ICSF is active before starting the Policy Agent when AT-TLS groups are configured in FIPS 140 mode (Required-IF, as of V2R1)

Required if AT-TLS groups are configured in FIPS 140 mode.

As of z/OS V2R1, FIPS140 support now requires ICSF services. Ensure ICSF is started before starting AT-TLS groups with FIPS140 support enabled. ICSF services will be used for random number generation and for Diffie Hellman support for generating key parameters, key pairs and key exchanges.

Migration action:
1. Ensure ICSF is active before starting AT-TLS groups configured to support FIPS140-2
2. If the CSFSERV class is defined, give READ access to the userid associated with the TCPIP stack and any application userid using the TTLSGroup to the CSFRNG resource within the RACF CSFSERV class.
3. If the CSFSERV class is defined and Diffie Hellman is being used, give READ access to the application userid to the CSF1TRC, CSF1DVK, CSF1GKP, CSF1GSK, CSF1GAV, and CSF1TRD resources within the RACF CSFSERV class.

System SSL: Ensure ICSF is available when running System SSL in FIPS 140-2 mode (Required-IF, as of V2R1)

Required if your installation runs System SSL in FIPS mode.

In z/OS V2R1, System SSL, when running in FIPS 140-2 mode, uses ICSF's random number generation and Diffie-Hellman support. Before running System SSL in FIPS 140-2 mode you must ensure that ICSF is running and that all user IDs that start SSL applications in FIPS 140-2 mode, invoke the gskkyman utility to manage FIPS 140-2 key database files, or invoke the GSKSRVR started task in FIPS mode have access to certain CSFSERV classes.

When it is running in non-FIPS mode, System SSL uses its own implementation of Diffie-Hellman and does not require ICSF. In non-FIPS 140-2 mode, however, System SSL attempts to use ICSF's random number generation as it would when running in FIPS 140-2 mode. If ICSF or the required resource is unavailable, System SSL uses its own random number generation capabilities as in earlier releases.

Migration action: To run System SSL in FIPS 140-2 mode, you must now make sure that ICSF is running and that all user IDs that start SSL applications in FIPS 140-2 mode, invoke the GSKSRVR started task in FIPS 140-2 mode, or invoke the gskkyman utility to manage FIPS 140-2 key database files can access the necessary ICSF callable services.

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

DFS Migration Actions for z/OS V2R1

Migration Actions you can do NOW:

- **Determine whether to accept the new default values for certain zFS variables (Required-IF, as of V2R1)**
  - As of z/OS V2R1, IOEPRMxx parmlib member variables `meta_cache_size`, `metaback_cache_size`, `user_cache_size`, (for better performing caches) or `convert_auditfid` have new defaults.
    - If `meta_cache_size` and `metaback_cache_size` are not specified, it is typically 10% of the real storage available during zFS initialization*, ***
    - If `user_cache_size` is not specified, it is typically 10% of real storage available during zFS initialization**, ***
    - If `convert_auditfid` is not specified, the default is on. so that all files and directories in zFS file systems can be uniquely identified in SMF audit records.

  *See *zFS Admin* book for calculations when real is <64M, 64M-100M, 100M-(200G+100M), and >200G+100M.

  24   **See *zFS Admin* book for calculations when <256M, 256M-2G, and >2G.
  © 2015 IBM Corporation
  ***Use health check for specific calculations to help know what the default values will be pre-V2R1.

DFS Migration Actions for z/OS V2R1

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`

  25
  © 2015 IBM Corporation

DFS Migration Actions For z/OS V2R1

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

Determine whether to accept the new default values for certain zFS variables in the zFS IOEFSPRM configuration file **(Required-IF, as of V2R1)**

Required if you do not want the new default values to take effect or if you want to change existing values to the new default values.

Before z/OS V2R1, certain default values were used for the IOEFSPRM files or IOEPRMxx parmlib member variables meta_cache_size, metaback_cache_size, user_cache_size, or convert_auditfid.

Starting in z/OS V2R1, new default values are created for them. In z/OS V2R1, the zFS IOEFSPRM configuration file variable convert_auditfid default value was changed to ON so that all files and directories in zFS file systems can be uniquely identified in SMF audit records. The zFS IOEFSPRM configuration file variable user_cache_size default value will be changed to a value that is calculated based on the amount of real storage in the system. The zFS IOEFSPRM configuration file variables meta_cache_size and metaback_cache_size default values will be changed when both values are not specified to also be calculated based on the amount of real storage in the system. These are so that a system that has the capacity for more storage use and has sufficient space in the ZFS address space can have better performing caches.

If enough real storage exists, zFS can use more storage by default than it would have used in prior releases.

Migration action: Look for these:

- IOEFSPRM files or IOEPRMxx parmlib members that do not specify both meta_cache_size and metaback_cache_size options.
- IOEFSPRM files or IOEPRMxx parmlib members that do not specify the user_cache_size option.
- IOEFSPRM files or IOEPRMxx parmlib members that do not specify convert_auditfid settings.
- Programs that use **zfsadm format** commands where unique auditfids are not desired.
- JCL that contains calls to **ioeagfmt** that create aggregates for which unique auditfids are not desired.

- Programs that use zFS format API where unique auditfids are not desired.

Take these actions:

- For meta_cache_size, metaback_cache_size, or user_cache_size, if the old default values are desired, specify these values in your IOEFSPRM files or IOEPRMxx parmlib members.
- For auditfid, if you want the previous defaults, specify -nonewauditfid on calls to **ioeagfmt** or **zfsadm format** and convert_auditfid=OFF in your IOEFSPRM files or IOEPRMxx parmlib members.

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).

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.**

DFS Migration Actions Pre-First IPL

zFS: Ensure that the zFS kernel is active when using the batch utility ioeagfmt (Required-IF, as of V2R1)

Required if you do not run ioeagfmt after the kernel was initialized. However, it would be unusual not to have zFS active when running ioeagfmt.

Before z/OS V2R1, the batch utility **ioeagfmt** did not require that the zFS kernel be active. Starting in V2R1, **ioeagfmt** requires that the zFS kernel be active.

Migration action: Look for JCL or vendor applications that might be creating or submitting JCL that contain **ioeagfmt** calls that are run when the zFS kernel is not active. Run them when the zFS kernel is active.

If you are formatting file systems to be used by another system, you should be setting the -version parameter on **ioeagfmt** because the local zFS kernel would not have the information for a remote zFS system. You can also use the IOEFSUTL program to format aggregates. IOEFSUTL does not require the zFS kernel to be active to format a file system if you specify the -version parameter. But it will require the zFS kernel to be active if the -version parameter is left out.

JES2 Migration Actions for z/OS V2R1

Migration Actions you can do NOW:

- **Migrate to z11 mode (Recommended, as of R11, will be Required in z/OS V2R2)**
  - Determine your z11 checkpoint activation readiness with the $D ACTIVATE command. This command indicates if activation to z11 mode will succeed.
  - Run the JES2 $ACTIVATE command to activate z11 mode following the considerations for this command found in *z/OS JES2 Commands*.
  - You may fallback to z2 mode, if necessary.

- **Change JESJOBS profiles (Required – IF, as of V2R1)**
  - New JESJOBS resource names may match your existing JESJOBS profile names, thereby granting unintended user authority to modify jobs.
  - Update the profile, or create a new profile, if necessary.

27

© 2015 IBM Corporation

JES2 Migration Actions for z/OS V2R1

Migration Actions Pre-First IPL:

- **Review changes applicable to JES2 exits (Required – IF, as of V2R1)**

  - Review the z/OS JES2 Installation Exits book (section "JES2 exit migration considerations") for changes that may impact you.
  - Generally, assembly errors in JES2 exits indicate that you were affected by a JES2 data area change.
  - **Exit 6 has been known to have some customer changes required for z/OS V2R1.**
  - Other exits include: 7 and 8, 36 and 37, 44, and 59.

28

© 2015 IBM Corporation

JES2 Actions for z/OS V2R1

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

Migrate to z11 checkpoint level (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 checkpoint level. Activating z11 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 **$D ACTIVATE** command. This command indicates if activation to z11 level will succeed.
    2. Review your current utilization of BERT data to determine if there are sufficient BERTS, as detailed in "Check BERT utilization."
    3. If you issue the **$ACTIVATE,LEVEL=z11** command, activation of LARGEDS support is required.
    4. An additional *nnn* 4K records for CKPT1 is required for z11.
*   Run the **JES2 $ACTIVATE** command to verify non-configuration changes that must be accommodated before going to z11, and to activate z11 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. In z11 checkpoint level, 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 level (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 level, specify UNACT on PARM=. On a cold start JES2 starts in z11 checkpoint level 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 level.

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 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 checkpoint level 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. 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, 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 checkpoint level, see *z/OS Introduction and Release Guide*.
- For $ACTIVATE, $D ACTIVATE, $D CKPTSPACE and $JDHISTORY command details, see *z/OS JES2 Commands*.

Change JESJOBS profiles (Required-IF, as of V2R1)

Required if JESJOBS profiles could grant unintended authority to modify the job.

In z/OS V2R1, the new Job Modify SSI 85 defines additional resource names to the JESJOBS class. If any existing JESJOBS entity name matches one of the new resource names, it could result in unintended user authorization to modify job attributes. If necessary, update your JESJOBS profiles to avoid granting unintended user authority to modify jobs.

Migration action:

1. Search for JESJOBS profile names which match any of the new JESJOBS resource names:
    - SUBMIT.localnodeid.jobname.userid
    - HOLD.nodename.userid.jobname
    - RELEASE.nodename.userid.jobname
    - PURGE.nodename.userid.jobname
    - CANCEL.nodename.userid.jobname
    - START.nodename.userid.jobname
    - RESTART.nodename.userid.jobname
    - SPIN.nodename.userid.jobname
    - MODIFY.nodename.userid.jobname
    - REROUTE.nodename.userid.jobname
      For more information, see *z/OS JES2 Initialization and Tuning Guide*.
2. Ensure that your existing JESJOBS profiles grant the intended authority, given the new use of JESJOBS by the Job Modify SSI 85.
3. If any JESJOBS profile inadvertently allows user authority to Job Modify SSI 85 actions, update the profile or create a new profile, if necessary.

JES2 Migration Actions Pre-First IPL

Review changes applicable to JES2 exits (Required-IF, as of V2R1)

Required if you have any affected exits..

In z/OS V2R1, there may be exit changes necessary depending on what exits you are using, and what JES2 data areas those exits are referencing.

Migration action: Refer to the "JES2 exit migration consideration" section found in the *JES2 Installation Exits*. This section details the changed information which you may need to accommodate. The changes include (but are not limited to) Exit 6 considerations, with the macro $DTECNV containing fields which have changed.

Generally, assembly errors in JES2 exits are an indication that you are affected by a change.

Remove BRODCAST= from the OUTDEF initialization statement (Recommended, as of V2R1)

Not required, but recommended to keep your initialization deck clean of outdated and obsolete specifications.

In z/OS V2R1, JES2 ignores any specifications of the BRODCAST= parameter on OUTDEF. If a TSO user is not currently logged on, notify messages will always be issued locally (processing as with BRODCAST=YES) instead of being routed to the member that submitted the job (processing as with BRODCAST=NO). BRODCAST= never affected message processing when the TSO user is logged on. You can no longer use BRODCAST= to specify which member issues TSO notify messages when a user is not logged on.

Migration action:

1. Check the OUTDEF statement in the JES2 initialization deck for BRODCAST= specifications.

2. Remove BRODCAST= specifications from the OUTDEF statement. Any value you provide on BRODCAST= will be ignored. The external behavior will be to issue notify messages for a TSO user that is not logged on, on the member that requested the message. The message will be placed in the broadcast data set by TSO on that member.

Remove JCLERR= from the JOBDEF initialization statement (Recommended, as of V2R1)

Not required, but recommended to keep your initialization deck clean of outdated and obsolete specifications.

Before z/OS V2R1, for JCL job card errors that were detected during the input phase that the converter also detected, you could either have JES2 fail the job during input phase (JCLERR=YES) and the job never was sent to conversion, or you could have JES2 ignore the errors (JCLERR=NO) and send the job to the conversion phase where the converter could detect them. Starting with z/OS V2R1, input phase still detects errors, but jobs are always queued to the conversion phase and the input errors are added to those found by conversion and reported in the same way. The message id associated with the message indicates where the error occurred.

In z/OS V2R1, JES2 ignores any specifications of the JCLERR= parameter. If an error on the JOB card is encountered during the INPUT phase, the job is sent to the converter for INPUT phase error message processing. You can no longer use JCLERR=YES to specify for the job to be failed under these conditions.

Migration action:
1. Check the JOBDEF statement in the JES2 initialization deck for JCLERR= specifications.
2. Remove any JCLERR= specifications from the JOBDEF statement.

JES2 Migration Actions Post-First IPL

JES3 Migration Actions For z/OS V2R1

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

<none>

JES3 Migration Action Pre-First-IPL

Remove DUMP=JES and DUMP=MVS from the OPTIONS initialization statement (Recommended, as of V2R1)

Not required, but recommended to avoid warning messages. Any references to unsupported values are ignored, DUMP=PRDMP is used, a message is issued and initialization continues.

For enhanced problem determination and availability in z/OS V2R1, JES3 has dropped support for the DUMP=JES and DUMP=MVS parameter specifications on the OPTIONS initialization statement. In z/OS V2R1, DUMP=PRDMP is the default value and the only supported DUMP parameter specification.

Migration action:
1. Check the OPTIONS statement in the JES3 initialization deck for DUMP=JES and DUMP=MVS specifications.
2. Remove DUMP=JES and DUMP=MVS specifications from the OPTIONS statement, or change the values to DUMP=PRDMP.

Remove SDI from the OPTIONS initialization statement (Recommended, as of V2R1)

Not required, but recommended tavoid warning messages and command errors.

In z/OS V2R1, JES3 has removed the option to turn off Spool Data Integrity (SDI) functionality. If JES3 encounters the SDI keyword, command processing stops and a message is issued. If SDI is specified in the inish deck, the system issues the following message, and JES3 initialization continues: `IAT3256 SDI KEYWORD IGNORED, NO LONGER SUPPORTED`.

If SDI is used on a command (*F Q,SDI=), the system issues the following message, and the command is aborted:
```
IAT8050 SDI IS AN INVALID FIELD.
```
Migration action:
1. Remove the SDI keyword from the OPTIONS statement in the JES3 initialization deck.
2. Remove or update any automated instances of the *F Q and *I Q commands that specify the SDI parameter.

Change JES3 release level format (Required-IF, as of V2R1)

Required if code is dependent on the JES3 release level returned by SSI 54, SSI 82 or SSI 83.
In z/OS V2R1 SSI 54, SSI 82 and SSI 83 report the JES3 release level using the JES2 format z/OS 2.1. In previous versions of z/OS, JES3 reported the release level using the format z 1.13.0. The SSI fields which are affected by the format change are:

*   SSI 54: SSIVERS field from macro IEFSSVI
*   SSI 82: JPXVERSN field from macro IAZJPLEX
*   SSI 83: JPSYVERN field from macro IAZJPLXI.

Migration action:
1. Search for fields SSIVERS, JPXVERSN and JPSYVERN in invocations of SSI 54, SSI 82 and SSI 83.
2. For any code that requires the z/OS 2.1 JES3 release level, change the expected format to 'z/OS 2.1' (z/OS #.#). Ensure that the increase in ESQA usage caused by 24K per CPU used on the LPAR will not adversely affect your system and adjust values accordingly.

JES3 Migration Action Post-First-IPL

z/OS UNIX Migration Actions for z/OS V2R1

Migration Actions You Can Do Now:

Accommodate the new Shell and Utilities version of the zlsof utility (Required-IF, as of V2R1)

- As of V2R1, the z/OS UNIX Tools and Toy utility zlsof is incorporated. It differs from the Tools and Toy version in several ways. (zlsof lists users or processes using files or files systems, and lists files or file systems in use by users or processes.)
- If you use the Tools and Toy version, then decide if you want to continue to use it.
- z/OS' zlsof installs into /bin. Make sure that does not interfere if you wish to continue to keep your Tools and Toy zlsof.

Migration Actions Pre-First IPL:

Update applications that use SMF type 92 subtype 11 close records (Required-IF, as of V2R1)

- Before V2R1, SMF type 92, subtype 11 close records were written when sockets or char special records were closed. They have a very high close rate and do not provide useful information to determine how often files and directories are used.
- As of V2R1, they are written as subtype 16 close records instead, and will not appear in type 92 subtype 11.
- If you want to process this information, you need to include type 92 subtype 16 records in SMFPRMxx to collect them.

31   © 2015 IBM Corporation

---

z/OS UNIX Migration Actions for z/OS V2R1

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!

© 2015 IBM Corporation

z/OS UNIX Migration Actions for z/OS V2R1

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

Accommodate the new Shell and Utilities version of the zlsof utility (Required-IF, as of V2R1)

*Required if you currently use the Tools and Toys version of the **zlsof** utility.*
Before z/OS V2R1, the **zlsof** utility was obtained from the Tools and Toys section of the z/OS UNIX website. Starting in z/OS V2R1, Shell and Utilities support of the **zlsof** utility has been added. The supported version differs from the Tools and Toys version in a number of ways. For example, the new **zlsof** version includes support for displaying file lock holders and waiters when the byte range lock manager is used.
Migration action: Look for current use of the Tools and Toys version of **zlsof**. If there is no current use of the Tools and Toys version of **zlsof**, then no actions or changes are required.

If there is current usage of the Tools and Toys version of **zlsof**, determine if the command is 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. Note that **zlsof** can also reside in data sets where rexx execs can be run.
1.   If you want to preserve the Tools and Toys version, ensure that you save it into a directory that z/OS V2R1 will not install into. z/OS V2R1 provides **zlsof** in the /bin directory.
2.   If you do not want to preserve the Tools and Toys version and it is in /bin, then the installation of z/OS V2R1 automatically replaces the Tools and Toys version with the new officially supported version. If the Tools and Toys version is not in /bin, remove it from its current location.

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.

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. HFS is supported on z/OS V2.1. 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.*

The DDNAME() keyword of the BPXPRMxx ROOT and MOUNT statements is not supported by zFS. Use the FILESYSTEM(*name*) keyword instead. **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. 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. 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:
Before you begin the migration:
- Ensure that the following requirements have been met:
    - All systems in the sysplex are at least 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:
    o 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.
    o 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 applications that use SMF type 92 subtype 11 close records (Required-IF, as of V2R1)
Required if you want continue to get close records for sockets and character special files.
Before z/OS V2R1, SMF type 92, subtype 11 close records were written when sockets or character special records were closed. Sockets and character special files have very high close rates and do not provide useful information in determining how often files and directories are used. Starting in z/OS V2R1, they are written as subtype 16 close records instead. The sockets and character special file close records will no longer appear in type 92 subtype 11 records. Instead, you need to include type 92 subtype 16 records in the SMFPRMxx parmlib member to collect them.
Migration action:

1. Determine whether you have applications that use SMF type 92 subtype 11 close records. For those applications, SMF92TYP is set to SMF92#CLOSE (11) for subtype 11. SMF92CTY is set to FT_SOCKET (7) for sockets and FT_CHARSPEC (2) for character special files.
2. Change the application to look at subtype 16 records. SMF92TYP will be set to SMF92#CLSSOCCHARSPEC (16).

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.

Add to the maximum number of open files under z/OS UNIX Systems Services for PFA (Required-IF, as of R13)

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.
Note: If your installation has a BPXPRMxx MAXFILEPROC value greater than 5000, this action is not required.
Migration action: PFA requires the maximum number of open files under z/OS UNIX
System Services to be 5000 or greater. You can increase the value in one of two ways. The first option is preferable:
1. Update the FILEPROCMAX field in the OMVS segment of the PFA started task id, for example: TSO ALU pfauser OMVS(FILEPROCMAX(5000))
2. Update the BPXPRMxx MAXFILEPROC value in the parmlib member BPXPRMxx.

z/OS UNIX System Services Migration Actions Post-First IPL

Use the BPX.UNIQUE.USER profile instead of BPX.DEFAULT.USER (Required-IF, as of V2R1)

Required if you are using BPX.DEFAULT.USER. 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. As of z/OS V2R1 BPX.DEFAULT.USER has been removed.

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
```

RACF APAR OA42554 provides assistance with the conversion to BPX.UNIQUE.USER on z/OS V1R13 and z/OS V1R12. With this APAR you can model the user's home directory path by specifying &racuid in the model user's OMVS segment. Then, when the user's OMVS segment is automatically created, RACF will substitute the correct user ID. For more information on this capability, see the information in APAR OA42554.

Marna's "Big Migs" for Migrating to z/OS V2R1

Migration actions you should not overlook:

1. **RACF: Remove BPX.DEFAULT.USER (Required-IF, as of V2R1)**
2. **HTTP Server: Move from Domino to Apache (Recommended on V2R1, Required on z/OS V2R2.)**
3. **BCP: Convert your existing IBM Health Checker for z/OS set-up for automatic start-up (Required-IF, as of V2R1)**
4. **BCP: Set AUTHQLVL parameter in GRSCNFxx to recognize new GRS qnames (Required-IF, as of R13)**
5. **IP Services: Define a user ID for the system resolver with an associated OMVS segment (Required-IF, as of R13)**
6. **FIPS 140 Users: Must use ICSF (5 migration actions) (Required-IF, as of V2R1)**
7. **DFS: Determine whether to accept the new default values for certain zFS variables in the zFS IOEFSPRM configuration file (Required-IF, as of V2R1)**
8. **DFS: Accommodate new DASD space requirements for zFS file systems (Required, as of R13)**
9. **DFS: Ensure that sysplex=filesys is available on all systems *in shared zFS file system environment* (Required-IF, as of R13)**

Migrating to z/OS V2.1: Part 2 of 2 Summary

IBM

- **General:**
  - IFAPRDxx for V2, new address spaces, new and old data sets, changed Health Checks.
- **BCP:**
  - Move from Consoles Tracker to Generic Tracker, Increase in nucleus and ESQA, CON=DISTRIBUTED default, review AUTOR00 changes, LOADxx DYNCPADD default changes, SETLOAD xx,IEASYM and IEASYMU2.
- **Communications Server Migration Actions:**
  - ~~FTP in AUTHCMD and AUTHPGM IKJTSOxx,~~ IPv6 for IP fragment IDS and Policy-based routing change, VIPARANGE matching, VIT relocation.
- **DFSMS Migration Actions:**
  - Do not use IEBCOPYO, hsm VTOCCOPIES default change, replace path to enable ARC1809I messages.

35

© 2015 IBM Corporation

Migrating to z/OS V2.1: Part 2 of 2 Summary

- **JES2:** z11 mode, applicable JES2 exit changes, new JESJOBS profiles, OUTDEF and JOBDEF cleanup.

- **JES3:** DUMP= and SDI OPTIONS cleanup.

- **z/OS UNIX:** zlsof addition into z/OS, SMF 92 subtype 11 moving to subtype 16 for socket or char special records, changes for read-only root support, look at BPXPRMxx MOUNT SYSNAME statements.

36   © 2015 IBM Corporation