

The slide features a green header with the text "IBM System z" and "Technical University 2011". The main content area is white with the title "Migrating to z/OS R13 - Part 3 of 3" in green. Below the title is the code "zZS26" and the speaker's name "Marna WALLE" followed by her affiliation "IBM Systems and Technology Group", location "Poughkeepsie, New York USA", and email "mwalle@us.ibm.com". A green footer contains the copyright notice "©2011 IBM Corporation".

IBM System z
Technical University 2011

Migrating to z/OS R13 - Part 3 of 3

zZS26

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

©2011 IBM Corporation

Abstract: Migrating to z/OS R13 - Part 3 of 3

This is part three of a three-part session that will be of interest to System Programmers and their managers who are migrating to z/OS R13 from either z/OS R11 or z/OS R12. It is strongly recommended that you attend all three sessions for a complete migration picture. In part three, the speaker will continue to cover the specific migration actions for getting to z/OS R13. Selected elements such Distributed File Service, Infoprint Server, Language Environment, JES2, JES3, and z/OS UNIX System Services will be included. In addition, z/OS R13 installation changes and a couple of interesting z/OS R13 system programmer enhancements will also be covered.

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

Trademarks



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

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.
 Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

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

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

The following are trademarks or registered trademarks of other companies.

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

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

Notes:
 Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
 IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
 All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
 This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
 All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
 Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
 Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Migrating to z/OS R13 - Part 3 of 3 Agenda



• Migration Actions for z/OS R11 and R12:

- Distributed File Service
- JES2
- JES3
- Language Environment
- z/OSUNIX



- SMP/E and ServerPac Enhancements for Install
- Some Helpful z/OS R13 Sysprog Enhancements!
- Summary

Structure of *Migrating to z/OS R13* Presentations



• Session Part 1 of 3

- Content of z/OS R13: changing and withdrawn elements and functions
- Ordering and Deliverables
- z/OS Policies: coexistence support between z/OS R11, R12, and R13
- Planning for z/OS R13: system requirements for hardware and software
- Programmatic Verification of Migration Actions
- Positioning for sharing IEASYSxx, if desired

• Session Part 2 of 3

- Migrations actions for z/OS R13 and R12 from selected elements:
 - General
 - BCP
 - Communications Server
 - DFSMS
 - Infoprint Server
 - Cryptographic Services System SSL
 - TSO/E



• Session Part 3 of 3

- Content is what you just saw on the prior foil

4

© 2011 IBM Corporation

Definition of “Migration Action”:



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

5

 <p>Means “don’t overlook!”</p>		<p>Means some programmatic assistance is available</p>
---	---	---

Migration Definitions and Classifications

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

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

Migration Requirement Classification and Timing

The migration actions are classified as to their requirement status:

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

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

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

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

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

Icons used in this presentation:



means that you shouldn't overlook this migration action.



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

Elements with Migration Actions for z/OS R13



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

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

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

- | | |
|---|--|
| <ul style="list-style-type: none"> ➤ BCP ➤ Communications Server ➤ Crypto Services – ICSF, SSL, OCSF, and PKI ➤ DFSMS ▪ DFSORT ➤ Distributed File Service – zFS ➤ Infoprint Server ➤ JES2 ➤ JES3 | <ul style="list-style-type: none"> ➤ Language Environment <ul style="list-style-type: none"> ▪ RMF ▪ SDSF ▪ Security Server (RACF) ▪ SMP/E (covered in Part 1) ➤ TSO/E <ul style="list-style-type: none"> ▪ XL C/C++ ➤ z/OS UNIX |
|---|--|

- means that some of that element's migration actions will be discussed in these presentations
- means that you need to look for that element's migration actions in the *z/OS R13 Migration book*

6

© 2011 IBM Corporation

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

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

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

<http://www.ibm.com/systems/z/os/zos/installation/>.

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



Migration Actions you can do NOW:

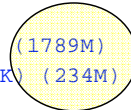
- **Verify virtual storage usage for zFS (Required, as of R11 APAR OA33451)**



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

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

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



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

7

© 2011 IBM Corporation

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



Migration Actions you can do NOW:

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



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

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

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

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

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

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

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

8

© 2011 IBM Corporation

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

Migration Actions You Can Do NOW:



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

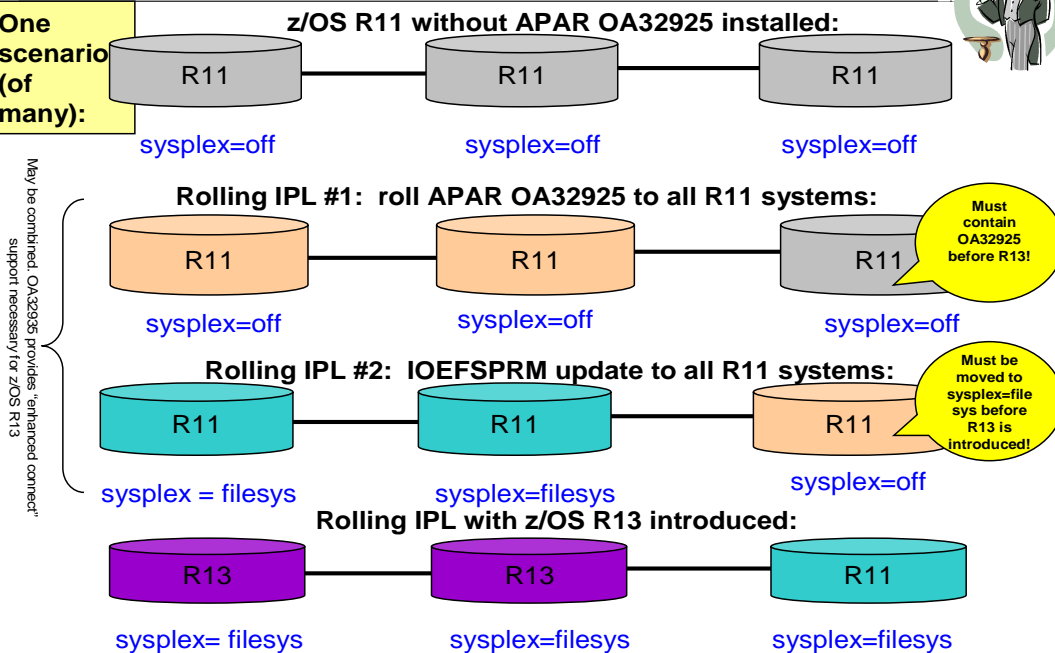


9

© 2011 IBM Corporation

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

One scenario (of many):



10

© 2011 IBM Corporation

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

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

DFS Migration Actions You Can Do Now



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

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

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

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

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

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

For all zFS file systems:

1. If you have not specified the zFS aggrgrow option in your IOEFSPRM configuration options file, recognize that the default is changing in z/OS V1R13 from aggrgrow=off to aggrgrow=on. This means that by default, a zFS read-write mounted file system that is mounted on z/OS V1R13 will attempt to dynamically extend when it runs out of space if a secondary allocation size is specified and there is space on the volume(s).
2. If you do not want that default change and you want it to act as in prior releases, specify aggrgrow=off in your IOEFSPRM configuration options file so that it takes effect on the next IPL. You can dynamically change the aggrgrow option to off with the **zfsadm config -aggrgrow off** command. You can see your current value for aggrgrow 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 <http://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 agrinfo** 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 V1R12 systems, you will receive error messages when you
try to bring up zFS on a z/OS V1R13 system.
```

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



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

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

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

IOEZ00662I: ZFS is low on storage.

Migration action:

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

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

Heap Usage By Component

```

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

```

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

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

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

DFS Migration Actions Pre-First IPL



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

Required, if you are using the DFS client.

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

Migration action: If your installation uses the DFS client, you must remove the following statement from the BPXPRMxx 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.

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





Migration Actions Pre-First IPL:

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

Migration Actions Post-First IPL:

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



11

© 2011 IBM Corporation

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

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

JES2 Migration Actions You Can Do Now

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

Required if installation exits use PDBLENG equate.

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

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

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

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

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

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

JES2 Migration Actions Pre-First IPL

<none>

JES2 Migration Actions Post-First IPL



Migrate to z11 mode (Recommended, as of R11)

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

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

Migration action:

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

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

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

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

Check BERT utilization

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

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

```

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

In the example, there are 108 JQEs that have a total of 211 BERTs associated with them. This example is for a system in z2 mode and does not have any BERTs associated with JOEs.

- The \$D ACTIVATE command displays the number of BERTs that are needed for activation to z11 mode. This is the number of BERTs that will be associated with JOEs after the \$ACTIVATE. The example below shows the output of the \$D ACTIVATE command.

```

$HASP895 $DACTIVATE
    
```

```

$HASP895 JES2 CHECKPOINT MODE IS CURRENTLY Z2
$HASP895 THE CURRENT CHECKPOINT:
$HASP895 -- CONTAINS 1100 BERTS AND BERT UTILIZATION IS 30
$HASP895 PERCENT.
$HASP895 -- CONTAINS 158 4K RECORDS.
$HASP895 z11 CHECKPOINT MODE ACTIVATION WILL:
$HASP895 -- EXPAND CHECKPOINT SIZE TO 165 4K RECORDS.
$HASP895 -- REQUIRE 22 ADDITIONAL BERTS AND UTILIZATION
$HASP895 WOULD REACH 32 PERCENT.
$HASP895 z11 ACTIVATION WILL SUCCEED IF ISSUED FROM THIS MEMBER.
  
```

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

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

```

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

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

```

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

Reference information:

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

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



Migration Actions Pre-First IPL:

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

- Before R13, an IXC102A DOWN response (**XCF IS WAITING FOR SYSTEM *sysname* DEACTIVATION. REPLY DOWN WHEN MVS ON *sysname* HAS BEEN SYSTEM RESET**) required operations to enter the *S main,FLUSH command. Without that command, jobs on the local were held up until the local processor was reconnected.
- As of R13, JES3 automatically flushes the active jobs on the local processor as soon as the operator responds to IXC102A. This reduces the time gap between the local processor being removed from the sysplex, and job recovery actions.
- If you still enter the *S main,FLUSH command, there will be no effect because affected jobs will have already been flushed by the new automatic processing.

12

© 2011 IBM Corporation

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

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

JES3 Migration Actions You Can Do Now

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

Required if your installation has a code dependency on the format of messages in the JES3 DLOG.

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

```
10250 1143599 SY1 R= XXX33913 ICH70001I IBMUSER LAST ACCESS AT 11:39:46 ON MONDAY, NOVEMBER 1, 2010
10250 1144001 SY1 R= XXX33913 IEF403I XXX33913 - STARTED - TIME=11.44.00
MLG 10250 1144004 &SY1 R= XXX33913 +XXX100I 9012345678901234567890123456
MLG 10250 1144004 &SY1 R= XXX339137890123456789012345678901234567890123456789012345678901234567890123456
10250 1144004 SY1 R= XXX33913 IEF404I XXX33913 - ENDED - TIME=11.44.00
```

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

```
10305 1000486 SY1 R= XXX33913 ICH70001I IBMUSER LAST ACCESS AT 09:57:46 ON MONDAY, NOVEMBER 1, 2010
10305 1000488 SY1 R= XXX33913 IEF403I XXX33913 - STARTED - TIME=10.00.48
MLG 10305 1000494 &SY1 R= XXX33913 +XXX100I 9012345678901234567890123456
MLG 10305 1000494 &SY1 R= XXX33913 78901234567890123456789012345678901234567890123456789012345678901234567890123456
10305 1000494 SY1 R= XXX33913 IEF404I XXX33913 - ENDED - TIME=10.00.49
```

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

JES3 Migration Actions Pre-First IPL

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

*Not required, but recommended to avoid redundant *S main,FLUSH commands.*

Prior to z/OS V1R13, a DOWN response to message IXC102A issued for a JES3 local processor required operations to enter the ***S main,FLUSH** command. Without the ***S main,FLUSH** command, jobs on the local were held up until the local processor was reconnected.

Starting in z/OS V1R13, JES3 flushes the active jobs on the local processor automatically as soon as the operator responds to message IXC102A. This automatic flush eliminates the step of issuing the command and reduces the time gap between the local processor being removed from the sysplex and job recovery actions.

In z/OS V1R13 and later releases, if you run the ***S main,FLUSH** command in response to the XCF messages, the command will have no effect because the affected jobs will have already been flushed by the new automatic processing.

Migration action: If you want to avoid redundant ***S main,FLUSH** commands, remove the ***S main,FLUSH** command from all automated procedures or operating procedures.

Modify code that uses DATLOREC and DATINPTR (IATYDAT) as a programming interface (Required-IF, as of APAR OA34642)

Required if you use DATLOREC and DATINPTR (IATYDAT) as a programming interface.

When a job's JCL contains in-stream data sets, the in-stream data is stored in multi-record files apart from the JESJCLIN file. These files are located by SYSIN pointer records that are written into a job's JESJCLIN file by input service. Prior to z/OS V1R12 (without APAR OA34642), SYSIN pointer records were included in the JESJCLIN file's record count which is saved in DATLOREC (IATYDAT). In z/OS V1R12 z/OS V1R11, and z/OS V1R10 (with APAR OA34642), the SYSIN pointer records are no longer included in the record count for JESJCLIN files. A new flag, DATCTLRD (IATYDAT), will be on for any record, including SYSIN pointer records, that are not included in a file's record count.

Migration action: Before applying the PTF for APAR OA34642, examine and modify any code that uses DATLOREC and DATINPTR (IATYDAT) as a programming interface. Note that IATYDAT is an internal JES3 control block that resides on spool and that this change affects only JESJCLIN data sets.

JES3 Migration Actions Post-First IPL

<none>

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



Non-existent Migration Actions ☺ :

- Determine the impact of added and changed runtime options
 - In R12: none.
 - In R13: none.

Migration Actions You Can Do Now:

- Convert to CEEPRMxx to set system-level default runtime options **(Recommended, will be Required after R13)**
 - Move from assembler usermods to CEEPRMxx.

Migration Actions Pre-First IPL:

- Examine programs that read output when a D CEE command is issued **(Required-IF, as of R12)**

• Command response is changed: only suboptions specified with a parmlib member or SETCEE command (not those not explicitly set, as before) are displayed. Commas are now displayed for suboptions not explicitly set.

SETCEE CEEDOPT,ALL31,ANYHEAP(4K),FILETAG(,AUTOTAG)

a subsequent D CEE,CEEDOPT command displays the following:

```
CEE3745I 09.32.13 DISPLAY CEEDOPT
NO MEMBERS SPECIFIED
LAST WHERE SET OPTION
```

```
-----
SETCEE command ALL31()
SETCEE command ANYHEAP(4096,,,)
SETCEE command FILETAG(,AUTOTAG)
```

13

© 2011 IBM Corporation

Language Environment Migration Actions Between z/OS V1 R11 and z/OS V1 R13

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

Language Environment Migration Actions You Can Do Now

<none>

Language Environment Migration Actions Pre-First IPL

Examine programs that read output when a D CEE command is issued (Required-IF, as of R12)

Required if you have an application that reads the output of a D CEE command.

Starting in z/OS V1R12, changes are made to the Language Environment runtime options report when a D CEE command is issued. Before z/OS V1R12, the Language Environment runtime options report displayed all suboptions, even if they were not explicitly set for any runtime option that was specified. Starting in z/OS V1R12, when a valid option is specified with a parmlib member or a SETCEE command, only the suboptions specified are displayed when a D CEE command is issued. A comma is displayed as a placeholder for those suboptions not specified.

Migration action: Examine any programs that read the output of a D CEE command to ensure compatibility with the updated runtime options report. Commas are now displayed for any suboptions that are not explicitly specified in a parmlib member or with a SETCEE command.

For example, if the following SETCEE command is issued:

```
SETCEE CEEDOPT,ALL31,ANYHEAP(4K),FILETAG(,AUTOTAG)
```

a subsequent D CEE,CEEDOPT command displays the following:

```
CEE3745I 09.32.13 DISPLAY CEEDOPT
NO MEMBERS SPECIFIED
LAST WHERE SET OPTION
```

```
-----
SETCEE command ALL31()
SETCEE command ANYHEAP(4096,,,)

```

SETCEE command FILETAG(,AUTOTAG)

Set runtime options as overrideable or nonoverrideable in CEEPRMxx parmlib member (Recommended, as of R12)

Not required, but recommended so you can eliminate use of the assembler language USERMODs to specify installation-wide runtime options, and use parmlib member CEEPRMxx instead.

Before z/OS V1R12, all runtime options specified in a CEEPRMxx parmlib member were overrideable by default. Beginning with z/OS V1R12, you can set runtime options as overrideable or nonoverrideable in the CEEPRMxx parmlib member or with a SETCEE command using the OVR or NONOVR attribute. The ability to specify an option as overrideable or nonoverrideable removes a barrier to using CEEPRMxx.

Migration action: Set runtime options as overrideable or nonoverrideable in the CEEPRMxx parmlib member or by issuing the SETCEE command using the OVR or NONOVR attribute.

Now that runtime options can be specified as overrideable or nonoverrideable in a CEEPRMxx parmlib member, and with a SETCEE command, you can eliminate the use of assembler language USERMODs to specify installation-wide runtime options.

Determine the impact of added and changed run-time options

Periodically, Language Environment introduces new runtime options, adds new suboptions to existing runtime options, and changes the defaults of runtime options. For z/OS R12 and R13, no runtime options were changed or new suboptions added. Therefore, no migration action is required for your migration to z/OS R13.

Language Environment Migration Actions Post-First IPL

Examine programs that read output from a CICS CLER transaction (Required-IF, as of R12)

Required if you have an application that reads the output of a CICS CLER transaction.

Starting with z/OS V1R12, the Language Environment runtime options report displayed from the CICS CLER transaction is changed. The report is modified to have a wider LAST WHERE SET column to accommodate longer values, such as "Installation Non-overrideable." In addition, the report heading OPTIONS is changed to OPTION to match the other runtime options reports.

Migration action: Examine programs that read the output of a CICS CLER transaction to ensure compatibility with the updated CLER runtime options report. The LAST WHERE SET column is now wider and the OPTIONS heading is changed to OPTION. The following is a subset of the new report to show the formatting changes:

LAST WHERE SET	OPTION
Installation default	ABPERC(NONE)
Installation default	ABTERMENC(ABEND)
Installation default	NOAIXBLD

Use Unicode Services to create conversion tables (Required-IF, as of R12)

Required if you use the iconv() family of functions to test to a "known conversion result" and experience testcase failures. Also, if you use custom conversion tables replacing those listed in either ucmapt.lst or genxlt.lst.

Beginning with z/OS V1R12, the C/C++ runtime library will no longer include any ucmapi source code or genxlt source code for character conversions now being performed by Unicode Services.

Migration action:

- If you use customized conversion tables, you should now generate custom Unicode Services conversion tables.
- If you use the iconv() family of functions testing to a "known conversion result" and experience test case failures, you need to update your expected results to the new conversion results.
- If you want to create custom conversion tables involving any of the CCSIDs related to the conversion table source no longer being shipped, you should now generate custom Unicode Services conversion tables instead of custom Language Environment conversion tables.

The *installation prefix*.SCEEUMAP data set will no longer be shipped. The /usr/lib/nls/locale/ucmap HFS directory will no longer be shipped.

Note: The _ICONV_TECHNIQUE environment variable must be set to the same technique search order value used for the customized Unicode Services table in order for the iconv() family of functions to use the customized Unicode Services table. For example, if you want the iconv() family of functions to use a user-defined Unicode Services table with a technique search order of 2, the _ICONV_TECHNIQUE environment variable should be set to 2LMREC.

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



Migration Actions You Can Do Now:

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

14

© 2011 IBM Corporation

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



Migration Actions Pre-First IPL:


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



© 2011 IBM Corporation

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

Migration Actions Pre-First IPL:


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

16

© 2011 IBM Corporation

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

Migration Actions Post-First IPL:

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

17

© 2011 IBM Corporation

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

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

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

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

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

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

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

1. If you want to keep the `-o fsoptions` option, determine the type of the file system and specify it, using the `-t` option.
2. If the file system is zFS, verify that the options string that was specified in `-o fsoptions` is valid.

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

Required because invocations of /usr/sbin/unmount might fail if the path name is not the name of the mount point.

Before z/OS V1R13, if a path name was specified when using the `/usr/sbin/unmount` command to unmount a file system, the path name could be (1) any file or directory in the file system or (2) the mount point of the file system. Starting in z/OS V1R13, when the `/usr/sbin/unmount` command is issued, only the mount point can be specified for the path name. Unmounting based on using any file or directory in a file system can sometimes cause the wrong file system from being unmounted. There can be instances, for example, where the file system has already been unmounted and a subsequent unmount on the same path name would result in accidentally unmounting the wrong file system.

Migration action: If you have invocations of the `/usr/sbin/unmount` command that do not specify a mount point on the `name...` parameter, follow these steps:

1. Look for instances where the specified path names in `/usr/sbin/unmount` invocations are files or directories.
2. Select one of the following actions:
 - a. Use the `-m` option in `/usr/sbin/unmount` command if you have confirmed that something other than mount point needs to be specified. Note the file system that contains the file or the directory will be unmounted.
 - b. Change the invocation so that only mount points are specified for the path name.

Review programs that invoke the BPX1EXM/BPX4EXM callable service (Required-IF, as of R12, and prior releases with OA30897)

Required if you have a program that fits the criteria described below.

Before z/OS V1R12 (and without the PTFs for APAR OA30897 installed on z/OS V1R11, z/OS V1R10, and z/OS V1R9), an unauthorized caller could pass a parameter greater than 100 characters on the BPX1EXM/BPX4EXM (execmvs) syscall. Beginning with z/OS V1R12 (and with the PTFs for APAR OA30897 installed to z/OS V1R11, z/OS V1R10, and z/OS V1R9), EC6 abends with reason code 0B26C048 will occur if: (1) the caller is not authorized, (2) the parameter is greater than 100 characters, and (3) the program would be executed in an authorized state.

Message BPXP027I goes to the system hardcopy log. See message BPXP027I for the program name and the argument length:

```
BPXP027I JOBNAME jobname, ATTEMPTED TO ISSUE AN EXEC OF THE APF AUTHORIZED MVS
PROGRAM, pgmname, WITH A PARM LENGTH OF xxx
```

Migration action: Review BPX1EXM/BPX4EXM calls from unauthorized callers that might call an APF-authorized program with an argument parameter greater than 100 characters.

- If the program can handle an argument greater than 100 characters, define a profile for the FACILITY class resource name `BPX.EXECMVSAPF.program_name`.

- **Note:** Individual users do not need to be given access to the profile; the profile just needs to be defined. Also, the *program_name* defines the target APF-authorized program. .
- If the program cannot handle an argument greater than 100 characters, change the BPX1EXM/BPX4EXM calls to pass an argument that is 100 characters or less.

Accommodate new Shell and Utilities version of the tsocmd command (Required-IF, as of R12)

Required if you currently use the Tools and Toys version of the tsocmd command.

Before z/OS V1R12, the **tsocmd** command was obtained from the Tools and Toys section of the z/OS UNIX Web site. Starting with z/OS V1R12, Shell and Utilities support of the **tsocmd** command has been added. The supported version differs from the Tools and Toys version in a number of ways. For example:

- Exit values are consistently sent if the issued TSO/E command fails.
- The tsoin and tsoout environment variables are not supported. Instead, stdin and stdout are supported, as is done for most other Shell and Utilities commands.
- The TSOPROFILE environment variable is supported.

Migration action: Look for current use of the Tools and Toys version of **tsocmd**. If there is no current use of the Tools and Toys version of **tsocmd**, then no actions or changes are required.

If there is current use of the Tools and Toys version of **tsocmd**, determine if the command is located in /bin or in another directory. Also, determine if you want to preserve the Tools and Toys version in addition to the officially shipped version.

1. If you want to preserve the Tools and Toys version, ensure the Tools and Toys version of **tsocmd** is not located in /bin prior to the installation of z/OS V1R13.
2. If you do not want to preserve the Tools and Toys version, and it is located in /bin, then the installation of z/OS V1R13 will automatically replace the Tools and Toys version with the new officially supported version. If the Tools and Toys version is not located in /bin, remove it from its current location. In either case, you will also need to remove the Tools and Toys **tsocmd** load module from the authorized load library (either prefix.TSOCMD.LOADLIB or SYS1.LINKLIB) as described in the **tsocmd** Tools and Toys README documentation available at <ftp://ftp.software.ibm.com/s390/zos/tools/tsocmd/tsocmd.readme.txt>.

Note: A new version of **tsocmd** that matches the version shipped with z/OS V1R12 will be made available on the Tools and Toys Web site. If you have multiple systems at different releases and you want to have the same version of the tool on all releases, or if you want to try out the new command prior to z/OS V1R12, you can download the new version to the earlier systems and replace the previous Tools and Toys version of **tsocmd** if appropriate. If this action is taken, you will need to clean up the authorized load library as described in Step 2 prior to downloading the new tool.

Remove MAXSOCKETS values from AF_UNIX in the BPXPRMxx parmlib member (Recommended, as of R12)

Not required, but recommended to avoid confusion if users do not know that a value does not need to be specified for the MAXSOCKETS keyword on the NETWORK statement for AF_UNIX for z/OS V1R12.

Before z/OS V1R12, a value had to be specified for the MAXSOCKETS keyword in the NETWORK statement for AF_UNIX in the BPXPRMxx parmlib member of SYS1.PARMLIB if the maximum number of AF_UNIX sockets for the system needed to be greater than the default of 100. As of z/OS V1R12, the value does not need to be specified because a maximum value of 10,000 has been set for MAXSOCKETS for AF_UNIX. The MAXSOCKETS keyword is still allowed on the NETWORK statement for AF_UNIX, but will be ignored on a z/OS V1R13 system.

If you share your BPXPRMxx parmlib member with lower level systems, only remove your MAXSOCKETS statement from your AF_UNIX domain if you will accept the default MAXSOCKETS value for AF_UNIX for those lower level systems. (The default on z/OS V1R11 and z/OS V1R10 for MAXSOCKETS in the AF_UNIX domain is 100.) Otherwise, remove the MAXSOCKETS statement from the AF_UNIX domain when all systems that share the BPXPRMxx parmlib member are at z/OS V1R12 or later.

Migration action: Remove any MAXSOCKETS statement from NETWORK statements for DOMAINNAME(AF_UNIX). In a configuration with a shared BPXPRMxx parmlib, MAXSOCKETS should only be removed when all systems are at z/OS V1R12 or higher.



Migrate from HFS file systems to zFS file systems (Recommended)

Not required, recommended because the action is planned to become a requirement in a future release. zFS is the strategic file system for z/OS UNIX and continues to be enhanced to provide superior performance, reliability, and data integrity. IBM has not announced removal of support for the HFS file system.

Before z/OS V1R7, the HFS file system was the primary hierarchical file system. As of z/OS V1R7, you can use any combination of HFS and zFS file systems. Because zFS has higher performance characteristics than HFS and is the strategic file system, you should migrate your HFS file systems to zFS.

The HFS and zFS file system types in mount statements and command operands are now generic file system types that can mean either HFS or zFS. Based on the data set type, the system will determine which is appropriate. But note that you must still specify a type (HFS or zFS and it cannot be defaulted), and if the type you specify is not correct for the file system being mounted, any associated parameter string setting in the mount statement or command is ignored, even though the system sets the type correctly and processes the mount.

Understand the zFS recommendations and limits. For more information, see Minimum and maximum file system sizes in *z/OS Distributed File Service zSeries File System Administration*

Migration action:

1. Before beginning the migration, do the following:

- Establish backout procedures.
- Decide on naming conventions.
- Decide on unavailability.
- Understand any cloning or deployment changes required by zFS systems being linear data sets. Considerations would include any copy utility invocations, BPXPRMxx specifications for symbolics, and placement of zFS file systems on system volumes.

2. Perform the conversion from an HFS to zFS file system.

Tip: Use the BPXWH2Z tool to perform the conversion. It is an ISPF-based tool that migrates HFS file systems to zFS file systems. Using its panel interface, you can alter the space allocation, placement, SMS classes, and data set names. A HELP panel is provided. With this tool, you can:

- Migrate HFS file systems (both mounted and unmounted) to zFS file systems. If the HFS being migrated is mounted, the tool automatically unmounts it and then mounts the new zFS file system on its current mount point.
- Define zFS aggregates by default to be approximately the same size as the HFS. The new allocation size can also be increased or decreased.
- Have the migration run in TSO foreground or UNIX background.

You can use the JCL sample ISPBTC in SYS1.SAMPLIB to invoke BPXWH2Z as an ISPF batch job. Before you run the job, be sure to read the Notes section. When you run BPXWH2Z on your z/OS V1R13 system, it uses the z/OS V1R13 level of the **pax** command. This level was enhanced in z/OS V1R7 for sparse file support and other characteristics that are of concern when migrating from an HFS to zFS file system. You can manually migrate from an HFS to zFS file system without using the tool. However, you would need to allocate and format the target zFS file systems.

Requirement: The BPXWH2Z tool requires the zFS address space to be operational. Therefore, before attempting to migrate existing HFS to zFS file systems using BPXWH2Z, make sure that the zFS address space has been successfully configured and initialized.

Tip: Consider migrating the sysplex root file system to zFS after you IPL z/OS V1R13. You can dynamically migrate the HFS sysplex root in a shared file system configuration to zFS while the root is in use, without disrupting workloads. Before z/OS V1R10, you could not migrate without disrupting active workloads. Although the shared file system configuration is required, the sysplex can be a single system. For the steps involved in migrating the sysplex root file system from HFS to zFS, see below.

Tip: Use check USS_HFS_DETECTED to verify all file systems mounted. This check issues exception message BPXH068E if any HFS file systems are found.

In z/OS R12! DFSMSDfp will allow a zFS data set to be recataloged with an indirect volume serial or system symbol. This is designed to allow the zFS file systems used for z/OS system software files (called version root file systems) to be cataloged using an indirect volume serial or a system symbol the same way as non-VSAM data sets to make cloning and migration easier.

3. Change policies and scripts, and so forth, to reflect the change from the HFS file system to zFS file system.

Tip: Use the RMF Monitor III option to report on zFS activity.

Migrating the sysplex root file system from HFS to zFS after IPLing z/OS V1R13:

Before you begin the migration:

- Ensure that the following requirements have been met:
 - All systems in the sysplex are at the V1R13 level.

- The current sysplex root file system PFS, and the new sysplex root file system PFS, are up in all the systems in shared file system configuration.
- Be aware of the following restrictions:
 - The current sysplex root file system must be mounted as a read-only file system.
 - The systems that do not meet the requirements for this migration action cannot join the sysplex during the sysplex root file system migration processing, but they can join the sysplex after the sysplex root migration has completed.
 - The current sysplex root and the new sysplex root must be either HFS or zFS in any combination. If the new sysplex root is zFS, then it must be HFS-compatible.
 - The sysplex root or any directories on it cannot have been exported by the DFS or SMB server.
- Note the following:
 - Remote NFS mounts of the sysplex root or any directories on it are considered active use of the current sysplex root file system.
 - During the migration, the new zFS sysplex root file system must not be HSM-migrated, mounted, or in use.
 - Mount parameters are preserved during the migration or replacement of the sysplex root file system of the same file system type (PFS). They are dropped if the file system type is different.
 - Directories, data, files, and links are not copied from one file system to another.

Perform the migration as follows:

1. Ensure that a file system has been mounted read-only as the current sysplex root file system. When the root is mounted read-only, there are no function-shipping clients as long as physical paths to the DASD are available to each system. To verify that there are no function-shipping clients, issue: `D OMVS,F,NAME=root_file_system_name`. You should see `CLIENT=N` on each system.

2. Allocate and set up the new zFS sysplex root file system:

- Create a new zFS file system to be used as the new sysplex root file system. *z/OS Distributed File Service zSeries File System Administration* discusses creating and managing zFS file systems.

Rules:

- The UID, GID and the permission bits of the root directory in the new sysplex root file system must be same as the root directory in the current sysplex root file system.
- If the SECLABEL class is active and the MLFSOBJ option is active, the security label for the new zFS file system must match the assumed security label of the current sysplex root file system.
- On the new sysplex root file system, set up the active mount points and the symbolic links. The mount points and symbolic links must be the same as the ones on the current sysplex root file system. You can set them up either (1) manually or (2) by using the **pax** shell command to populate the new sysplex root file system using the existing sysplex root as a source. To do it manually, create a mount point in the existing sysplex root (for example, `/newroot`) and mount the new sysplex root file system in the `MODE(RDWR)` on that mount point. After mounting the new sysplex root file system, manually issue `MKDIRs` and `ln -s` to create the mount point directories and symbolic links similar to the existing sysplex root file system. Note that the new sysplex root file system must contain all active mount points and symbolic links exactly as on the existing sysplex root file system.
- Use the **pax** shell command to populate the new file system, using the existing sysplex root as a source.

Example:

```
cd /
pax -wr -pe -XCM ./ /newroot
```

For more information about using **pax** to copy data from an HFS file system to a zFS file system, see *z/OS Distributed File Service zSeries File System Administration*.

- Unmount the new zFS file system.

3. Before proceeding with this step, ensure PTF UA49782 for APAR OA29567 is installed on your z/OS V1R10 systems. Then, on any system in the shared file system configuration, issue:

`F OMVS,NEWROOT=new.root.file.system.name,COND=<Yes|No>`

YES Proceed conditionally. The system checks for active usage in the current sysplex root file system and reports the active usage in a `BPXF245I` message. If file activity is found, the command fails with `EBUSY` return code and `JrActivityFound` reason code. If file activity is not found, the command continues processing to replace the sysplex root. **YES** is the default.

NO Proceed unconditionally. The system checks for active usage in the current sysplex root file system and reports the active usage in a BPXF245I message. Replacement of the sysplex root file system will continue.

The migration of the sysplex root file system will begin. During the migration, active connections to files and directories in the current sysplex root file system are broken.

After the migration completes:

- The root CWD('/') is updated on all systems in the sysplex to point to the new sysplex root file system.
- New opens go to the new sysplex root file system. The current sysplex root for the root directory is replaced for all processes in all systems. The current directory for root directory is replaced for any processes using it.
- Old connections in the previous sysplex root file system might get EIO errors.

4. Update the TYPE parameter and name of the sysplex root file system in the BPXPRMxx member of SYS1.PARMLIB.

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

Update invocations of MOUNT statements in the BPXPRMxx parmlib member (Required-IF, as of R13)

Required because required file systems might not be mounted.

Before z/OS V1R13, if the MOUNT statement in the BPXPRMxx 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 BPXPRMxx 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 BPXPRMxx 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 BPXPRMxx member is also specified as a z/OS UNIX parmlib member for the specified target
2. ~~Move~~ Move the MOUNT statements from that BPXPRMxx member to a BPXPRMxx 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 (*rexex-variable*) format of passing variables and do not check RETVAL and or ERRNO upon return, follow these steps.

1. Run these programs.
2. Fix parsing errors as they occur, replacing invalid REXX variables with valid ones. The error number will identify the invalid REXX variable. Note that REXX variable names cannot start with a digit or include arithmetic operators. For example, x*y is an invalid REXX variable.
 - a. If the program produces a REXX parsing error, the error will be: IRX0041I Error running
..... line xx: This is a TSO REXX error message.
 - b. If the program produces a z/OS UNIX syscall parsing error, the error code will be -21,-22,..., as described under the section "Returned from the SYSCALL environment" in the USS REXX documentation.

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

Consider skulker invocations due to updated restriction (Required-IF, as of R12)

Required if you are concerned about not getting a warning message if objects in the specified directory or subdirectories whose path name includes single quotes or a newline character are found.

Before z/OS V1R12, if the **skulker** command was issued and there were objects in the specified directory (and in any subdirectories if the **-r** option was used) whose path name included single quotes or a newline character, those objects were not deleted, a warning message was issued, and all processing stopped. Starting with z/OS V1R12, a warning message is not issued, those objects are ignored, and processing continues.

Migration action: Use the **find** command to determine if there are any objects in the specified directory or subdirectories whose path name includes single quotes or a newline character. Do not use the **find** command to locate the objects whose path name include single quotes or a newline character as a standard practice before using the **skulker** command if you want the directories removed because issuing the **find** command will modify the access times of directories that are part of that **find** request.



Use the BPX.UNIQUE.USER profile instead of BPX.DEFAULT.USER (Recommended, as of R11)

Recommended because in the release following z/OS V1R13, support for the BPX.DEFAULT.USER profile is planned to be removed. The use of BPX.UNIQUE.USER increases security because shared IDs are not used.

Before z/OS V1R11, if the BPX.DEFAULT.USER profile in the FACILITY class was defined, users who accessed z/OS UNIX services who did not have an OMVS user or group segment were assigned the default OMVS segments for the length of the user session. All users of the default OMVS segments shared the same UID and GID. As of z/OS V1R11, if BPX.UNIQUE.USER has been defined, users who access z/OS UNIX services who do not have an OMVS user or group segment are automatically assigned an OMVS segment with a unique UID and GID. The new OMVS segments are added to the user and group profiles in the RACF database.

Migration action: Follow the steps in *z/OS UNIX System Services Planning* to set up the BPX.UNIQUE.USER profile. If BPX.DEFAULT.USER has not been deleted, BPX.UNIQUE.USER takes precedence when default OMVS segments are used.

To remove the BPX.DEFAULT.USER profile, use the following RACF commands:

```
RDELETE FACILITY BPX.DEFAULT.USER
SETROPTS RACLIST(FACILITY) REFRESH
```

Marna's "Big Migs" for z/OS R13 from z/OS R11



Migration Actions You Should NOT Overlook from Part 3:

1. Verify virtual storage usage for zFS (Required, as of R11 APAR OA33451)
2. Accommodate new DASD space requirements for zFS file systems (Required, as of R13)
3. Ensure that sysplex=filesys is available on all systems in shared zFS file system environment (Required-IF, as of R13)
4. Update invocations of MOUNT statements in BPXPRMxx (Required-IF, as of R13)



Migration Actions You Should NOT Overlook from Part 2:

5. Find replacements for removed functions and elements (Required)
6. Review the list of WTORs in parmlib member AUTOR00 (Required, as of R12)
7. Set AUTHQLVL parameter in GRSCNFxx to recognize new GRS qnames (Required IF, as of R13)
8. IP Services: Define a user ID for the system resolver with an associated OMVS segment (Required IF, as of R13)
9. Increase the space in the Printer Inventory file system (Required, as of R12)
10. Remove V2 Printer Inventory files at fallback (Recomm, as of R12)



SMP/E V3R6 (non-z/OSMF) Enhancements



- **Enhancements not related to z/OSMF:**

- **Cross Global Zone Reporting**

- REPORT CROSSZONE and REPORT SYSMODS allow zones to be specified that are in different GLOBALs. This allows better identification of missing conditional requisites and SYSMODS between two zones!
 - REPORT CROSSZONE **ZONES((SYS1.ZOS12.CSI,ZOS12T) , (SYS1.ZOS13.CSI,ZOS13T)) .**
 - REPORT SYSMODS INZONES (ZOS12T)
COMPAREDTO(**SYS1.ZOS13.CSI.ZOS13T**) .

- **SYSMOD Comparison HOLDDATA Reporting on REPORT SYSMODS command**

- A new report (**SYSMOD Comparison HOLDDATA**) lists the SYSTEM and USER HOLDDATA that must be resolved to install the identified missing SYSMODs. HOLDDATA will now no longer be deleted on ACCEPT when NOPURGE is off, and during RESTORE when NOREJECT is off.

```

SYSMOD COMPARISON HOLDDATA REPORT FOR TARGET ZONE ....
TYPE REASON ID FMID SYSMOD ++HOLD DATA
-----
SYSTEM ACTION HBB7770 UA58677 ++ HOLD(UA58677) SYS FMID(HBB7770) REASON(ACT ...
                                     COMMENT
                                     (*****
    
```

19

© 2011 IBM Corporation

New ServerPac Install Enhancements for z/OS R13



- **ServerPac in z/OS R13:**

- Improved handling for **user-supplied installation jobs** by preserving their placement in the list of jobs generated by the CustomPac Installation Dialog.
 - You can now **merge IBM data sets into target user-defined data sets**. They can be used as target data set to merge eligible IBM component data sets. However, user-defined data sets are not eligible for merge component data sets.
 - **Data Set Origin** (IBM or User-Defined) is a new attribute off of Modify System Layout "View and change data sets by selected attributes".
 - Support for WebSphere products that use **IBM Installation Manager, and for additional z/OSMF plug-ins.**
 - **Barcoded labels on 3590 and 3592 tape** cartridges designed to enable them to be used in IBM Automated Tape Libraries (ATLs) without first having to label them locally.
 - **Program Directories and ServerPac: Installing Your Order in Adobe PDF** format on DVD for orders on tape, so they can be read immediately from an optical drive on a workstation without the need to download them from z/OS data sets after the RECEIVE job has been run.

20

© 2011 IBM Corporation

Previous ServerPac Install Enhancements

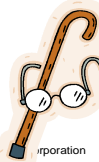


ServerPac in z/OS R12:


- DVD delivery support for product offerings.
- Allows SDSF and JES SMP/E zones to be merged, without having to merge either into the z/OS BCP zone.
- Propagation of your zone names into the mlq of the SMP/E CSI data set names in Modify System Layout.
- A number of jobs, which are no longer needed, are removed.
- Several z/OSMF setup improvements have been made, including new tasks for z/OSMF applications new in z/OSMF.

And some oldies that you may not have appreciated from z/OS R11:

- No more overallocated error on volumes with existing space after ALLOCDS has run! (That is, MSL changes are allowed.)
- Order Tivoli OMEGAMON and other Tivoli monitoring products by themselves, with OUT z/OS in group "IBM Tivoli Monitoring Products" on Shopz.




z/OS R13 Install-Related Enhancements

Some Helpful z/OS R13 Sysprog Enhancements! 

- ☺ **z/OS UNIX: Prevent Content Overlay during MOUNT**
- ☺ **DFSMS: IEBPDSE Batch Program**
- ☺ **SDSF: Cursor-sensitive Sort**

R12 goodies you also get in R13:

- ☺ **BCP: Timed Event Data Report**
- ☺ **BCP: Several PROGxx-related enhancements**
- ☺ **DFSMS: IDCAMS DELETE pdsname(*) command**



22 © 2011 IBM Corporation

z/OS V1R12 enhancements you may have missed

z/OS R12 Enhancement: Timed Event Data Report

Timed Event Data Service allows users to record events to a Timed Event Data Table to help determine program flow and performance. Each event is time-stamped and stored with data collected by the service as well as optional data provided by the user. A new macro, IEATEDS, is used to invoke the REGISTER service to establish a Timed Events Data Table for the user, and a RECORD service to add events to the table. A new REXX Exec, IEAVFTED, is provided to format the data in the table as a Timed Event Data Report and write it to a data set or z/OS UNIX file. The Timed Event Data Report includes a human-readable section and a spread sheet section that can be imported into a spread sheet program for analysis. Using the IEATEDS service, users will be able to RECORD events at various places in their code to determine elapsed time performance, and the optional data can be used to provide context, such as loop counters and return codes. By using the Timed Event Data service, users can determine how factors such as code changes, tuning changes, and configuration changes affect elapsed time performance. The IEATEDS macro contains a complete description of the service along with several examples.



Note: The IEAVFTED REXX exec is a compiled REXX program which requires the full REXX compiler runtime libraries (the REXX Library Product) installed before attempting to use IEAVFTED. Note that IEAVFTED will not work with the REXX Alternate Runtime Library z/OS Base (which is FMID HWJ9143).

There are other commands and tools that you may be familiar with using for determine execution time in certain areas. The program IPLSTATS is mentioned heavily in the Redbook *System z Mean Time to Recovery Best Practices*.

Reference information: *z/OS MVS Programming: Authorized Assembler Services Reference, Volume 2 (EDT-IXG)*

z/OS R12 Enhancement: Several PROGxx-related enhancements in z/OS R12. Only a few are included below.


- Dynamic LPA Enhancements: Automatic alias processing option in PROGxx and on CSVDYLPA
- Dynamic Exit support for REPLACE
- New PROGxx DEFAULTS support

z/OS R12 Enhancement: Using the access method services DELETE command to delete all members of a PDS or PDSE

In z/OS V1R12, DFSMS access method services (IDCAMS) adds a new wildcard option to the DELETE command, which lets you delete all members of a PDS or PDSE.

Previously, only one data set member could be deleted on each invocation of the DELETE command, by specifying the member name in parenthesis, for example, DELETE pdsname(member name). Now, you can supply an asterisk (*) as a wild card, to delete all the members of the data set - for example: DELETE pdsname(*)

For more information about this update to the IDCAMS DELETE command, see [z/OS DFSMS Access Method Services for Catalogs](#).


z/OS R13 Install-Related Enhancements 

z/OS UNIX: Prevent Content Overlay during MOUNT

- Have you ever “lost” updates to files when they’ve been over-mounted?
- Do you want the system to let you know when you’re mounting over something?


★ **Exploitation Example:**

```
SETOMVS NONEMPTYMOUNTPT=WARN
```



```
0290 BPXF263I FILE SYSTEM 901
0290 IBMUSER.PRODUCT.ZFS
0290 HAS BEEN MOUNTED ON A NONEMPTY DIRECTORY
0290 IFF1061 IFF2271 0080 ALLOCATED TO SYS00050
```

```
SETOMVS NONEMPTYMOUNTPT=DENY
```



```
BPXF135E RETURN CODE 00000088, REASON CODE 055B063C. THE MOUNT FAILED FOR FILE
SYSTEM IBMUSER.PRODUCT.ZFS.
***
```

23 © 2011 IBM Corporation

z/OS UNIX: Prevent Content Overlays During MOUNT

The BPXPRMxx parmlib statement NONEMPTYMOUNTPT can be used to control how the system mounts the file systems on the non-empty mount points.

- The NOWARN option specifies that the mount is to take place without any warning message when the mount point is a non-empty directory. The contents of that directory are hidden for the duration of the mount.
- The WARN option specifies that the mount is to take place with a warning message when the mount point is a non-empty directory. The contents of that directory are hidden for the duration of the mount.
- The DENY option specifies that mounting is not to take place when the mount point is a non-empty directory.

During OMVS initialization, if the mount point is contained in an NFS file system, the NONEMPTYMOUNTPT setting is not honored.

D OMVS,OPTIONS on z/OS R13 shows the default (NOWARN) when I haven't done anything:

```

0090 IPCSHMNSEGS      =          500      IPCSHMSPAGES      =      262144
0090 SUPERUSER        = BPXROOT      FORKCOPY          = COW
0090 STEPLIBLIST       = /system/steplib
0090 USERIDALIASTABLE=
0090 PRIORITYPG VALUES: NONE
0090 PRIORITYGOAL VALUES: NONE
0090 MAXQUEUEDSIGS     =          1000     SHRLIBRGNSSIZE    =      67108864
0090 SHRLIBMAXPAGES    =          4096     VERSION           = /
0090 SYSCALL COUNTS    = NO            TTYGROUP          = TTY
0090 SYSPLEX            = NO            BRM SERVER        = N/A
0090 LIMMSG             = NONE          AUTOCVT           = OFF
0090 RESOLVER PROC     = RESOLVER      LOSTMSG           = ON
0090 AUTHPGMLIST       = NONE
0090 SWA                = BELOW        NONEMPTYMOUNTPT  = NOWARN
0090 SERV_LINKLIB      =
0090 SERV_LPALIB       =

```

MA c 04/02

Connected to remote server/host pokvmt14.pok.ibm.com using port 23

As expected (and what you'll see on pre-R13) systems, a mount on a non-empty mountpoint gives no indication that contents were covered up. After this file system is mounted at a mountpoint in the root, I've "lost" everything in /samples. Obviously ,for covered contents that are SMP/E-maintained, this could lead to some problems down the road:

```

READY
MOUNT FILESYSTEM( 'IBMUSER.PRODUCT.ZFS' ) MOUNTPOINT( '/samples/' ) TYPE(ZFS)
MODE(RDWR)
READY

```

Now, I'll unmount the file system, and then turn on NONEMPTYMOUNTPT to WARN to give me an indication if the mountpoint is not empty.

```

SETOMVS NONEMPTYMOUNTPT=WARN
BPXO015I THE SETOMVS COMMAND WAS SUCCESSFUL.
BPXO062I NONEMPTYMOUNTPT WAS CHANGED FROM NOWARN TO WARN.

```

Doing the same mount as before, you can see that a warning now appears in **SYSLOG**.

```

BPXF263I FILE SYSTEM 875
IBMUSER.PRODUCT.ZFS
HAS BEEN MOUNTED ON A NONEMPTY DIRECTORY

```

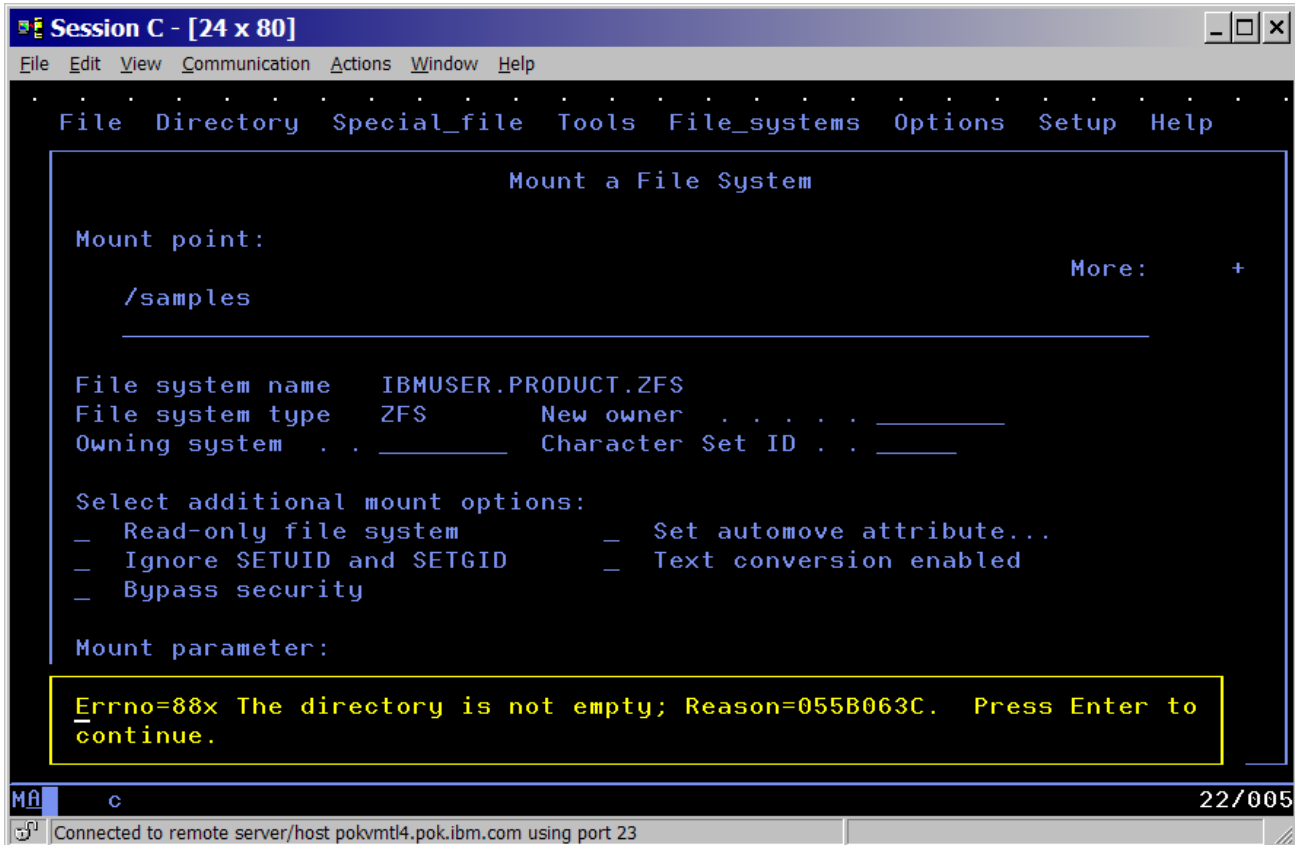
Now, I'll unmount the file system, and then turn on NONEMPTYMOUNTPT to DENY to cause the mount to fail when the mountpoint is not empty.

```

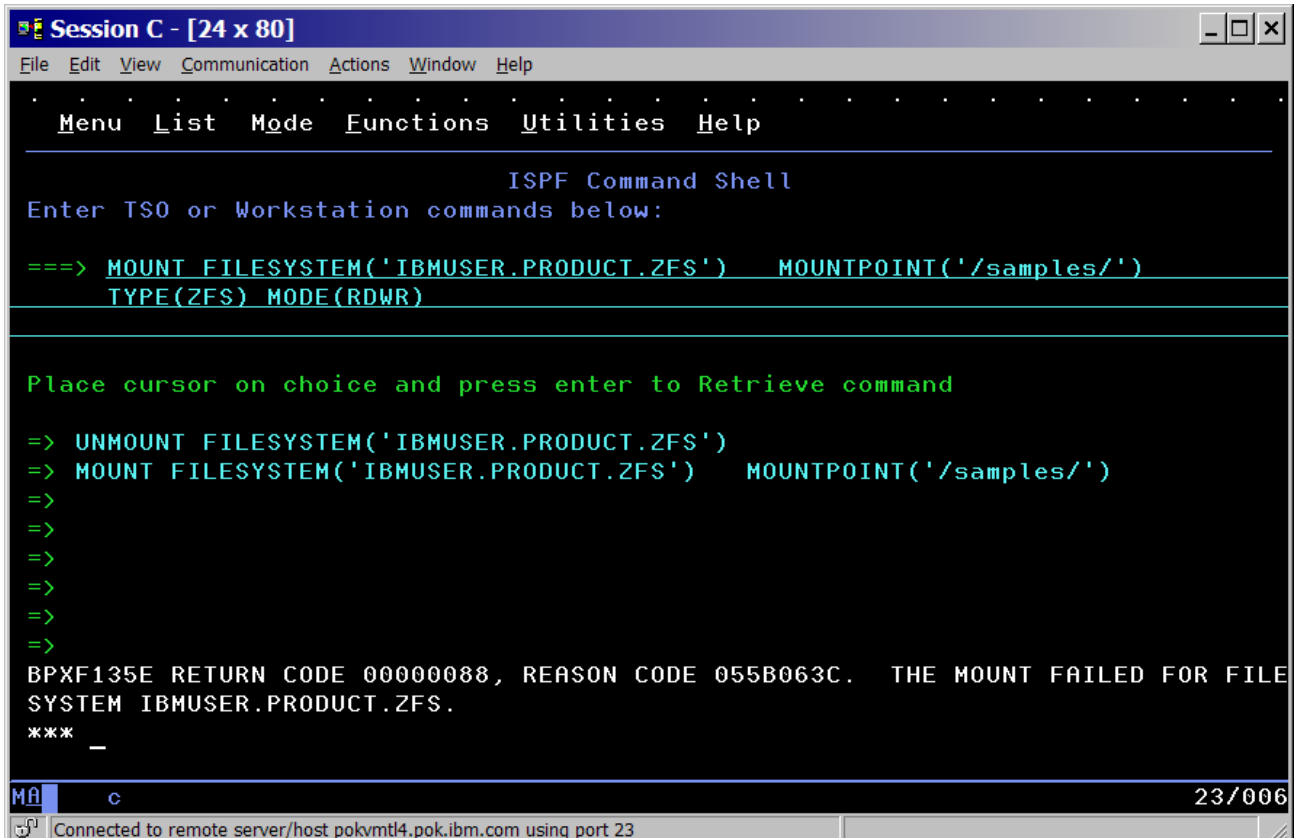
SETOMVS NONEMPTYMOUNTPT=DENY
BPXO015I THE SETOMVS COMMAND WAS SUCCESSFUL.
BPXO062I NONEMPTYMOUNTPT WAS CHANGED FROM WARN TO DENY.

```

Doing the same mount as before, you can see that the mount fails. This is from an ISHELL MOUNT:



This is from a TSO/E MOUNT:



A bpxmtext 063C gives:

Notice: unknown modid, reason text may be incorrect

JrNonEmptyMntPtDir: The mount point directory is not empty.

Action: Retry the mount on an empty mount point directory.

DFSMS: IEBPDSE Batch Program



- Want to check the structural integrity of a PDSE?
- It's a good idea to verify a PDSE's integrity *before* its deployed, and a problem would be propagated.
- If the DUMP parameter option is specified, the PDSE validation utility issues an ABEND in the PDSE address space, which results on an SVC dump

★ **Exploitation Example:**

```
//STEPCHK EXEC PGM=IEBPDSE, PARM= 'DUMP '
//SYSPRINT DD SYSOUT=*
//SYSLIB DD DSN=SYS1.SIEALNKE, DISP=SHR
// DD DSN=MWALLE.UTIL.JOBS, DISP=SHR
```

★ **Sample Success Output:**

```
***** TOP OF DATA ***
IGW700I PDSE Directory Validation Successful
DSN:SYS1.SIEALNKE
ADPages:36 IXRecords:1054
NDPages:6 IXRecords:315
AD ND Tree Nodes:315
IGW700I PDSE Directory Validation Successful
DSN:MWALLE.UTIL.JOBS
ADPages:65 IXRecords:3985
NDPages:29 IXRecords:1590
AD ND Tree Nodes:1590
***** BOTTOM OF DATA *
```

DFSMS: IEBPDSE (PDSE Validation) Program

You can use IEBPDSE to validate a PDSE data set and determine whether it is valid or corrupted.

IEBPDSE uses the following input:

- A PDSE data set, to be validated.

IEBPDSE produces the following output:

- A message data set that contains informational messages (for example if the data set was found to be corrupted), the results of the validation check, and error messages.

IEBPDSE is controlled by job control statements. Utility control statements are not used.

A PARM keyword may be specified: **PARM [DUMP| NODUMP]** If the DUMP option is specified, the PDSE validation utility issues an ABEND in the PDSE address space, which results on an SVC dump.

IEBPDSE Return Codes

IEBPDSE returns a code in register 15 to indicate the results of program execution. The return codes and their meanings are:

Codes Meaning

- 00 (X'00')** Successful completion.
- 04 (X'04')** The input PDSE is slightly damaged. Processing continues.
- 08 (X'08')** The input PDSE is corrupted. The utility ends.
- 12 (X'0C')** The input PDSE could not be opened. The utility ends.
- 16 (X'10')** The input data set is not a PDSE. The utility ends.

z/OS R13 Install-Related Enhancements

SDSF: Cursor Sensitive Sort



- You can sort on a column by placing the cursor on the column title and pressing Enter.
- Additionally, the column titles are now ISPF point-and-shoot fields, so you can tab to them.
 - Make sure you have “Tab to point-and-shoot fields” enabled under ISPF Settings in Option 0.

★ **Exploitation Example:**

Before:

```
SDSF OUTPUT ALL CLASSES ALL FORMS      LINES 408      LINE 1-9 (9)
COMMAND INPUT ==>
NP  JOBNAME  JobID  Owner  Prty C Forms  Dest  Tot-Rec
TCAS  STC00017  IBMUSER  144 A STD  LOCAL  14
IBMUSERM  JOB00024  IBMUSER  144 W STD  LOCAL  47
IBMUSERN  JOB00025  IBMUSER  144 W STD  LOCAL  50
IBMUSERO  JOB00026  IBMUSER  144 W STD  LOCAL  46
IBMUSERZ  JOB00021  IBMUSER  144 W STD  LOCAL  71
IBMUSERZ  JOB00022  IBMUSER  144 W STD  LOCAL  1
IBMUSERZ  JOB00022  IBMUSER  144 W STD  LOCAL  89
IBMUSERZ  JOB00023  IBMUSER  144 W STD  LOCAL  1
IBMUSERZ  JOB00023  IBMUSER  144 W STD  LOCAL  89
```

After:

```
SDSF OUTPUT ALL CLASSES ALL FORMS      LINES 408      LINE 1-9 (9)
COMMAND INPUT ==>
NP  JOBNAME  JobID  Owner  Prty C Forms  Dest  Tot-Rec
IBMUSERZ  JOB00022  IBMUSER  144 W STD  LOCAL  1
IBMUSERZ  JOB00023  IBMUSER  144 W STD  LOCAL  1
TCAS  STC00017  IBMUSER  144 A STD  LOCAL  14
IBMUSERO  JOB00026  IBMUSER  144 W STD  LOCAL  46
IBMUSERM  JOB00024  IBMUSER  144 W STD  LOCAL  47
IBMUSERN  JOB00025  IBMUSER  144 W STD  LOCAL  50
IBMUSERZ  JOB00021  IBMUSER  144 W STD  LOCAL  71
IBMUSERZ  JOB00022  IBMUSER  144 W STD  LOCAL  89
IBMUSERZ  JOB00023  IBMUSER  144 W STD  LOCAL  89
```

25

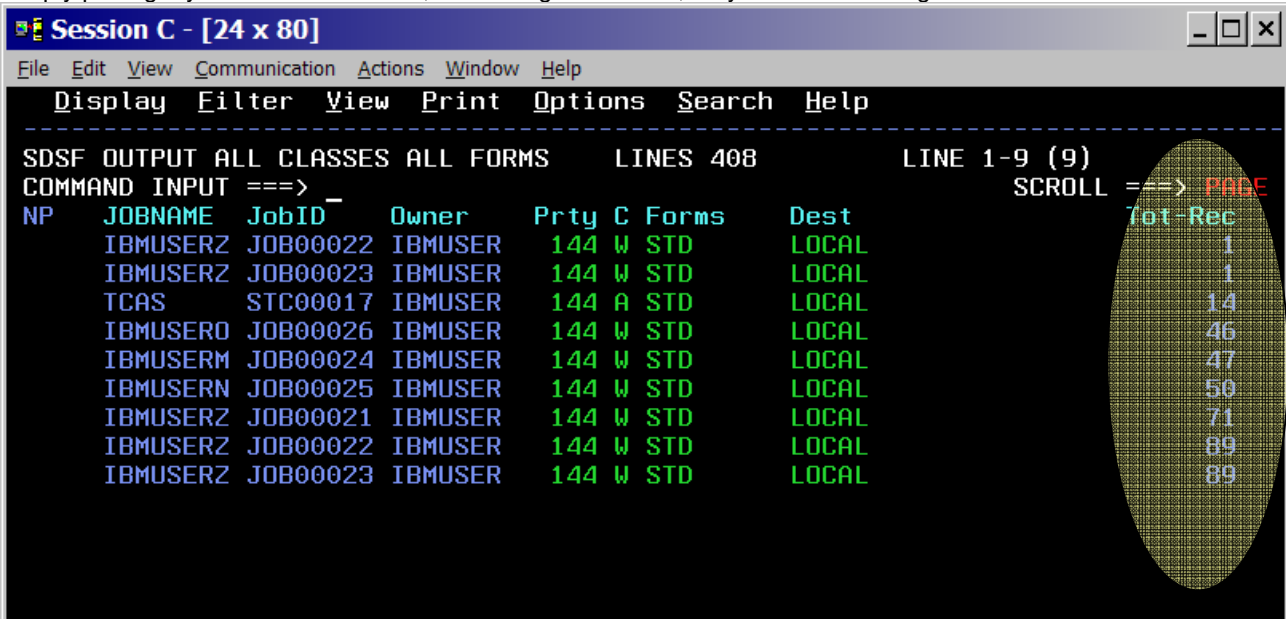
SDSF: Cursor-sensitive sort

You can now sort a tabular panel by placing the cursor on a column title and pressing Enter. Under ISPF, you can use the Tab key to move the cursor to the column titles. This is a quick alternative to typing the SORT command.

This function is so easy to use! Here’s an example of a tabular column (Tot-Rec) that I want to sort in ascending order, below is the original order:

The screenshot shows a terminal window titled "Session C - [24 x 80]". The window contains a table of job information. The columns are: NP, JOBNAME, JobID, Owner, Prty, C, Forms, Dest, and Tot-Rec. The Tot-Rec column values are: 14, 47, 50, 46, 71, 1, 89, 1, 89. A yellow oval highlights the Tot-Rec column.

Simply putting my cursor on "Tot-Rec", and hitting enter once, they are in ascending order:



Hitting enter again, puts the column in descending order:



Yet another enter, puts it back in original order.

Migrating to z/OS R13: Part 3 of 3 Summary



- **JES2:**
 - [PDBLENG](#) references, [z11 mode](#).
- **JES3:**
 - [*S main](#), [FLUSH](#) after XCF DOWN response.
- **Language Environment:**
 - Nothing for runtime options ☺, [CEEPRMxx](#), and [D CEE](#) command response
- **z/OS UNIX:**
 - Shell mount and unmount changes, changer for read-only root support, [tsocmd](#), [MAXSOCKETs](#), [skulker](#).
- **SMP/E and ServerPac Enhancements for Install:**
 - [REPORT CROSSZONE](#), [REPORT SYSMODS](#) in different globals, [SYSMOD Comparison](#) [HOLDDATA](#) report on [REPORT SYSMODS](#) command.
 - More user data set support, barcodes, PDFs for program directories
- **Helpful z/OS R13 Sysprog Enhancements:**
 - [IEBPDSE](#), [NONEMPTYMOUNTPT](#), and [Cursor-sensitive Sort](#)