

### **z/OS 1.10: Migration – 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 1.10. It is strongly recommended that you review all three of these sessions for a complete migration picture.

In part three, the migration actions for selected elements in z/OS 1.10 will be covered. Other selected elements such as DFSMS, Language Environment, and z/OS UNIX System Services will be included. Some of the system programmer enhancements introduced in z/OS 1.10 will be included.

*The general availability date for z/OS V1 R10 is planned for September 26, 2008.*

Systems and Technology Group IBM

## Trademarks

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

AFF	DFSMShsm	FICON	MVS/SP	PROSM	VTAM
ADK	DFSMShsm	HiperStrike	NetSpool	Print Services Facility	WebSphere
AspIlet	DFSORT	IBM	NetView	PrintWiz	zArchitecture
IBM Mail manager	Enclia	IBM i.xt	Open Class	RACF	zOS
CICS	ESCON	IMS	OS/390	Resource Link	zVM
DB2	Eservr	Infoplat	Parallel Sysplex	RSAP	zSeries
DB2 Universal Database	FICON	Language Environment		SS/390	
DFSMShsp		Multiple		SS/390 Parallel Enterprise Server	
DFSMShsm		MVS		Storage Timer	
		MVSESA		SystemPac	
				Systemz.5	
				ThinL	
				TotalStorage	
				Vantage	
				VSE/ESA	

\* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

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

**Notes:**

Performance is an inherent throughout IBM sales based on measurements and judgments 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 work being done, the user's job definition, the I/O configuration, the device configuration, and the workload pattern. Therefore, an accurate cost for plans that are based on actual results is based on measurements completed in the performance sales office.

IBM trademarks include those shown on this page, as well as trademarks of other companies. Reproduction, or use, in any form, is prohibited without the express written permission of IBM Corporation.

All customer computer files or documents in this publication are provided as the property of the customer to 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 sell other products, services or features discussed in this document internationally, and the information may be subject to change without notice. Contact 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 included for the convenience of those products or their published manufacturers. IBM does not test these products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Decisions 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 geographic area.

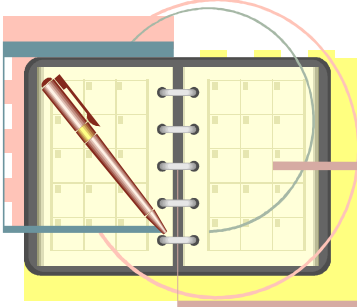
ENABLING BUSINESS.  
A THROUGH Z.

© 2008 IBM Corporation


z/OS 1.10: Migration Part 3 IBM

## z/OS 1.10: Migration – Part 3 Agenda

- **Migration Actions**
  - DFSMS
  - Distributed File Service – zFS and SMB
  - ES2
  - ES3
  - Language Environment
  - SDSF
  - z/OSUNIX
- **z/OS R10 Installation Enhancements**
- **z/OS R10 System Programmer Enhancements**
- **Summary**



3 © 2008 IBM Corporation

z/OS 1.10: Migration Part 3 

## Structure of *z/OS R10: Migration Presentations*

**f Part 1**

- Planning information for R10: Content, Coexistence, System Requirements, ...

**f Part 2**

- Migrations actions for z/OSR10 from selected elements: General, BCP, Communications Server, ICSF, HCD, ICKDSF, LDAP, ISPF, RACF, TSO/E and XL C/C++.

**f Part 3**


- Migrations actions for selected elements you just saw.
- Also included are highlights of some enhancements you'll see in z/OSR10 for system programmers.

You need to review all 3 to get a complete migration picture!  
*(along with z/OS Migration book!)*

4

© 2008 IBM Corporation

z/OS 1.10: Migration Part 3



## Definition of "Migration Action":

*f* **Migration is**

- the installation of a new version or release of a program to replace an earlier version or release.

*f* **After a successful migration, the applications and resources on the new system function the same way they did on the old system, if possible.**


*f* **Migration does not include exploitation of new functions except for new functions that are now required to use.**

*f* **Migration actions are classified as:**


- **Required:** required for all users
- **Required-IF:** only required in certain cases
- **Recommended:** good to do because it 1) is good practice, 2) may be required in the future, 3) resolves performance or usability problem

*f* **Migration actions are also classified as when they may be performed:**

- **NOW , Pre-First IPL, or Post-First IPL**



**Means "don't overlook!"**



**Means a Health Check (or the Migration Checker) can help**

5

### Migration Definitions and Classifications

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

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

### Migration Requirement Classification and Timing

The migration actions are classified as to their requirement status:

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

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

- **Now.** These are migration actions that you perform on your current system, either because they require the current system or because they are possible on the current system. You don't need the z/OS V1R10 level of code to make these changes, and the changes don't require the z/OS V1R10 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 V1R10.
- **Pre-First IPL.** These are migration actions that you perform after you've installed z/OS V1R10 but before the first time you IPL. These actions require the z/OS V1R10 level of code to be installed but don't require it to be active. That is, you need the z/OS V1R10 programs, utilities, and samples in order to perform the migration actions, but the z/OS V1R10 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 V1R10 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 V1R10 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 V1R10. You need a running z/OS V1R10 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 V1R10 to be active to perform the task. You might perform the task quite a while after the first IPL.

Icons used in the subsequent foils:



means that you shouldn't overlook this migration action.



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

z/OS 1.10: Migration Part 3 IBM

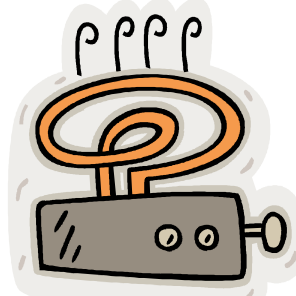
## Elements with Migration Actions for z/OS R10

*f* Documented in **z/OS V1R10 Migration (available on 8/5/08)**

- For complete migration tasks for z/OS R10, see this book!
  - from R8 to R10, and R9 to R10, customized books
  - "When behaviors aren't the same anymore, migration actions are called for."

*f* **From z/OS R8 to z/OS R10:**

<ul style="list-style-type: none"> <li>➤ BCP</li> <li>▪ CIM</li> <li>➤ Communications Server</li> <li>➤ Cryptographic Services – ICSF and OCSF</li> <li>➤ DFSMS</li> <li>▪ DFSORT</li> <li>➤ Distributed File Service – zFS and SMB</li> <li>▪ FFST</li> <li>➤ HCD</li> <li>• HCM</li> <li>▪ IBM Tivoli Directory Server</li> <li>➤ ICKDSF</li> <li>• Infoprint Server</li> <li>➤ Integrated Security Services - LDAP</li> </ul>	<ul style="list-style-type: none"> <li>➤ ISPF</li> <li>➤ .JES2</li> <li>➤ .JES3</li> <li>➤ Language Environment               <ul style="list-style-type: none"> <li>• Library Server</li> <li>• NFS</li> <li>• RMF</li> </ul> </li> <li>➤ SDSF</li> <li>➤ Security Server (RACF)               <ul style="list-style-type: none"> <li>• SMP/E</li> </ul> </li> <li>➤ TSO/E</li> <li>➤ XL C/C++</li> <li>➤ z/OS UNIX</li> </ul>
--	---



➤ means that some of that element's migration actions follow within this session and the previous one  
 • means that you need to look for that element's migration actions in the book


6 © 2008 IBM Corporation

### Migration Actions for Elements Between z/OS R8 and z/OS R10

When migrating from z/OS R8 to z/OS R10, the specified elements in the foil above have required migration actions. Refer to *z/OS 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 R9 to R10, there are fewer migration actions that are applicable. Use the customized *z/OS Migration* book to see just the migration actions appropriate for this path.


z/OS 1.10: Migration Part 3



### DFSMS Migration Actions for z/OS R10 from z/OS R8

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


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

 • IMBED and REPLICATE support are planned to be removed in a future release. KEYRANGE isn't recommended to use.

• Determine which VSAM data sets are affected, then schedule a time to redefine them.

• Use tools (IMBDSHIP and Migration Checker).

• In a future release of z/OS, when DFSMSshm or DFSMSdss recalls or restores a VSAM data set with either IMBED or REPLICATE attribute or both, the attributes will be removed.

 • **Specify a control data set ID for DFSMSrmm (Required, as of R9)**


• A control data set ID is now required, where it was optional previously.

• Define it using the CDSID operand of the OPTION command in EDGRMMxx, and then set it in the data set using the utility or restarting RMM.

7

© 2008 IBM Corporation

z/OS 1.10: Migration Part 3



### DFSMS Migration Actions for z/OS R10 from z/OS R8

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

• **Accommodate changes to DEVSERV command output (Required-IF, as of R10)**

• DEVSERV QDASD output is changed: cylinder column expanded, new column heading ECK and is reduced to three characters.

• DEVSERV PATHS output is changed: adds the number of cylinders in a new column called CYL, ALT column has been replaced with the CYL column.

• Both outputs have been shifted one position to the right.

• Change any dependency you have on this changed output.

• **Accommodate changes in the CA sizes assigned to newly allocated VSAM data sets (Required-IF, as of R10)**

• The CA size selection is done differently on z/OSV1R10 for all new VSAM data sets (not just those on EAVs).

• The CA size is 1, 3, 5, 7, 9 or 15 tracks and is determined by the primary and secondary space allocation amounts.


• As before, you cannot explicitly specify a CA size. Before z/OSV1R10, other possible CA sizes existed. Adjust any primary and secondary requests to accommodate the CA size you rely upon.

8

© 2008 IBM Corporation

z/OS 1.10: Migration Part 3

**DFSMS Migration Actions for z/OS R10 from z/OS R8**



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


- **Allow the dynamic volume count (data class attribute) to exceed 1 when allocating SMS-managed data sets with partitioned organization (Required-IF, as of OA22738)**
  - As of R8 and R9 base level, dynamic volume count could not be greater than 1 when allocating an SMS-managed data set with PO. This was an unpopular change, so it was reversed.
  - Make sure OW 22738 is installed to keep this behavior as it was before.
- **Update automation for changed messages (Required-IF, as of R10)**
  - Many many messages have changed: message text, messages have been added, changed, and removed.
  - Make sure automation can handle these changes.

9

© 2008 IBM Corporation

z/OS 1.10: Migration Part 3

**DFSMS Migration Actions for z/OS R10 from z/OS R8**



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

- **Define default partitioning support for DFSMSrmm (Recommended, as of R10)**
  - Before R10: default processing for volumes that were not defined was to add only system-managed volumes, and not non-system managed volumes to the RMM control data set.
  - As of R10: default processing when PARTITION commands are in use (either because they are defined or because there are no REJECT commands) is to add all undefined volumes.
  - For the same default processing, you must add a PARTITION command if you do not use REJECT commands in parmlib. May want to use the new default behavior though!
- **Use DELETE disposition support for tape data sets for DFSMSrmm (Required-IF, as of R10)**
  - Before R10: RMM didn't consider the disposition of a tape data set. If a job step abended, RMM tracked this in the data set record and provided VRS management by using the ABEND VRSes.
  - As of R10: RMM considers the normal disposition for a tape data set. All tape data sets are considered. If normal disposition is DELETE, the data set record is updated to track the data set as "deleted" by disposition processing. This is just like the "abend" flag.
  - Change JOBNAME(DELETED) or DSNAME('DELETED') in data set name VRSes to avoid conflict.

10

© 2008 IBM Corporation



## DFSMS Migration Actions for z/OS R10 from z/OS R8



### • Migration Actions Post-First IPL:

- **Update programs or procedures that depend on output in reports from the LISTDATA PINNED command (Required-IF, as of R10)**
  - Track addresses in output of IDCAMSLISTDATA PINNED are in a different format, regardless of volume size.
  - As of R10: a range is identified, and the track address is printed in 8 hex digits.
- **Accommodate the change of IDCAMS DEFINE for RECORDS involving VSAM linear data set (Required-IF, as of OA21369 on R9)**
  - Before OA21369: when space for linear data sets was calculated for the DEFINE command with the RECORDS parameter, an implied CI size of 4096 was used when the CI size was larger than 4096.
  - As of OA21369: in the same scenario, the actual larger CI size is used instead of 4096.
  - Consider adjusting the value of RECORDS when a linear data set is defined with CISIZE greater than 4096. Using the same IDCAMS DEFINE CLUSTER job will result in the allocated space being larger than before because the actual CISIZE (instead of 4096) has an influence on the amount of space allocated to the data set. You need to decrease the value of RECORDS to achieve an allocation amount similar to before.
  - Install DB2 PTF PK42840 to avoid DB2 side effects from this change.

11

© 2008 IBM Corporation

## DFSMS Migration Actions for z/OS R10 from z/OS R8



### • Migration Actions Post-First IPL:

- **Use the new version of the DEFRAG command in DFSMSdss (Recommended)**
  - Before R10: DEFRAG command contained a CONSOLIDATE keyword.
  - As of R10: a new DEFRAG command is introduced (for EAV usage), which drops the CONSOLIDATE keyword. Replacement for the CONSOLIDATE keyword on the new DEFRAG command is the CONSOLIDATE command. Need facility class profile access to STGADMIN.ADR.CONOLID and STGADMIN.ADR.DEFRAG to use.
  - Old DEFRAG command is still available, when specifying the VERSION1 keyword (which isn't for EAV usage).
  - Old JCL that uses CONSOLIDATE with DEFRAG (and without VERSION1) will continue to run, however CONSOLIDATE command will be executed first followed by the DEFRAG command.
- **Define facility class names to protect (H)BACKDS NEW NAME commands (Required-IF, as of R10)**
  - As of R10: new hsm facility class names protect the BACKDS NEW NAME and HBACKDS NEW NAME commands. Also, authorization for data set name on NEW NAME is checked. New facility class names must be defined to protect from unintended use of this support by unauthorized users.
  - Define new facility class profiles to authorize to the (H)BACKDS commands:
    - STGADMIN.ARC.BACKDSNEWNAME
    - STGADMIN.ARC.ENDUSER.HBACKDSNEWNAME

12

© 2008 IBM Corporation

## DFSMS Migration Actions Between z/OS V1 R8 and z/OS V1 R10

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

### DFSMS Migration Actions You Can Do Now



#### **DFSMSdftp: Back up SMS control data sets (Recommended)**

*Recommended to ensure data integrity.*

**Migration action:** Do the following on your pre-z/OS V1R10 systems:

1. Back up SMS control data sets according to established procedures in the event that fallback is required. The control data set format is VSAM linear.
2. Install all coexistence PTFs defined for z/OS V1R10.

In addition, if you modified and activated a higher-level policy on a pre-z/OS V1R10 system, do the following to ensure that the ACDS can be accessed on z/OS V1R10:

1. On the pre-z/OS V1R10 system, save the active ACDS as an SCDS with the SETSMS SAVESCDS command.
2. On z/OS V1R10, update, translate, validate, and activate the saved SMS policy.

**Note:** With z/OS V1R10 (and prior releases by APAR OA22373), CDS uplevel checking only checks the formatted release against the updating release level. The construct size change no longer causes warning messages..



#### **DFSMSdftp: Redefine existing VSAM data sets that contain the IMBED, REPLICATE, and KEYRANGE attributes (Recommended)**

*Recommended to avoid degraded performance and wasted DASD space.*

No supported release of z/OS honors the IMBED, REPLICATE, and KEYRANGE attributes for new VSAM data sets. In fact, using these attributes can waste DASD space and often degrades performance. Servicing these VSAM data sets has become increasingly difficult. In some cases, unplanned outages have occurred. For these reasons, IBM recommends that you stop using IMBED and REPLICATE, and that you minimize or eliminate your use of KEYRANGE. IMBED and REPLICATE were intended as performance improvements and have been obsoleted by newer, cached DASD devices. Striped data sets provide much better performance than KEYRANGE and should be viewed as a candidate for any existing KEYRANGE data sets.

**Migration action:**

1. Determine which VSAM data sets were defined with the IMBED, REPLICATE, or KEYRANGE attribute. To help you perform this task, you can get a tool that reads existing VSAM data sets and reports which ones have these attributes. The tool is available from the software server (<ftp://software.ibm.com>) in the `s390/mvs/tools` directory as `IMBDSHIP.JCL.TRSD`. Download the file in binary format and uncompress it on your z/OS system using `AMATERSE` or `TRSMMAIN`. Instructions for using the tool are included in the downloaded JCL. See APAR II13894 for additional information.

**Notes:**

- The tool only checks data sets that are on DASD. Data sets that were migrated by DFSMSShsm or backed up by DFSMSDss continue to retain their IMBED, REPLICATE, and KEYRANGE attributes when recalled or restored. In a future release of z/OS, when support for IMBED and REPLICATE is removed and data sets are recalled or restored by DFSMSShsm or DFSMSDss, the IMBED and REPLICATE attributes will be removed during the recall or restore and the data sets will be in a usable state. Because the data sets will be in a usable state, there is no need for a tool to identify data sets that were backed up or migrated at a time when the data sets had the IMBED or REPLICATE attribute and support for IMBED and REPLICATE still existed.
- “AMATERSE” and “TRSMMAIN” are names for a service aid that compresses and decompresses data exchanged with IBM. “AMATERSE” is the preferred program name since its integration into z/OS V1R9. (AMATERSE is also available for z/OS V1R7 and V1R8 by PTFs UA36927 and UA36928 for APAR OA19194.) “TRSMMAIN” is the original program name and is now shipped as an alias entry point to AMATERSE. For more information about AMATERSE, including several differences with TRSMMAIN, see *z/OS MVS Diagnosis: Tools and Service Aids*.

- Schedule a time for the affected VSAM data sets to be unavailable, and redefine them.

Please note that while no announcement has been made of the specific date or release that data sets with these attributes will no longer be able to be opened, it would be prudent to identify the data sets in your installation and begin converting them to versions that do NOT contain these attributes.



### **DFSMSrmm: Specify a DFSMSrmm control data set ID (Required, as of R9)**

Before z/OS V1R9, a control data set ID was optional. If you did not define one in z/OS V1R8 (with the PTF for DFSMSrmm APAR OA17965 installed), warning message EDG0229E NO CDSID WAS SPECIFIED FOR OPTION IN PARMLIB was issued but DFSMSrmm initialization did not fail.

Starting with z/OS V1R9, a control data set ID is required. You must define it using the CDSID operand of the OPTION command in parmlib member EDGRMMxx, and then set it in the control data set either by using the EDGUTIL utility or by restarting DFSMSrmm. If you do not define a control data set ID, you cannot start DFSMSrmm, and warning message EDG0237E MISSING IDENTIFIER FOR THE CONTROL DATA SET is issued.

#### **Migration action:**

- If you have entries in GRSRNLxx for QNAME(SYSZRMM) RNAME(MASTER.RESERVE), complete the steps in “DFSMSrmm: Update GRSRNLxx” before continuing here.
- Define a control data set ID in parmlib member EDGRMMxx by using the CDSID operand of the OPTION command. Do this on each system in the RMMplex. Use a different CDSID for each control data set.
- Set the control data set ID in the control data set by using the EDGUTIL utility or by restarting DFSMSrmm on one system.

### **DFSMShsm: Define the BCDS with a maximum record size of 6544 for the FRBACKUP command (Required-IF, as of R9)**

*Required if you plan to issue an FRBACKUP command.*

DFSMShsm control data set records created for the FRBACKUP function can be up to 6544 bytes in length. In order for an FRBACKUP command to complete successfully, the maximum record size of the BCDS must be defined as 6544. While an FRBACKUP command that processes only a limited number of volumes may process successfully with a smaller record size, this environment is not recommended. In z/OS V1R9, if the BCDS maximum record size is less than 6544, DFSMShsm will fail FRBACKUP commands with message ARC1843I FAST REPLICATION BACKUP FUNCTIONS WILL NOT BE PERFORMED. BCDS MAXIMUM RECORD SIZE MUST BE 6544.

**Migration action:** Follow the standard procedures for reorganizing the DFSMShsm BCDS. Define all BCDS clusters with RECORDSIZE(334 6544) and CISIZE(12288).

### **DFSMSrmm: Use the new default for the EDGHSKP XREPTTEXT DD statement (Recommended, as of R10)**

*Recommended because the new default requires less DASD space and fewer resources to create the extract.*

The default processing for the EDGHSKP XREPTTEXT DD statement is changed so that only extended records are created. Before z/OS V1R10, DFSMSrmm report extract processing used allocated ddnames to determine whether extended records were required. If the ddname was REPTTEXT, DFSMSrmm created all records except the extended records in the extract. If the ddname was XREPTTEXT, DFSMSrmm created all records in the extract and then used the D and V records to create extended records.

Starting in z/OS V1R10, by default, the XREPTTEXT DD statement receives only extended (X) records. To tailor the contents of the REPTTEXT and XREPTTEXT DD statements, DFSMSrmm report extract processing now checks for the existence of the RPTTEXT command in the SYSIN file, and if the RPTTEXT command is found, DFSMSrmm writes the selected extract records to the data set specified by either the REPTTEXT or XREPTTEXT DD statement.

The new default of only X records matches what most customers want. In addition, the new default requires less DASD space and fewer resources to create the extract.

**Migration action:** To obtain the new behavior, which is recommended, do nothing.

If you want the prior behavior, which is for the XREPTTEXT DD statement to contain all records, not just X records, update any report extract JCL used to create a report extract file containing extended records (run to create records other than the extended records) by adding the following statements to your existing JCL:

```
//SYSIN DD *
RPTTEXT RECORDS(X,O,R,B,S,P)
/*
```

This change must be made to your JCL before IPLing z/OS V1R10 and running extract, but it can be made on your current (pre-z/OS V1R10) release (with the coexistence PTF installed if z/OS V1R9) because it will only be used by DFSMSrmm on z/OS V1R10.

**DFSMSrmm: Update GRSRNLxx (Required, as of R9)**

Starting with z/OS V1R9, an additional ENQ name is used by DFSMSrmm for CDS serialization. The new ENQ name now includes the CDSID, which is now a required parmlib option. If you have a GRSRNLxx entry for DFSMSrmm, you must either convert it to a generic RNL entry or add an additional specific RNL entry. Furthermore, if you add an additional specific RNL entry, the RNAME value specified in that entry must be of the correct length. If you do not update existing GRSRNLxx entries, or do not specify the RNAME value correctly, there could be performance or serialization problems related to the CDS volume.

DFSMSrmm uses both the existing and the new ENQ names, so if you have existing GRSRNLxx entries for DFSMSrmm you must update them to handle the serialization the same way for both ENQ name cases. The new serialization is enabled automatically the first time z/OS V1R9 or later DFSMSrmm is started.

**Migration action:**

1. If you have entries in GRSRNLxx for QNAME(SYSZRMM) RNAME(MASTER.RESERVE), update those entries as shown in the following step. Do this on all coexisting and fallback systems and on your planned z/OS V1R10 system before the first IPL. If you do **not** have entries in GRSRNLxx for QNAME(SYSZRMM) RNAME(MASTER.RESERVE), you have no action to take; DFSMSrmm CDS serialization will be based on both a hardware reserve and SYSTEMS ENQs.
2. Update existing entries in GRSRNLxx for QNAME(SYSZRMM) RNAME(MASTER.RESERVE). The simplest way to do this is by using a generic RNAME entry, that is, by changing the existing definition from TYPE(SPECIFIC) to TYPE(GENERIC). Here is an example for an exclusion RNL:  
RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(SYSZRMM) RNAME(MASTER.RESERVE)  
The same change is required for any conversion RNLs you might have for DFSMSrmm. For example:  
RNLDEF RNL(CON) TYPE(GENERIC) QNAME(SYSZRMM) RNAME(MASTER.RESERVE)  
Updating the entries in this way ensures that, regardless of your CDSID, the correct exclusion or conversion action is always taken by global resource serialization and both the original and the new ENQ names are handled the same way.

An alternative to creating generic RNL entries is to add a second specific RNL entry that includes the CDSID, as shown in the following RNL(EXCL) example. In this case, you would replace *cdsid* in RNAME with your selected CDSID as specified in the EDGRMMxx parmlib member OPTION statement:

```
RNLDEF RNL(EXCL) TYPE(SPECIFIC) QNAME(SYSZRMM) RNAME(MASTER.RESERVE)
RNLDEF RNL(EXCL) TYPE(SPECIFIC) QNAME(SYSZRMM) RNAME(MASTER.RESERVE.cdsid)
```

**Requirement:** The length of RNAME must be exactly 23 characters. If the length of your CDSID is less than eight characters, you should use a generic entry as described above. If you do not use a generic entry, you must specify RNAME as the hexadecimal constant

RNAME(X'D4C1E2E3C5D94BD9C5E2C5D9E5C54Bxx..xx4040'), where

X'D4C1E2E3C5D94BD9C5E2C5D9E5C54B' represents MASTER.RESERVE and xx..xx4040 represents your CDSID appended with blanks to eight characters in length.

Update the GRSRNLxx entries on each system before implementing a control data set ID and before starting DFSMSrmm on z/OS V1R9 or later.

**DFSMS Migration Actions Pre-First IPL****DFSMSdfp: Ensure that the Language Environment run-time library is available for DLLs (Required-IF)**

*Required if your installation builds or references DLLs.*

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

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

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

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

If you intend to use the Stand-Alone Services provided by DFSMSdss, you must use the DFSMSdss BUILDSA function to create the Stand-Alone Services IPL-capable core image

**Migration action:**

1. Prepare for Stand-Alone Services by creating a Stand-Alone Services IPL-able core image with the BUILDSA command. With the BUILDSA command you can specify the device (card reader, tape drive, or DASD volume) from which Stand-Alone Services will be IPLed. You can also specify the operator console to

be used for Stand-Alone Services. The BUILD\_SA function builds the IPL-able core image under the current operating system and determines a record size based on whether the IPL is from card, tape, or DASD.

2. Use RACF or another external security system to protect the SYS1.ADR.SAIPLD.Vvolser data set and the Stand-Alone Services modules.
3. If you haven't done so already, make a backup copy of your system that can be restored by this function. For information about backing up volumes, see *z/OS DFSMSdss Storage Administration Guide*.

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

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

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

**Migration action:**

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

**DFSMSdss: Replace the INDD and INDYNAM keywords in COPY DATASET and DUMP DATASET jobs (Recommended)**

*Recommended if your jobs issue the COPY DATASET or DUMP DATASET commands and you do not want to receive informational messages stating that DFSMSdss has replaced INDD and INDYNAM with the correct keywords.*

The INDD or INDYNAM keyword on a COPY DATASET or DUMP DATASET command is used to perform either a logical data set copy or a physical data set dump, depending on the command.

Starting in z/OS V1R8, the INDD and INDYNAM keywords are being replaced as follows:

- For uses of the COPY DATASET command, the INDD and INDYNAM keywords are replaced with the LOGINDD and LOGINDYNAM keywords, respectively.
- For uses of the DUMP DATASET command, the INDD and INDYNAM keywords are replaced with the PHYSINDD and PHYSINDYNAM keywords, respectively.

You can still use INDD and INDYNAM on z/OS V1R8 and higher. However, if you do, you will receive an informational message stating that DFSMSdss has replaced INDD or INDYNAM with the correct keywords. To avoid receiving such messages, review and update any DFSMSdss jobs that issue these commands to use the correct keywords.

**Note:** If the PTF for APAR OA20242 (for R8 and R9) is not installed, message ADR146I is not issued for COPY DATASET with INDD, COPY DATASET with INDYNAM, or DUMP DATASET with INDYNAM. (The PTF does not affect DUMP DATASET with INDD.)

**Migration action:** Check your DFSMSdss jobs for usage of the COPY DATASET and DUMP DATASET commands. Update the JCL, as follows:

- For uses of the COPY DATASET command, replace any occurrences of the INDD or INDYNAM keywords with the LOGINDD or LOGINDYNAM keywords, respectively.
- For uses of the DUMP DATASET command, replace any occurrences of the INDD or INDYNAM keywords with the PHYSINDD or PHYSINDYNAM keywords, respectively.

**DFSMSdfp: Provide a 140-byte work area when using OBTAIN (Required-IF, as of R10)**

*Required if you have programs that do not specify at least a 103-byte DSCB return area.*

Programs must pass the address of a 140-byte return area when they search for a data set name using the OBTAIN macro with the CAMLST SEARCH option or the OBTAIN SVC 27 with the SEARCH for DSNAME operation code. In releases prior to z/OS V1R10, the system returned data in only the first 101 bytes of the return area. Beginning with z/OS V1R10, the system will return data in the first 103 bytes. Therefore, you should upgrade your programs to provide the entire 140-byte return area; otherwise, you might encounter storage overlay problems or program interruptions.

**Migration action:** Upgrade programs to provide the entire 140-byte return area as stated in the reference information below.

**DFSMSdfp: Accommodate new defaults for z/OS Global Mirror (XRC) parameter values (Req-IF, as of R10)**

*Required if you rely on the previous XRC default values.*

Default values have changed for two parameters of the SHADOW statement in the XRC parmlib member. Before z/OS V1R10, the RequireUtility parameter had a default value of NO, and the UtilityDevice parameter had a default value of FLOAT. In z/OS V1R10, these default values have changed as follows:

- The default value of the RequireUtility parameter is now YES, which prevents XADDPAIR of a non-XRCUTL volume when an active storage control session does not exist for the specified (or default) SCSESSION. The previous default of NO enables XADDPAIR of a non-XRCUTL volume to proceed when an active storage control session does not exist.
- The default value of the UtilityDevice parameter is now FIX, which changes the method used for selecting a utility device. The FIX value causes XRC to use the most recently added XRCUTL volume as the utility device for the storage control session. The previous default of FLOAT allows the storage control to dynamically pick the primary volume to be used as the utility device for the storage control session. The nonreserved device with lowest I/O activity will typically be selected

**Migration action:** If your XRC sessions rely on the previous default values, you must add statements in your XRC parmlib to specify those values, thus overriding the new defaults.

#### **DFSMSdftp: Update programs that process SMF record type 19 (Required-IF, as of R10)**

*Required if you rely on the previous SMF 19 record length.*

In z/OS V1R10, the length of SMF type 19 records has increased. A type 19 record is created for each DASD volume that is online when IPL occurs, when a HALT EOD or SWITCH SMF command is issued, and when a DASD is varied offline. The type 19 record describes the space usage on the volume, and new fields have been added to the record for track-managed free space statistics and volume size information. Programs that process this record type might need to be modified to support the expanded length of the record or to utilize the new fields. SMF type 19 records are recorded by LSPACE processing when SMF=YES or SMF=TEST is specified on the LSPACE macro invocation. YES specifies that the caller wants an SMF record containing volume information to be written. TEST specifies that LSPACE is to test for an active SMF system that has been defined to request volume accounting before writing the SMF record.

**Migration action:** Check any programs that process SMF type 19 records to determine if you must make any changes. If necessary, modify the programs to use the new record length.

#### **DFSMSdftp: Accommodate changes to DEVSERV command output (Required-IF, as of R10)**

*Required if you have a dependency on the output of the DEVSERV QDASD or DEVSERV PATH command.*

z/OS V1R10 includes several changes to output from the DEVSERV command:

- DEVSERV QDASD output is changed as follows: the CYL column was expanded to accommodate a nine-digit number for cylinders. In addition, the extended function consistency check column EF-CHK is reduced to three characters. The new column heading is ECK.
- DEVSERV PATHS output is changed to add the number of cylinders, in a new column labeled CYL. To accommodate the largest possible number of cylinders for a device (nine digits) in this new column, the original ALT column is replaced with this new CYL column and unnecessary spaces around the PIN and DDC columns are removed.
- The output for both DEVSERV QDASD and PATHS has been shifted one position to the right, leaving the first character position blank.

**Migration action:** Change any dependency you have on the output of DEVSERV QDASD or DEVSERV PATHS. Examples of the output changes can be found in *z/OS Migration*.

#### **DFSMSdftp: Update programs that use DCOLLECT (Required-IF, as of R10)**

*Required if you use DCOLLECT to process data set (type D) records to determine whether data sets are overallocated.*

In a DCOLLECT type D record, the DCDOVERA field is a 4-byte field that reports on the amount of overallocated space for a data set. Before z/OS V1R10, the field was defined as a 31-bit signed number. Beginning with z/OS V1R10, DCDOVERA is defined as a 32-bit unsigned number in order to accommodate larger values.

**Migration action:** If you use the IDCDOUT macro to map DCOLLECT records, reassemble your programs to pick up the current mapping.

If you do not use the IDCDOUT macro to map DCOLLECT records, be aware that the field definition has changed for DCDOVERA from signed to unsigned. Use the new field definition when you create DCOLLECT reports.

#### **DFSMSdftp: Accommodate changes in the CA sizes assigned to newly allocated VSAM data sets (Required-IF, as of R10)**

*Required if you have a dependency on the CA size selected for VSAM data set and the corresponding primary or secondary space amounts.*

z/OS V1R10 includes a change in control area (CA) size selection for all new allocations of VSAM data sets on any volume type (not only extended address volumes). The system selects a control area size of 1, 3, 5, 7, 9, or 15 tracks. Previously, the system selected from a wider variety of possible CA sizes. As previously, you cannot explicitly specify control-area size. Generally, the primary and secondary space allocation amounts determine the control-area size. For requests where either the primary or secondary allocation amount is smaller than one cylinder, the system might adjust the primary and/or secondary quantity. The system might also select a CA size that is different from what was selected in a prior release. For example, a TRK(24,4) request results in a control area of 5 tracks, and primary and secondary amounts of 25 and 5 tracks, respectively.

**Migration action:** CA size is derived from the requested primary and secondary allocation. Because the CA size is computed differently in z/OS V1R10, adjust any primary and secondary allocation requests to accommodate the CA size you rely upon.

If you have any programs that depend upon a specific CA size that is no longer supported, change the programs to accept a CA size of 1, 3, 5, 7, 9, or 15 tracks.

**DFSMSdfp: Tell users that the ISMF SORTREC line operator is no longer available (Required-IF, as of R10)**

*Required if you use the SORTREC line operator.*

Beginning with z/OS V1R10, the DFSORT interactive storage management facility (ISMF) panels are no longer supported. Consequently, the SORTREC line operator is no longer available.

**Migration action:** Tell users that the SORTREC line operator is no longer available. Trying to use it will result in the message COMMAND SORTREC NOT FOUND. An alternative to SORTREC is to use DFSORT against the data set from a batch job.

**DFSMSdfp: Allow the dynamic volume count (data class attribute) to exceed 1 when allocating SMS-managed data sets with partitioned organization (Required-IF, as of OA22738)**

*Required if you use data classes that specify dynamic volume counts when allocating data sets with partitioned organization, and you want to allow the dynamic volume count (data class attribute) to exceed 1 (which is the pre-z/OS R8 situation).*

Before z/OS V1R8, when allocating an SMS-managed data set with partitioned organization, you could assign to it a data class that specified a dynamic volume count greater than 1. Then, in z/OS V1R8 and V1R9, the base code was changed so that the dynamic volume count could not be greater than 1. However, APAR OA22738 on z/OS V1R8 and V1R9 restored the original situation, that is, the dynamic volume count could be greater than 1 again. Now, in z/OS V1R10, the original (pre-z/OS V1R8) situation (greater than 1 allowed) has been permanently restored.

This migration action is not relevant to non-SMS data sets because the dynamic volume attribute only applies to SMS-managed data sets.

**Migration action:** If you desire, allow ACS routines to select a data class with a dynamic volume count greater than 1 for SMS-managed data sets with partitioned organization. Note that because PDSs and PDSEs can only exist on one volume (and cannot be extended to a second volume), the dynamic volume count will be ignored.

**DFSMSdfp: Update automation for changed multiline WTO messages (Required-IF, as of R9 with APAR)**

*Required if you have automation which examines the message text of messages described.*

In z/OS V1R8 without the PTF for APAR OA15780 installed, and z/OS V1R9 without the PTF for APAR OA21810 installed, some DFSMSdfp IEAxxxx messages are formatted using multiple single-line WTO messages when the length of the message requires more than one line. The second WTO message includes the text “[CONTINUATION OF IEAXXXX]”, followed by the text “SNS=xxxxxxxxx...”.

Starting with z/OS V1R8 with the PTF for APAR OA15780 installed, and z/OS V1R9 with the PTF for APAR OA21810 installed, the complete message is formatted using a single multiline WTO message, and the text “SENSE=xxxx xxxx ...”, if applicable, is included as part of the multiline message. The text “[CONTINUATION OF IEAXXXX]” is no longer provided.

Example in z/OS V1R8 without the PTF for APAR OA15780 installed, and z/OS V1R9 without the PTF for APAR OA21810 installed:

```
IEA442E dev REPORTS DISABLED INTERFACE ON nn -- FAULT CODE = cccc -
NOTIFY CE
[CONTINUATION OF IEA442E SNS=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx]
```

Example in z/OS V1R8 with the PTF for APAR OA15780 installed, z/OS V1R9 with the PTF for APAR OA21810 installed, and later:

```
IEA442E dev REPORTS DISABLED INTERFACE ON nn -- FAULT CODE = cccc -
NOTIFY CE,SENSE=xxxx xxxx xxxx xxxx xxxx xxxx xxxx
```

Note that the word “SENSE” is now spelled out, and the 32 bytes of sense data have a blank character after every four sense bytes.

The specific messages affected are IEA442E, IEA448I, IEA449A, IEA451I, IEA452I, IEA453I, IEA454E, IEA455E, IEA457I, IEA458I, IEA459I, IEA468I, IEA469E, IEA473I, IEA474E, IEA475E, IEA476E, IEA477I, IEA478E, IEA479E, IEA480E, IEA481I, IEA482I, IEA484I, IEA491E, IEA494I, IEA497I, and IEA498I.

**Migration action:** Change your automation to expect the new message text format.

**DFSMSdftp, DFSMSdss, and DFSMShsm: Update automation for changed messages (Req-IF, as of R10)**

*Required if you have automation routines that examine the message text of the messages listed below.*

In z/OS V1R10, the text for messages with the following IDs has been changed: ADRxxxxx (DFSMSdss), ANTxxxxx (DFSMSdftp), ARCxxxxx (DFSMShsm), DMOxxxxx (DFSMSdftp), IDCxxxxx (DFSMSdftp), IEAxxxxx (DFSMSdftp), IECxxxxx (DFSMSdftp), IEExxxxx (DFSMSdftp), and IGDxxxxx (DFSMSdftp).

Some messages have had a period (.) removed from the end of the message text. Other messages have had text added, changed, or removed. These changes can affect automation programs that examine the text of the messages.

**Migration action:** Update your automation to handle the following **DFSMSdftp** message changes:

- The message text or explanation has been changed in the following messages to support extended address volumes, including new track, cylinder, or return code information:
  - ANTF073E, ANTF0401E, ANTF0443E, ANTX5117E, ANTX8030W, ANTM6004I, ANTM6005I, ANTM6006E, ANTM6007E, ANTM6013E, ANTM6015I, ANTM6016I, ANTM6020I, ANTM6021I, ANTM6022I, ANTM6023I, ANTM6024I, ANTM6025I, and ANTM8809I
  - DMO0011I, DMO0012I, DMO0013E, DMO0031E, DMO0032E, DMO0033I, DMO0040I, and DMO0041I
  - IDC3009I
  - IEA434I, IEA479E, IEA480E, and IEA480E
  - IEC142I, IEC144I, IEC602I, IEC609I, and IEC987I
  - IEE459I
  - IGD002I, IGD031I, IGD17268I, IGD17279I, IGD17269I, IGD17318I, and IGD17389I
- Message IGD17038I now provides a number of volumes (“n volumes”) in place of the words “this request”.

Update your automation to handle the following **DFSMSdss** message changes:

- The period has been removed from the end of the following messages: ADR021I, ADR032W, ADR037E, ADR045T, ADR109I, ADR146I, ADR148I, ADR204W, ADR238I, ADR239I, ADR240W, ADR226W, ADR297I, ADR299E, ADR301E, ADR325E, ADR343E, ADR344I, ADR345D, ADR359W, ADR363E, ADR365E, ADR388W, ADR390I, ADR397E, ADR401W, ADR405E, ADR444E, ADR459E, ADR499I, ADR701E, ADR702E, ADR711I, ADR715W, ADR716W, ADR719E, ADR721I, ADR731W, ADR733I, ADR734I, ADR736E, ADR737W, ADR767I, ADR785E, ADR795E, ADR796E, ADR801I, ADR804W, ADR805E, ADR865E, ADR880I, ADR901E, ADR902I, ADR903I, ADR904I, ADR906I, ADR907I, ADR908E, ADR912I, ADR913E, ADR916W, ADR940I, ADR941E, ADR942E, ADR943E, ADR944E, ADR946E, ADR969E, ADR970E, ADR974I, ADR978W, ADR979E, ADR985W, ADR990E, and ADR993E.
- The message text or explanation has been changed in the following messages to support extended address volumes, including new track, cylinder, or return code information: ADR024E, ADR037E, ADR037E, ADR040I, ADR071I, ADR145I, ADR152E, ADR201E, ADR210E, ADR226W, ADR232E, ADR234I, ADR235I, ADR247W, ADR250I, ADR251W, ADR255E, ADR256E, ADR258I, ADR260I, ADR261I, ADR262I, ADR299E, ADR309E, ADR317I, ADR336I, ADR345D, ADR364W, ADR366W, ADR373E, ADR379E, ADR433E, ADR436E, ADR441W, ADR498E, ADR520I, ADR557I, ADR558I, ADR735W, ADR817W, ADR818E, ADR887I, ADR888I, ADR932W, ADR935W, ADR938E, ADR939E, ADR945W, ADR946I, and ADR973E. (The following messages have also been changed but they will never be analyzed by automation because they are only issued during stand-alone restore when z/OS is not in operation: ADRY0500I, ADRY0700I, ADRY1500W, ADRY1524W, ADRY1711I, ADRY2500I, and ADRY3521I.)
- The ending period has been removed and additional message text has been added to messages ADR037E and ADR345D.
- Message ADR049E has been enhanced to include abend reason code information at the end.
- References to JOBCAT and STEPCAT have been removed from messages ADR380E, ADR415W, ADR470W, ADR472E, and ADR485E.

Update your automation to handle the following **DFSMShsm** message changes:

- The message text or explanation has been changed in the following messages to support extended address volumes, including new track, cylinder, or return code information: ARC0120I, ARC0157I, ARC0185I, ARC0269I, ARC0638I, ARC0742I, ARC1309I, ARC1359I, ARC1520I, and ARC1524I.



**DFSMSrmm: Define default partitioning support (Recommended, as of R10)**

*Recommended if you have no REJECT commands in the DFSMSrmm parmlib so that you get, by default, the desired type of volumes added to the DFSMSrmm control data set when the volumes are not defined to DFSMSrmm.*

Before z/OS V1R10, the DFSMSrmm default processing for volumes that were not defined to DFSMSrmm was to add only system-managed volumes, and not non-system-managed volumes, to the DFSMSrmm control data set. Starting in z/OS V1R10, the DFSMSrmm default processing when PRTITION commands are in use, either because they are defined or because there are no REJECT commands, is to add all undefined volumes. If you do not use REJECT commands in parmlib, you must add a PRTITION command if you want to ensure that the DFSMSrmm default processing stays the same.

**Migration action:**

- To ensure that the DFSMSrmm default processing stays the same, add the following PRTITION command to parmlib: PRTITION VOLUME(\*) TYPE(NORMM) NOSMT(IGNORE)  
As a result, system-managed volumes are automatically added to the DFSMSrmm control data set during library entry/insert and OPEN processing, but non-system-managed volumes are not added automatically. Note that when adding volumes automatically, DFSMSrmm no longer creates a rack number but will use an empty rack number if it exists.
- To get the new behavior, do one of the following:
  - If you have no REJECT commands in parmlib, take no action. The new default is provided by the default PRTITION command created by DFSMSrmm.
  - If you have REJECT commands in parmlib, migrate to PRTITION and OPENRULE commands to exploit the new capabilities.

**DFSMSrmm: Update volume replacement policies (Required-IF, as of R10)**

*Required if you are using the usermod to the EDGMUPD module to override the shipped value.*

If you are using the usermod to the EDGMUPD module to set LIMIT=*n*, where *n* is shipped as X'00000001', you can no longer use this usermod. Before z/OS V1R10, you could use the usermod to override the shipped value. The usermod could disable the volume replacement action setting by setting a value of X'00000000'. The usermod was first available with APAR OW43870 and may be called RMREPL.

Starting in z/OS V1R10, you must use the function of MEDINF REPLACE in parmlib to implement any override or disablement. The usermod no longer works with the EDGMUPD module, and DFSMSrmm processing is based solely on MEDINF REPLACE.

**Migration action:**

- To disable the replacement of IBM media, you can specify the following in the EDGRMMxx parmlib member: MEDINF NAME(IBM) REPLACE(PERM(0))
- To override the DFSMSrmm built-in default processing of PERM(1), use any of the available REPLACE operand values.

**DFSMSrmm: Use DELETE disposition support for tape data sets (Required-IF, as of R10)**

*Required if you use JOBNAME(DELETED) or DSNNAME('DELETED') in any data set name VRSEs.*

Before z/OS V1R10, DFSMSrmm did not consider the disposition of a tape data set. If a job step abended, and CLOSE was entered because ofabend processing, DFSMSrmm tracked this in the data set record and provided VRS management by using the ABEND VRSEs.

Starting in z/OS V1R10, DFSMSrmm considers the normal disposition for a tape data set. All tape data sets, whether using system temporary data set names or not, are considered. If the normal disposition (as coded in the JCL or dynamic allocation) is DELETE, the data set record is updated to track the data set as "deleted" by disposition processing. This is a flag just like the "abend" flag. VRSEL processing is updated to process the new restricted "DELETED" VRSEs.

**Migration action:** If you use JOBNAME(DELETED) in any data set name VRSEs, you must change the VRS to use JOBNAME(DELETED\*) to avoid a conflict with the new 'DELETED' VRS support.

If you use DSNNAME('DELETED') in any data set name VRSEs, you must change the VRS to use DSNNAME('DELETED\*') to avoid a conflict with the new 'DELETED' VRS support.

**DFSMSrmm: Use the backward-compatible CIM provider (Required-IF, as of R10)**

*Required if you use a CIM client with the DFSMSrmm CIM agent and decide not to update your code to handle the new key formats.*

The keys used for the DFSMSrmm CIM classes have changed with z/OS V1R10. If you have CIM clients that have been working on z/OS V1R9 or z/OS V1R8 and you want to use them against z/OS V1R10, and the clients have some hard-coded or specific processing of the keys of DFSMSrmm CIM classes, you can do either of the following:

- Use the backward-compatible CIM provider.
- Update your code to handle the new key formats and then use the z/OS V1R10 CIM provider. If you choose this path, see migration action “DFSMSrmm: Replace CIM providers and CIM classes”.

The following table shows the old keys of DFSMSrmm CIM classes and the new compound keys, which have formats of concatenated strings containing the values of the old keys delimited with “+” and appended with spaces by the fix length if needed.

Class Name	Key Transformation
IBMRMM_Dataset	DataSetName, PhysicalFileSequenceNumber, VolumeSerialNumber, and CdsID keys have been replaced with Name key of format DataSetName+FileSeq+Volser+CdsID.
IBMRMM_Location	LocationName, LocationType, and CdsID keys have been replaced with Tag key of format LocationName+LocationType+CdsID.
IBMRMM_LogicalVolume	VolumeSerialNumber and CdsID keys have been replaced with DeviceID key of format Volser+CdsID.
IBMRMM_Owner	OwnerId and CdsID keys have been replaced with InstanceID key of format OrgID:OwnerID+CdsID.
IBMRMM_PhysicalVolume	VolumeSerialNumber and CdsID keys have been replaced with Tag key of format Volser+CdsID.
IBMRMM_PolicyRule	PolicyRuleType, PolicyRuleName, JobNameMask, and CdsID keys have been replaced with PolicyRuleName key of format RuleType+RuleName+JobNameMask+CdsID.
IBMRMM_Product	ProductNumber and CdsID keys have been replaced with IdentifyingNumber key of format ProductNumber+CdsID.
IBMRMM_ShelfLocation	LocationName, MediaName, ShelfLocationNumber, and CdsID keys have been replaced with Tag key of format LocationName+MediaName+Number+CdsID.

Starting in z/OS V1R10, in order to simplify processing of the compound keys, every changed class now has the KeyWithCdsIdName attribute containing the name of its compound key. Additionally, xxxFormat and xxxMask attributes are provided, where xxx is the name of the compound key. For example, TagFormat property of IBMRMM\_PhysicalVolume is set to “Volser+CdsID” and TagMask string contains the consecutive concatenation of six blanks, symbol “+”, and eight blanks.

**Migration action:** The backward-compatible CIM provider is provided as rmmcim19.tar.Z compressed tar archive within the /usr/lpp/dfsms/rmm directory. To use it, go to the Miscellaneous Functions panel provided by the rmmutil.sh script and invoke “Unpack RMM CIM Agent for V1R9”. As a result, you will have the old agent unpacked in the “v1r9” path within the current directory. For example, if you use the /usr/lpp/dfsms/rmm deployment directory, the new agent will reside in /usr/lpp/dfsms/rmm/v1r9.

### **DFSMSrmm: Create or update VRSEs for DFSMSrmm data set names with lowercase or mixed case letters (Recommended)**

*Recommended to ensure that existing data sets with lowercase or mixed case letters are correctly VRS-retained.*

Before z/OS V1R9, data set names using lowercase letters, mixed case letters, or special characters were correctly recorded when created, but the DFSMSrmm TSO subcommands did not support the listing, changing, or deleting of data sets that did not follow MVS data set naming conventions. In addition, you had limited capability to retain such data sets through vital record specifications (VRSEs).

Starting with z/OS V1R9, you can use DFSMSrmm to fully record and manage all data sets names that can be specified in job control statements, including those with lowercase or mixed case letters. You can also use all of the DFSMSrmm TSO subcommands and create VRS data set name masks that allow data sets with lowercase or mixed case letters to be VRS-retained.

**Migration action:** Following the instructions in *z/OS Migration*:

1. Check whether you have any lowercase or mixed case data set names that are on volumes that are VRS-retained.
2. If you have lowercase or mixed case data set names on volumes that are VRS-retained, identify the VRSEs that retain the data set.

3. If any of the matching DSNAMES contain uppercase letters that match the lowercase or mixed case letters in the data set names, the VRSes will no longer match the data set names in z/OS V1R9. You must create new VRSes or change the existing VRSes for the data sets.

Data sets that match generic characters in the VRS data set name masks are not affected. While you run VRSEL vital record processing on releases below z/OS V1R9, you should not delete any VRSes that retain data sets with lowercase or mixed case letters, or you might lose data.

**DFSMSrmm: Customize tape unit names to ensure automatic CBRUXVNL processing (Required-IF, as of R8 with APAR OA19374)**

*Required if you have not yet customized the list of tape unit names..*

Starting with z/OS V1R8, you must customize the list of tape unit names to ensure that DFSMSrmm CBRUXVNL processing is provided automatically for tape data sets.

If you did not install the PTF for APAR OA19374 on z/OS V1R8, if you did not customize the list of tape unit names, and if no match is found in the hardcoded tape unit name table, DFSMSrmm ignores the CBRUXVNL processing and your uncataloged tape data set jobs fail.

**Migration action:**

- To exploit the new tape unit recognition for uncataloged tape data sets, do the following:
  1. Customize the processing by turning on the new code by changing the setting &TAPEDEC SETC to YES.
  2. Customize the hardcoded list of tape unit names to include those used in your installation for your uncataloged tape data sets. The tape unit names are listed in the shipped table called TAPEUNITS.
- To change the decision to issue message EDG8197I for non-RMM managed volumes, customize the CBRUXVNL source code. The decision to issue message EDG8197I for non-RMM managed volumes is made selectable by an option flag in the EDGLCSUP parameter list. The CBRUXVNL exit shipped by DFSMSrmm sets this option flag depending on the circumstances:
  - When a cataloged tape data set is processed, option flag LCSUP\_8197 is set to B'1'.
  - When &TAPEDEC is set to YES and the unit name is a tape unit, option flag LCSUP\_8197 is set to B'1'.
  - Otherwise option flag LCSUP\_8197 is set to B'0'. To turn off message EDG8197I, set option flag LCSUP\_8197 to B'0'.

**DFSMS Migration Actions Post-First IPL**

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

*Required if you use OAM object support.*

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

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

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

**Note:** The following list itemizes the changes to OAM BIND jobs in z/OS V1R10. If you choose to edit a previous version, you must incorporate any new changes as described in the header of each new samplib BIND job:

- CBRPBIND: CBRIEDBS added
- CBRABIND: CBRIEDBS added to CBRIDBS plan
- CBRHBIND: no changes
- CBRIBIND: CBRIEDBS added to CBRIDBS plan.

**DFSMSdfp: Examine the job and hardcopy logs for important Delete/Rename and storage constraint messages (Required-IF as of R10)**

*Required if these messages are not reaching a storage administrator now.*

Before z/OS V1R10, some important DFSMS messages could be suppressed or not issued by some components of the system. An example of such a message is IGD17054I DATA SET NOT FOUND FOR DELETE/RENAME ON VOLUME xxxxxx DATA SET IS aaaa.bbbb.cccc.

In z/OS V1R10, DFSMS writes a number of these existing messages to the job log and hardcopy log for the first time. These include messages for Delete and Rename processing, as well as messages pertaining to storage constraints that need to be addressed to prevent possible storage-related failures. This change is made to help ensure that these important messages reach the attention of a storage administrator.

**Note:** This change does not alter the functioning of the system in any way beyond externalizing messages that may have been suppressed. Also, it applies to messages for SMS-managed data sets as well as non-SMS-managed data sets.

**Migration action:** On a regular basis, just as frequently as you would today, examine the job and hardcopy logs for important messages related to Delete, Rename, and storage constraint processing.

#### **DFSMSdfp: Update programs or procedures that depend on output in reports from the LISTDATA PINNED command (Required-IF as of R10)**

*Required if users or programs depend on output from the LISTDATA PINNED command.*

The track addresses in the output of the IDCAMS LISTDATA PINNED command are in a different format, regardless of the volume size. Before z/OS V1R10, the report listed each pinned track and its associated data set name.

Beginning with z/OS V1R10, the report identifies a range of pinned tracks associated with a data set. Each track address is printed in eight hex digits in the native format of the device, using the form CCCCcccH. In addition, instead of one track being listed per line, consecutive tracks for each data set are gathered into one line and the range of track addresses is shown.

The meaning of “CCCCcccH” is as follows: “CCCC” is the low-order 16 bits of the cylinder address, “ccc” is the high-order 12 bits of the cylinder address, and “H” is the 4-bit track address. When a 28-bit cylinder address is greater than 65535 (X'FFFF'), the F column in the report contains an asterisk.

**Migration action:** SYSPRINT and TSO terminal output from IDCAMS are not intended programming interfaces but the changes are stated here as a convenience to help you identify any changes to a program that uses this output. Change any affected program appropriately to parse the new format.

#### **DFSMSdfp: Accommodate the change of IDCAMS DEFINE for RECORDS involving VSAM linear data sets (Required-IF, as of R9 with OA21369)**

*Required if you define a linear data set with CISIZE greater than 4096 bytes and specify the RECORDS parameter.*

Previously, when space for linear data sets was calculated in response to an AMS DEFINE command with the RECORDS parameter, an implied control interval size of 4096 was used when the control interval size was larger than 4096. As of z/OS V1R9 with APAR OA21369, in the same scenario, the actual (larger) control interval size is used instead of 4096.

**Migration action:** Consider adjusting the value of the RECORDS specification when a linear data set is defined with CISIZE greater than 4096 bytes. If you continue to use the same IDCAMS DEFINE CLUSTER job in such a situation, the allocated space will be larger than in z/OS V1R8 or V1R7 because the actual CISIZE (instead of 4096 bytes) has an influence on the amount of space allocated to the data set. You need to decrease the value of the RECORDS specification to achieve an allocation amount similar to before migration.

**Notes:** 1. DB2 allocates VSAM linear data sets using the RECORDS parameter, so DB2 allocations would be affected by this change when CISIZE greater than 4 KB is used. To avoid this side effect, apply the PTF for DB2 APAR PK42840. With this PTF, the define and extend code in DB2 has been changed to use KB when issuing the IDCAMS DEFINE command.

2. In z/OS V1R8, SMS added a new code to validate the space quantity for volume selection. The space calculation is based on the number of records multiplied by the actual CISIZE instead of the 4 KB record size. If SMS fails the allocation because the primary quantity requested is larger than the total capacity of the largest available volume, the following error message is issued: IGD17279I (n) VOLUMES WERE REJECTED BECAUSE OF INSUFF TOTAL SPACE.

#### **DFSMSdfp: Increase dump space in a Sysplex for VSAM RLS abends (Required-IF as of R9)**

*Required if the existing dump space is not large enough to accommodate additional dumps.*

Before z/OS V1R9, when a VSAM RLS abend occurred, an abend dump was taken only on the originating system where the abend occurred. In z/OS V1R9, new sysplex-wide dumping support causes other systems running VSAM RLS in the sysplex to take dumps as well as the originating system. This change, might result in more dump space being needed on a given system, if dump space is currently limited.

**Migration action:** If more space for RLS abend dumps is needed, use the same procedures you currently do to increase dump data set size, but be aware that this might need to be done more frequently than in the past. In addition, you can use DAE to suppress dumps if needed.

**DFSMSdftp: Add column SUBLEVEL to the DB2 TAPEVOL table for OAM (Required-IF as of R9)**

*Required if you use OAM.*

Before z/OS V1R9, the OAM storage hierarchy consisted of three levels: disk, optical, and tape. In z/OS V1R9, two sublevels are introduced into the tape level, effectively expanding the OAM storage hierarchy into four levels: disk, optical, tape sublevel 1 (TSL1), and tape sublevel 2 (TSL2). You must modify and run the CBRSMR19 migration job to add the new SUBLEVEL column to the DB2 TAPEVOL table, whether you intend to exploit the new function or not.

**Migration action:**

- If your installation's DB2 TAPEVOL table does *not* contain a SUBLEVEL column, edit the sample job CBRSMR19 to make the changes specified in the comments section, and then run the job. The job adds the new TSL SUBLEVEL column to the DB2 TAPEVOL table and primes it with default values.
- If you are migrating from z/OS V1R7, and if your installation's DB2 object directory tables do *not* contain an ODLOBFL column, edit the sample job CBRSMR18 to make the changes specified in the comments section, and then run the job. The job adds the ODLOBFL column to the object directory tables, and drops and re-creates the object directory table views.
- Run OAM DB2 BIND and GRANT jobs. To determine which BIND and GRANT jobs you need to run, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*.

**DFSMSdftp: Accommodate the changed condition code form an IDCAMS DIAGNOSE command against the ICF catalog (Required-IF, as of R9, and on R8 with OA19984)**

*Required if you depend on the condition code from the IDCAMS DIAGNOSE command.*

When using the IDCAMS DIAGNOSE command against an ICF catalog, you might receive the following messages:  
IDC11374I THESE ADDITIONAL CATALOG REFERENCED VOLUMES WERE ENCOUNTERED: volume-serial-numbers

IDC000II FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 4

In this situation, the listed volume serial numbers are referenced by catalog entries but these volume serial numbers were not found as VVDS entries. This could indicate that entries are referencing volumes for which the catalog has no record. Although there is no impact on processing, DIAGNOSE considers this an attention condition and issues a warning message with processing continued.

As of z/OS V1R8 with APAR OA19984, message IDC11374I has been removed from IDCAMS DIAGNOSE command processing in the above situation. The condition code returned from this message is now zero.

**Migration action:** Change procedures that depend on the condition code from the IDCAMS DIAGNOSE command against the ICF catalog. The situation previously reported by warning message IDC11374I is no longer detected and condition code 4 is not set for this reason.

**Notes:** 1. There should be a "SYS1.VVDS.Vvolser" entry for each volume referenced by VSAM entries in the ICF catalog, except where the "SYS1.VVDS.Vvolser" is created by EOVS processing.

2. Message IDC11374I was an attention and might signal entries that are missing or damaged. To add an entry for the missing VVDS to the catalog, use IDCAMS DEFINE RECATALOG on the VVDS. If a VSAM data set incorrectly references the volume, remove the reference from the data set entry with IDCAMS ALTER REMOVEVOLUMES.

**DFSMSdss: Use the new version of the DEFrag command (Recommended, as of R10)**

*Recommended for improved efficiency, improved debugging, and compatibility with extended address volumes.*

Before z/OS V1R10, the DFSMSdss DEFrag command contained a CONSOLIDATE keyword. In z/OS V1R10, a new version of the DEFrag command is added that is compatible with extended address volumes. This new version of the command drops the CONSOLIDATE keyword and adds other keywords to improve efficiency and debugging. To replace the CONSOLIDATE keyword, a new DFSMSdss CONSOLIDATE command is added for use in conjunction with the new version of the DEFrag command. This new CONSOLIDATE command allows the consolidate function to be used without any additional DEFrag processing of a volume.

The pre-z/OS V1R10 version of the DEFrag command with its CONSOLIDATE keyword is still available if you specify it with the new VERSION1 keyword. Note that the previous version is not compatible with extended address volumes.

**Migration action:**

- To use the z/OS V1R10 version of the DEFrag command, specify the DEFrag command without specifying the CONSOLIDATE keyword, and use the new, separate CONSOLIDATE command. Using both the CONSOLIDATE and DEFrag commands consecutively in the same job will give the same information and do essentially the same thing as the old DEFrag command with the CONSOLIDATE keyword. But in addition, the new commands are compatible with extended address volumes, you can use the new MAXTIME and MMOVPCT keywords to limit the amount of time a job runs, and the new CONSOLIDATE command reduces the number of extents of a data set as much as possible rather than only when the extents can be reduced to one.  
To use the z/OS V1R10 CONSOLIDATE command, you must have access to the STGADMIN.ADR.CONSolid profile in the FACILITY class. (Using the z/OS V1R10 DEFrag command continues to require access to the existing STGADMIN.ADR.DEFRAG profile.)
- To use the pre-z/OS V1R10 version of the DEFrag command with its CONSOLIDATE parameter intact, specify the new VERSION1 keyword. Note that this previous version is not compatible with extended address volumes, and data sets will only have extent reduction when they can be reduced to one extent. Existing (created on z/OS V1R9 or earlier) JCL that specifies the DEFrag command with the CONSOLIDATE keyword on z/OS V1R10 (without VERSION1) will cause the CONSOLIDATE command to be executed first, followed by the DEFrag command using the new version of DEFrag.

#### **DFSMSHsm: Define facility class names to protect (H)BACKDS NEWNAME commands (Req-IF as of R10)**

*Required if you use the BACKDS NEWNAME and HBACKDS NEWNAME commands.*

Starting in z/OS V1R10, new DFSMSHsm facility class names protect the BACKDS NEWNAME and HBACKDS NEWNAME commands. DFSMSHsm also checks for RACF authorization to the data set name specified as the NEWNAME parameter. The new facility class names must be defined to protect from unintended use of this support by unauthorized users. This support is only available on z/OS V1R10. Earlier-level systems will be able to recover backup versions created with the NEWNAME keyword.

The NEWNAME keyword is added to the DFSMSHsm (H)BACKDS command. When users specify the (H)BACKDS NEWNAME command, DFSMSHsm creates a backup version of the data set specified on the command and makes the backup version look like it is a backup of the data set specified with the NEWNAME keyword.

**Migration action:** Define new facility class name STGADMIN.ARC.BACKDS.NEWNAME to authorize any BACKDS NEWNAME command and STGADMIN.ARC.ENDUSER.HBACKDS.NEWNAME to authorize any HBACKDS NEWNAME command.

#### **DFSMSHsm: Use the generic rather than the esoteric unit name for duplex generated tape copies (Required-IF as of R10)**

*Required if you need to use the generic unit name.*

Before z/OS V1R10, when a duplex failure generated a tape copy, DFSMSHsm passed the generic equivalent of a SETSYS unit name (such as 3590-1) to the tape copy. Beginning with z/OS V1R10, DFSMSHsm instead passes a SETSYS-specified unit name (such as SETSYS MIGUNITNAME (*esoteric-unitname*)).

This change ensures that the same output devices and unit name that are targeted for the failed duplex copy are targeted for the tape copy. This change affects the DFSMSHsm backup, migration, and recycle functions. The ABARS function is not changed.

**Migration action:** To change the behavior back to always passing the generic unit for the tape copy, use the DFSMSHsm PATCH command to set the MCVTF\_GENERIC\_TCN\_UNIT flag on. For example:

```
PATCH .MCVT.+196 BITS(...1...)
```

To return to the default behavior of using the SETSYS specified unit name, use the following command:

```
PATCH .MCVT.+196 BITS(...0...)
```



#### **DFSMSrmm: Replace DFSMSrmm CIM providers and CIM classes (Required-IF as of R10)**

*Required if migrating from z/OS V1R9: Yes, if you have chosen to use the z/OS V1R10 CIM provider (not the backward-compatible CIM provider) as described in migration action “DFSMSrmm: Use the backward-compatible CIM provider”.*

*Required if migrating from z/OS V1R8: Yes, if you have chosen to use the z/OS V1R10 CIM provider (not the backward-compatible CIM provider) as described in migration action “DFSMSrmm: Use the backward-compatible CIM provider”, or if you previously set up a CIM server and agent for DFSMSrmm and registered the DFSMSrmm CIM classes.*


In z/OS V1R10, the keys used for the DFSMSrmm CIM classes were changed. In z/OS V1R9, DFSMSrmm CIM providers and CIM classes were changed. You must unregister the previous CIM providers and CIM classes, and register the z/OS V1R10 CIM providers and CIM classes.

**Migration action:**

1. If you are planning to use the z/OS V1R10 CIM provider rather than the backward-compatible CIM provider, update your code to handle the new key formats in z/OS V1R10. For the new formats, see migration action “DFSMSrmm: Use the backward-compatible CIM provider”.
2. Using the rmmutil.sh tool, unregister all the pre-z/OS V1R10 CIM providers and unload all the pre-z/OS V1R10 CIM classes.
3. Using the same rmmutil.sh tool, register the complete set of z/OS V1R10 CIM providers and load the z/OS V1R10 CIM classes.


z/OS 1.10: Migration Part 3

## DFS Migration Actions for z/OS R10 from z/OS R8



**Migration Actions You Can Do NOW:**

- **Migration from zFS multi-file system aggregates to compatibility aggregates (Recommended)**
  - As of R8: can no longer mount multi-fs aggregates that are shared.
  - Convert any multi-fs aggregates to compatibility-mode aggregates.
- **Stop using zfsadm setquota on compat-mode aggr (Req-IF, as of R10)**
  - Prior to R10: zfsadm setquota command was accepted on compat-mode aggr (was intended only for multi-fs aggr).
  - As of R10: zfsadm setquota command is not accepted on compat-mode aggr.
- **Migration Actions Post-First IPL:**
  - **Fix syntax errors now reported during SMB startup (Recommended)**
    - Prior to R10: syntax of SBM environment variables weren't validated.
    - As of R10: They are. SMB starts as usual, but may not have started with the intended parameters.
    - Run the z/OSR10 dfssyntax command against the environment variable file (/opt/dfs/local/home/dfskern/envvar, for instance) to see if there are any syntax errors (and when any changes are made to it).



13 © 2008 IBM Corporation

### DFS Migration Actions Between z/OS V1 R8 and z/OS V1 R10

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



#### **Migrate from zFS multi-file system aggregates to compatibility mode aggregates**

**(Recommended)**

*Recommended if you are using zFS multi-file system aggregates because it will become a requirement in a future release.*

In February 2005 (before z/OS V1R7 was available), IBM recommended that zFS file systems contained in multi-file system aggregates that are shared across systems in a sysplex not be shared in a sysplex environment. Beginning with z/OS V1R8, you can no longer mount zFS file systems contained in multi-file system aggregates that are shared across systems in a sysplex. Any attempts to mount zFS file systems contained in multi-file system aggregates will fail with message IOEZ00522E in a z/OS UNIX shared file system environment. Mounting zFS compatibility mode aggregates, which have a single file system per data set, will continue to be supported in all environments.

In a future release, IBM plans to withdraw support for zFS multi-file system aggregates. When this support is withdrawn, only zFS compatibility mode aggregates will be supported.

**Migration action:** Determine if you are using zFS multi-file system aggregates. You can determine this by scanning your zFS IOEFSPRM configuration options file for define\_aggr statements. You should also scan your /etc/rc file for any zfsadm attach commands. If you are using zFS multi-file system aggregates, you should copy the data from each file system into its own zFS compatibility mode aggregate.

#### **Stop using zfsadm setquote of compatibility mode aggregates (Required-IF, as of R10)**

*Required if you normally use the zfsadm setquota command for compatibility mode aggregates.*

Before z/OS V1R10, using the zfsadm setquota command on a compatibility mode aggregate was discouraged but accepted. Starting in z/OS V1R10, if you attempt to set the file system quota for a compatibility mode aggregate,



zFS issues return code 121 (EINVAL) and reason code EFxx69B8. (The maximum size of the file system is known as its *quota*.)

**Migration action:** Do not use the zfsadm setquota command for compatibility mode aggregates.

### **DFS Migration Actions Pre-First IPL**

#### **Accommodate the removal of the allow\_duplicate\_filesystems options from IOEFSPRM (Required, as of R9)**

Before z/OS V1R9, zFS file system names had to be unique across all attached aggregates on a system unless you enabled the **allow\_duplicate\_filesystems** option in the IOEFSPRM file. When the default, **allow\_duplicate\_filesystems=off**, was in effect, any attempts to create a file system with duplicate names were denied.

In z/OS V1R9, to improve performance, the **allow\_duplicate\_filesystems** option has been removed from z/OS, and zFS duplicate file system names are always allowed, regardless of whether you are in a shared file system environment. (In other words, the z/OS V1R9 condition is always “**allow\_duplicate\_filesystems=on**”.) In addition, the zfsadm configquery option **-allow\_dup\_fs** and the zfsadm config option **-allow\_dup\_fs** are removed.

#### **Migration action:**

- If the zFS configuration option **allow\_duplicate\_filesystems** is in your IOEFSPRM file, remove it. If it is specified it is ignored, but removing it can prevent confusion for anyone who does not know that it is no longer valid.
- Discontinue using the zfsadm configquery option **-allow\_dup\_fs** and the zfsadm config option **-allow\_dup\_fs**, which are no longer valid.
- If you now have duplicate zFS file system names and you want to mount one of the duplicate file systems, you must specify the MOUNT PARM AGGREGATE and FILESYSTEM parameters. Note that, in z/OS V1R8, support was removed for mounting a zFS file system contained in a multi-file system aggregate in a shared file system environment.
- Update any automation that might be affected by these message changes:
  - The following message is no longer displayed: IOEZ00319E The file system *Filesystem* will not be attached.
  - Instead, when using multi-file system aggregates, the following message is displayed: IOEZ00315I There are now 2 file systems with the name *Filesystem* in use.
 If there are different values for different sysplex members because at least one of them is at a prior release, the system will act as it did before z/OS V1R9.

### **DFS Migration Actions Post-First IPL**

#### **Fix syntax errors now reported during SMB startup (Recommended, as of R10)**

*Recommended to avoid getting SMB startup errors and to ensure that SMB uses the configuration options you intended.*

Before z/OS V1R10, when you started SMB, the syntax of SMB environment variables was not validated. Beginning with z/OS V1R10, when you start SMB, a new command automatically validates the syntax of the `/opt/dfslocal/home/dfskern/envvar` file or another environment variable file you specify. SMB starts as usual, but if errors are not corrected, SMB might not start with the intended parameters.

**Migration action:** Before starting SMB, run the `dfssyntax` command against the file `/opt/dfslocal/home/dfskern/envvar` or another environment variable file that your installation uses, and then correct any syntax errors that are reported.

#### **Use the new zFS directory buffer cache default in IOEFSPRM (Recommended, as of R9)**

*Not required, but recommended if you use a default for `dir_cache_size` and you want optimal zFS performance.*

The **dir\_cache\_size** option specifies the size of the directory buffer cache in IOEFSPRM, the zFS configuration data set. Before z/OS V1R9, the default value for **dir\_cache\_size** was 2M. Beginning with z/OS V1R9, the default value is 32M. A value of 32M is likely to provide better performance than a value of 2M.

**Migration action:** If you use the default for the **dir\_cache\_size** option, accept the new default value of 32M to get better performance. However, if you prefer the previous default value of 2M (or a value in the range of 2M to 512M), specify it on **dir\_cache\_size** in configuration data set IOEFSPRM.

**Note:** The increased cache size will use more virtual storage, so there is a potential that more paging may be done.

z/OS 1.10: Migration Part 3

## JES2 Migration Actions for z/OS R10 from z/OS R8

### Migration Actions Pre-First IPL:

- Accommodate JES2 dynamic exits function (Required-IF, as of OA23561/OA21346)
  - Review (and change if necessary) any JES2 exit routines that use the following data areas and macros:
    - \$XRT and \$XIT data areas: \$XRT structure is altered. Some fields moved from \$XIT to \$XRT. Reassemblies will, in general, fail for field reference changes.
    - \$LMT data area: two new states may have to be considered.
    - \$MODLOAD and \$MODCHECK macros: message processing has changed.
  - Also, review your installation load modules (those loaded with the LOADMOD initialization statement) to determine if they can support dynamic loading and deletion. May want to use DYNAMIC=NO option on the \$MODULE macro to prevent unwanted processing.
- Update automation for command responses to \$D NODE, \$D EXIT, and \$D LOADmod (Required-IF, as of R10, and as of OA21346)
- Update exit routines that examine the \$JCT, and exits 8, 31, 42, and 45 (Required-IF, as of R9)

14
© 2008 IBM Corporation

### JES2 Actions Between z/OS V1 R8 and z/OS V1 R10

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

<none>

### JES2 Migration Actions Pre-First IPL



#### Accommodate JES2 dynamic exits function (Required-IF, as of R10)

*Required if you use JES2 exit routines or load installation load modules using the JES2 LOADMOD (or LOAD) initialization statement.*

The dynamic exits function of JES2, integrated in z/OS V1R10 and rolled back to prior releases by APARs, makes the following possible:

- Installation load modules (exit modules) can be dynamically loaded, refreshed, or deleted. Note that this function does not apply to modules that are part of the base JES2 product, meaning that you cannot use this function to apply IBM service.
- The list of routines associated with an exit can be dynamically altered.

Because of this new function, you should review and change (if necessary) any JES2 exit routines that use the following data areas and macros:

- The \$XRT and \$XIT data areas. The structure of the \$XRT has been significantly altered. In addition, some fields in the \$XIT have been moved to the \$XRT. Reassembly of your installation code will, in general, fail if it references any fields in the \$XRT or \$XIT that have changed.
- The \$LMT data area. Before z/OS V1R10, a \$LMT can be active (LMT1INVD bit off) or invalid/free (LMT1INVD bit on). Beginning with z/OS V1R10, a \$LMT can be in two new states: it can represent a module that has been logically deleted (LMT2DELT bit on) or a module whose storage has been freed (LMT3FREE bit on). Code that examines \$LMTs might need to take these two new states into consideration.
- The \$MODLOAD and \$MODCHK macros. Message processing for these macros has changed. Beginning with z/OS V1R10, if MESSAGE=NO is coded or defaulted on the macro invocation, the message area

returned in register 1 must be freed by the caller (using \$FREMAIN with SP=0, KEY=1, and LV=128). If the caller does not require the message to be returned, a new MESSAGE=SUPPRESS option can be used. In addition, not all exit routines can tolerate the fact that installation load modules (ones loaded using the JES2 LOADMOD initialization statement) can now be deleted and refreshed (a new copy loaded and the old copy deleted) using an operator command. You can use the new DYNAMIC=NO option on the \$MODULE macro to prevent unwanted processing.

#### Migration action:

1. **Update exit routines that refer to the JES2 \$XRT and \$XIT data areas, if necessary.** The \$XIT is a 256-element array that represents all possible JES2 exits. \$XRTs are pointed to by the \$XIT. Each \$XRT is an array in which each entry represents a routine associated with a JES2 exit. Before z/OS V1R10, the \$XRT was a simple array of elements, each one representing a routine. With the dynamic exits function in z/OS V1R10, the \$XRT starts with a header that describes the list, followed by an array of XRTEs in which each entry represents a routine.  
The change was needed to ensure predictability when routines are invoked by the JES2 \$EXIT facility. When the \$EXIT facility is invoked, it selects the appropriate \$XRT and increments the use count in that \$XRT. The list of routines in that instance of the \$XRT are used by the \$EXIT facility. Operator changes to the list of routines while the \$EXIT facility is in control do not affect the currently active exit environments. When the list of routines for an exit is altered, a new \$XRT is built and, if the existing one is in use, the new one is chained in ahead of the existing \$XRT and the old \$XRT is marked as logically deleted. When the old \$XRT is no longer in use, it is deleted.  
Note that the first \$XRT associated with an exit (pointed to by XITXRTAD) can be logically deleted (XRTFDEL bit on) if there are no longer any routines associated with the exit. If there is a chain of \$XRTs, the second and beyond are always logically deleted. (Only the first \$XRT pointed to by XITXRTAD can be active.) One consequence of this change is that some fields that were in the \$XIT had to be moved to the \$XRT header. In particular, the fields XITBSPL (SPLEVEL check bypassed), XIT#RTNS (number of routines associated with this exit), and XITUSCNT (exit use count) have been moved to the \$XRT.
2. **Update exit routines that refer to the JES2 \$LMT data area, if necessary.** The \$LMT is used to track modules loaded by the JES2 \$MODLOAD service, including modules loaded by the LOADMOD initialization statement and the new \$ADD LOADMOD command. Code that searches \$LMTs needs to consider the state of the \$LMT before examining fields (such as the module address) in the \$LMT. Before z/OS V1R10, the only state needing examination was invalid/free (bit LMT1INVD on). However, the dynamic exits function in z/OS V1R10 introduces two new states that need to be considered: logically deleted \$LMTs and force freed \$LMTs.  
Logically deleted \$LMTs (bit LMT2DELTD on) represent modules that are not being used for new requests (they are logically deleted) but are in use by existing processing. These \$LMTs should not be examined when looking for the current version of a module but can be used if trying to resolve what module a particular address is located in.  
Force freed \$LMTs (bit LMT3FREE on) represent load modules where the actual module is freed before JES2 is done using it. This can only occur if dynamic LPA services is used to delete the module from LPA (something IBM does not recommend doing). Code cannot access the actual load module because the storage is no longer valid. Force freed \$LMTs will eventually be logically deleted (LMT2DELTD set on), although the LMT3FREE bit will always be on.  
If an exit routine examines \$LMTs to find a particular routine address by name or to find the load module name associated with an address, there are JES2 services to do this that isolate the code from the internal JES2 data structures. In the JES2 environment, the LOCLMOD service obtains the module name associated with an address (including searching for MVS module names). Outside the JES2 environment, FINDLMOD performs the same function. To locate the address of a routine within a JES2 load module, you can use the LOCENTRY service. However, this is only available within the JES2 environment.
3. **Update exit routines that use the \$MODLOAD and \$MODCHK macros, if necessary.** Message processing for the \$MODLOAD and \$MODCHK macros has changed. Before z/OS V1R10, if MESSAGE=NO was coded or defaulted on the macro invocation, an error message could be returned to the caller in a data area pointed to by register 1. The caller could choose to use or ignore the returned message. The message text was in a work area chained off the HCT and did not need to be freed. However, with the introduction of the dynamic exits function, the data area is obtained by the service and must be freed by the caller using \$FREMAIN with SP=0, KEY=1, and LV=128. If the caller does not need the message text, change the macro invocation to specify MESSAGE=YES (message is issued via WTO) or MESSAGE=SUPPRESS (no message is issued or returned).
4. **Enable load modules to support dynamic processing, if necessary.** With the dynamic exits function in z/OS V1R10, JES2 supports loading, refreshing, and deleting installation load modules using an operator command. JES2 will process all dynamic tables (TYPE=DYNAMIC) that exist in the module just as they

were processed before z/OS V1R10 by initialization. Exit routine addresses are cleared when a module is deleted and replaced when a module is refreshed. (\$ADD LOAD will not affect any exit routine address.) However, even with this support, not all load modules will function properly if loaded, deleted, or refreshed after initialization. You must review all your installation load modules (ones loaded with the LOADMOD initialization statements) to determine whether they can be dynamically loaded or deleted and, if they cannot be dynamically loaded or dynamically deleted, decide what needs to be done.

Examples of code that cannot be dynamically loaded or deleted without changes are:

- Modules that have assembled tables that are pointed to by JES2 data areas, for example, a \$UCT data area pointed to by the \$UCT field in the \$HCT. Other tables can be pointed to by other user pointers in JES2 data areas, or the tables can be pointed to by a named token (created using the \$TOKENSR macro).
- Modules that front-end JES2 services.
- Modules that implement their own \$PCE or \$DTE.
- Modules with \$BERTTABS that were not loaded at initialization.
- Modules that define installation \$WSTABS.

Many of these modules will have an exit 24 (post initialization) associated with them. The module was loaded as part of normal initialization processing and then hooked to JES2 data structures using exit 24. Alternatively, they might have code that tests a user field for zero and, if it is zero, they do one-time initialization processing. However, if the initialization code involves obtaining storage and saving that storage address in a JES2 data area, it might not have a problem.

These exit routines might not function properly if they are processed with an operator command. In some cases there will be residual pointers in JES2 data areas that do not get updated. In other cases (such as user defined PCEs or DTEs) there might be no way to stop code that is running in the module. Modules with \$BERTTABS need special considerations. At JES2 initialization, the \$BERTTABS are processed into a table that is used while JES2 is running. Updating this table while JES2 is active could result in unexpected side effects. As a result, JES2 does not process new \$BERTTABS when a module is dynamically loaded. Only \$BERTTABS in modules that were loaded at JES2 initialization (including hot starts) are used. There is no restriction on refreshing modules with \$BERTTABS except that new \$BERTTABS will not be processed. Be careful with any module that contains \$WSTABS. For performance reasons, the address of the \$WSTAB is stored in various JES2 data areas. These addresses are only updated when the WS= parameter associated with a device is updated. New \$WSTAB entries are not processed until the WS= values on any device that use the new work selection criteria are updated (in general by operator command).

There are two ways to address modules that cannot be dynamically updated:

- Prohibit all dynamic processes for these modules. This can be done by setting the DYNAMIC=NO parameter on the \$MODULE macro in the modules.
- Use the new \$\$\$\$LOAD and \$\$\$\$DEL service to perform the initialization processing normally done by exit 24. This generally eliminates the need for exit 24 even for the case when the module is loaded by initialization processing. The \$\$\$\$LOAD and \$\$\$\$DEL routines can also be used to dynamically create and detach PCEs and DTEs. Furthermore, for the case where you want new code loaded but you also want the table that exists in the old module to remain, a function in \$\$\$\$DEL prevents JES2 from ever deleting the load module.

### **Update automation that handles \$D NODE, \$D EXIT, and \$D LOADmod command responses** **(Required-IF, as of OA21346, and as of R10)**

*Required if you have automation that examines these command responses.*

**For \$D NODE:** Before z/OS V1R10, the command response to \$D NODE would return all parameters. Beginning with z/OS V1R10, the command response is changed to display only an abbreviated list of parameters. For the complete list, you can issue the \$D NODE, LONG command. For more details about the parameters in the abbreviated list, see the description of the LONG parameter on the \$D NODE command.

**For \$D EXIT:** Before z/OS V1R10 (and APAR OA21346 on z/OS V1R9 and z/OS V1R8), the command response to \$D EXIT would return all parameters. Beginning with z/OS V1R10, the command response is changed to display only an abbreviated list of parameters. For the complete list, you can issue the \$D EXIT, LONG command. For more details about the parameters in the abbreviated list of \$D EXIT command parameters, see the description of the LONG parameter on the \$D EXIT command.

**For \$D LOADmod:** Before z/OS V1R10 (and APAR OA21346 on z/OS V1R9 and z/OS V1R8), the command response to \$D LOADmod would return all parameters. Beginning with z/OS V1R10, the command response is changed to display only an abbreviated list of parameters. For the complete list, you can issue the \$D

LOADmod, LONG command. For more details about the parameters in the abbreviated list of \$D LOADmod command parameters, see the description of the LONG parameter on the \$D LOADmod command.

**Migration action:** Review and update automation that examines these commands responses.

### **Update JES2 exit routines that examine the \$JCT (Required-IF, as of R9)**

*Required if you have any exit routines that examine the \$JCT eye-catcher field.*

Before z/OS V1R9 JES2, the field name for the \$JCT eye-catcher was JCTID. Starting with z/OS V1R9 JES2, the field name is JCTIDENT. If you have any exit routines that examine the \$JCT eye-catcher, change them to use the new field name.



### **Update JES2 exit 8 routines (Required-IF, as of R9)**

*Required if you use exit 8 and that exit routine examines the CBMTTR field.*

Before z/OS V1R9 JES2, exit 8 used the MTTR parameter to specify a track address. Starting with z/OS V1R9 JES2, exit 8 uses the MQTR parameter instead. If you have exit routines that examine field CBMTTR, change them to examine field CBMQTR instead of field CBMTTR.



### **Update JES2 exit 31 routines (Required-IF, as of R9)**

*Required if you use exit 31.*

Before z/OS V1R9 JES2, there was no mapping macro for the exit 31 parameter list. There was just an area pointed to by a register, with the contents at all applicable offsets described in the documentation. Starting with z/OS V1R9 JES2, exit 31 uses the \$XPL data area instead. If you use exit 31, you will have to change your exit routine.



### **Update JES2 exit 42 routines (Required-IF, as of R9)**

*Required if you use exit 42.*

To accommodate unauthorized SSI callers, the following changes have been made to the notify user exit (exit 42) to provide more input/output fields in its \$XPL data area:

- The pointer to the SSNU control block has been renamed to alert programmers about the changes.
- Additional fields are provided in the \$XPL to allow an exit to obtain functionality similar to that provided by the SSNU without having to reference the SSNU when modifying the notify user message. If an exit still must reference data in the SSNU, the SSI caller's key is provided.
- Condition indicator X042NOST in condition byte X042COND has been deleted. When storage cannot be obtained, the SSI no longer calls exit 42. Instead, it returns with SSNUERR set in the SSOBRETN field.



### **Update JES2 exit 45 routines (Required-IF, as of R9)**


*Required if you use exit 45 and it references any IAZSSSF fields.*

To accommodate unauthorized SSI callers, input and output to the SJF services user exit (exit 45) has been updated as follows:

- The SSI caller's storage key is in the X045CKEY field so that the exit can correctly reference SSSF fields if necessary.
- Additional \$XPL fields have been added, which correspond to SSSF fields. The benefit of using these new fields instead of the SSSF fields is that the new fields are in the JES key, so no keyed references are necessary.
- The address of the SSSF control block has been renamed to X045SSFA to alert programmers about the changes made to the \$XPL.
- You can influence the type of security authorization checking that is to be done by setting flag byte X045FLG1 as desired. Note, however, that if the exit is entered as a result of an unauthorized SSI 70 call, this value is not honored and X045JSSP is the default.
- Condition indicator X045NOST in condition byte X045COND has been deleted. When storage cannot be obtained, the SSI no longer calls exit 45. Instead, it returns with SSSFNOST set in the SSOBRETN field.

## **JES2 Migration Actions Post-First IPL**

<none>

z/OS 1.10: Migration Part 3	
<b>JES3 Migration Actions for z/OS R10 from z/OS R8</b>	
<b>Migration Actions Pre-First IPL:</b>	
<ul style="list-style-type: none"> <li>• <b>Update exit IATUX63 (Required-IF, as of R10)</b> <ul style="list-style-type: none"> <li>• Before R10: information strings returned by SSI 54 (the “Who Am I” SSI) were built during JES3 initialization.</li> <li>• As of R10: They are also built during CONNECT processing.</li> <li>• IATUX63, the exit that builds the installation-defined SSI 54 string, is likewise called during CONNECT processing. Decide when this exit is called during CONNECT processing if you want to: do nothing, create a new installation-defined string, or update your existing installation-defined string.</li> </ul> </li> <li>• <b>Add EXTOSNUM=NO to the OUTSERV initialization statement (Req-IF, as of R9)</b> <ul style="list-style-type: none"> <li>• As of R9, limit of 65,535 spool records for output scheduling element (OSE) structure has been increased. With R9 or higher on the JES3 global, output can be created beyond this limit, however, there are some restrictions. One of which is: the job's output cannot be processed by a JES3 global on a lower release.</li> <li>• EXTOSNUM=NO keeps the old limit in place, to help if you want to avoid the restrictions.</li> </ul> </li> <li>• <b>Modify exit routines and user modifications to support four-byte OSE sequence numbers (Required-IF, as of R9)</b> <ul style="list-style-type: none"> <li>• A number of fields in JES3 mapping macros have been renamed and expanded from two bytes to four bytes to support four-byte IATYOSE sequence numbers.</li> </ul> </li> </ul>	
15	© 2008 IBM Corporation

### JES3 Migration Actions Between z/OS V1 R8 and z/OS V1 R10

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

### JES3 Migration Actions You Can Do Now

<none>

### JES3 Migration Actions Pre-First IPL

#### Update JES3 exit IATUX63 (Required-IF, as of R10)

*Required if you use exit IATUX63.*

Before z/OS V1R10, information strings returned by SSI 54, also known as the “Who-Am-I” SSI, were built during JES3 initialization. Beginning with z/OS V1R10, the information strings are also built during CONNECT processing. This allows information to be updated when information about the global changes without the local having been restarted. The global information can change in two ways: (1) the global is IPLed and the JES3 release is changed during this IPL, and (2) a Dynamic System Interchange (DSI) is performed, which changes the name of the global and possibly the release. IATUX63, the exit that builds the installation-defined SSI 54 string, is likewise called during CONNECT processing.

**Migration action:** Update your IATUX63 code to test the flag YUX63CON. This flag is set during connect processing. When this bit is set, you have the following options:

- Return without changing anything.
- Free the storage for the old installation-defined string and create a new one.
- Update your existing installation-defined string without changing its location or length.

#### Check extended status applications for the need for testing STSTHTCP and STSTHBDT (Req-IF, as of R10)

*Required if you have an application that looks specifically at the existing STSTHSTA flags.*

Before z/OS V1R10 JES3, terse and verbose output information returned by SSI 80 (extended status) included output on the HOLD and WTR queues. Beginning with z/OS V1R10, output on the BDT and WTR queues is returned in addition to output on the HOLD and WTR queues. The flags that JES3 sets to indicate the TCP and BDT queues are STSTHTCP and STSTHBDT, respectively, in the flag byte STSTHSTA.

**Migration action:** Look at your application code that uses the extended status interface (SSOBFUNC=SSOBESTA) and answer the following questions to determine what to do:

1. Do you use the STATTYPE of STATOUTT or STATOUTV?
  - No: No action needed.
  - Yes: Proceed to the next question.
2. Do you examine flag byte STSTHSTA?
  - No: No action needed.
  - Yes: Proceed to the next question.
3. Do you test the hold type flags STSTHOPR, STSTHUSR, STSTHSYS, STSTHTSO, and STSTHXWT within STSTHSTA and take some actions if none of them are set?
  - No: No action needed.
  - Yes: Determine what action your code takes in this case and evaluate the need for adding code to test STSTHTCP and STSTHBDT.

**Delete the EDIT keyword from the CONSTD statement (Required-IF, as of R10)**

*Required if you have any automated procedures or disk readers that specify the backspace, line delete, or escape character.*

In z/OS V1R10 JES3, the CONSTD EDIT= keyword is no longer supported. Message IAT3256 EDIT KEYWORD IGNORED, NO LONGER SUPPORTED is issued. JES3 initialization continues.

The characters previously defined as backspace, line delete, and escape are no longer special characters and are now treated by JES3 console services as literal characters. The MVS command delimiter from the INIT CMDDELIM() parameter in SYS1.PARMLIB(CONSOLxx) continues to be honored by JES3.

**Migration action:**

1. Remove the EDIT= keyword from the CONSTD statement to clean up the warning messages.
2. Remove automation or disk reader commands if they are using the backspace, line delete, or escape character. For example, before z/OS V1R10 JES3, if you had the pound sign defined as the backspace character and an automation routine performing an action on a job with a pound sign as part of its name, you had to enter the escape character before the pound sign. Otherwise, the command would treat the pound sign as a backspace and change the command to a job name that you did not want the command to operate on. With z/OS V1R10 JES3, because the pound sign is interpreted literally, you can remove the escape character.

**Modify exit routines and user modifications to support four-byte OSE sequence numbers (Req-IF, as of R9)**

*Required if have any exit routines or user modifications that use any of the affected fields.*

A number of fields in JES3 mapping macros have been renamed and expanded from two bytes to four bytes to support four-byte IATYOSE sequence numbers. You need to modify any exit routines and user modifications that use any of these fields to use the new fields instead.

**Migration action:** Follow the steps to take in *z/OS Migration book*.

1. Examine your user exits and modifications to look for any references to fields that represent an OSE sequence number (sometimes identified as an OSE buffer number).
2. Look at the mapping macro that defines the OSE sequence number field to determine the new name and format.
3. Modify your code to use the new field.

**Add EXTOSENUM=NO to the OUTSERV initialization statement (Required-IF, as of R9)**

*Required if you want to keep the old limit of 65,535 IATYOSE spool records in place so that jobs cannot create output that cannot be processed on earlier releases.*

In z/OS V1R9 JES3, the limit of 65 535 spool records for the output scheduling element (IATYOSE) structure has been increased. Before z/OS V1R9, you received abend DM674 when a job reached the limit of 65 535 IATYOSE spool records. The job could not create more output and had to be restarted.

With z/OS V1R9 running on the JES3 global processor, output can be created beyond this limit. However, there are some restrictions:

- The job's output cannot be processed by a JES3 global on a lower release. A DSI to a processor running a lower release or a fallback to a lower release causes message IAT7604 to be issued, indicating that the job has been removed from output service. You must dump and restore the job using the dump job facility before its output can be processed.
- The output for the job that was created after the limit was exceeded cannot be processed by a process SYSOUT (PSO) application that is running on a JES3 local at an earlier release. Examples of PSO applications include the IBM-supplied external writer, the TSO OUTPUT command, and some vendor printing products. This limitation does not apply to applications using the SYSOUT application programming interface (SAPI).

A new keyword on the OUTSERV initialization statement, EXTOSENUM, controls whether jobs are allowed to create output after the old limit has been reached. Specifying EXTOSENUM=NO keeps the old limit in place, which ensures that jobs cannot create output that cannot be processed on earlier releases.

**Migration action:** To ensure that all output can be processed on earlier-level JES3 systems, specify EXTOSENUM=NO on the OUTSERV initialization statement that is executed the first time a z/OS V1R9 JES3 global processor is IPLed. (The default is EXTOSENUM=YES.) EXTOSENUM=NO keeps the old limit of 65 535 IATYOSE spool records in place.

**Note:** Specifying EXTOSENUM=NO has no effect on output that has already been created, nor will it prevent additional output scheduling elements from being added to spool records that have already been created.

If you have PSO applications running on earlier-level JES3 local processors, specify EXTOSENUM=NO on the OUTSERV statement until your entire complex has been migrated to z/OS V1R9. Likewise, you can do this to make sure that jobs do not create output that cannot be processed by earlier-level globals in the event of a fallback.

### **JES3 Migration Actions Post-First IPL**

<none>



z/OS 1.10: Migration Part 3

## Language Environment Migration Actions for z/OS R10 from z/OS R8

### Migration Actions Pre-First IPL:

- **Determine the impact of added and changed runtime options (Required-IF, as of R10)**
  - As of R10, and rolled back: HEAPPOOLS option was added to the CELQDOPT options group (was already valid in CEEDOPT and CEECOPT) As of R9: CEEDUMP option was added, to control the processing of the dump report CEEDUMP.
  - If you use CEEPRMxx: **no migration action** since options added don't change default behavior!
  - If you use assembler language modules: update and reinstall your usermods.
- **Use CEEPRMxx's CEEROPT instead of making copies of Language Environment initialization routines (Required-IF, as of R10)**
  - Many installations have copied LE init routines (making updates to the copies for specific runtime options), and STEPLIB'd to the copies for different applications. **This method will no longer work as of R10!**
  - Now in CEEPRMxx, CEEROPT(ALL) indicates that CEEROPT should be loaded for all application execution environments, and that the options found in that module should be used. **This method does not required any updates to applications!**
    1. Turn on CEEROPT(ALL) in CEEPRMxx, which is not the default.
    2. Create a separate CEEROPT module with your customization similar to how you made the changes before, and put in a data set. Use sample CEEWROPT.
    3. STEPLIB to that CEEROPT data set you customized for your application.
    4. Ensure you remove, and longer have the LE init modules in the STEPLIB concat.

16 © 2008 IBM Corporation

### Language Environment Migration Actions Between z/OS V1 R8 and z/OS V1 R10

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

#### Update the CICS CSD based on the newest CEECCSD sample (Required)

Each release, Language Environment adds or deletes load modules in the CICS system definition (CSD) file. Thus, you should update the file each release using the program definitions found in member CEECCSD and, if using CICS Transaction Server (TS) for z/OS V3 (5655-M15), in member CEECCSDX..

**Migration action:** Update the CSD file using the program definitions in member CEECCSD (and member CEECCSDX if using CICS TS V3) found in the *hlq.SCEESAMP* data set.

**Note:**

1. The group containing the Language Environment runtime routines must be in the group list used during CICS startup.
2. The size of the CEECCSD decreased dramatically in z/OS V1R9. The decrease was intentional; it occurred because iconv() started using the Unicode converters instead of the converters supplied with Language Environment.

#### Review Language Environment load modules in LPA (Required-IF)

*Required if you need to make modules accessible through the link pack area (LPA).*

Each release you must update the Language Environment load modules that you make accessible through the link pack area (LPA). In addition, each release you should review your list of load modules in the LPA to determine if it's still suitable.

**Migration action:** Review Language Environment load modules in the LPA.

- To move load modules into the LPA, use sample members are provided in the CEE.SCEESAMP data set.
- To see which modules are eligible for the LPA, refer to *z/OS Language Environment Customization*. The modules listed there can be put in the LPA or extended LPA (ELPA) depending on their RMODE value:
- If you are considering placing the modules listed in *z/OS Language Environment Customization* in the LPA or the ELPA, then IBM highly recommends that you place the SCEELPA data set in the LPA list (LPALST xx). SCEELPA contains Language Environment load modules that are reentrant, that reside above the 16 MB line, and that are heavily used by z/OS.
- In *z/OS Language Environment Customization* you will also see tables of modules eligible for the LPA and the ELPA above and beyond what is found in the SCEELPA data set. You will need to use the dynamic LPA or MLPA approach to move these modules into the LPA or ELPA. You do not need to include recommended modules if they contain functions your installation does not use. Language Environment modules not listed in these tables can be moved into the LPA or ELPA at your discretion.



### **Determine the impact of added and changed run-time options (Required-IF, as of R10)**

*Required if you are using assembler modules to specify your installation-wide run-time options (CEEDOPT, CEECOPT, or CELQDOPT). If you are using CEEPRMxx (introduced in z/OS V1R7) and the default specification for the newly added option is acceptable to you, then no migration action is required.*

Periodically, Language Environment introduces new run-time options, adds new suboptions to existing run-time options, and changes the defaults of run-time options.

In z/OS V1R10:

- One runtime option, HEAPPOOLS, was added to the CELQDOPT options group. (HEAPPOOLS was already valid in both CEEDOPT and CEECOPT.) This support is integrated in z/OS V1R10; it was initially made available on z/OS V1R9 by APAR PK49427 and on z/OS V1R8 and V1R7 by APAR PK57579.
- No suboptions were added and no default changes were made to existing runtime options.

In z/OS V1R9:

- One run-time option, CEEDUMP, was added. CEEDUMP is used to specify options to control the processing of the Language Environment dump report CEEDUMP.
- No suboptions were added and no default changes were made to existing run-time options.

Because of these changes, you might have to perform migration actions.

**Migration action:**

- If you are using CEEPRMxx to specify your installation-wide runtime options:
  - The new z/OS V1R10 runtime option HEAPPOOLS that was added to CELQDOPT does not change the default behavior. You have no migration action.
  - The new z/OS V1R9 CEEDUMP runtime option does not change the default behavior. You have no migration action.

**Tip:** Use IBM Health Checker for z/OS to check that default Language Environment runtime options are set within a CEEPRMxx parmlib member. The check is named CEE\_USING\_LE\_PARMLIB.

- If you are using the assembler modules to specify your installation-wide runtime options, compare your existing source for the installation-wide runtime options CSECT, CEEDOPT (non-CICS environment), CEECOPT (CICS environment), or CELQDOPT (AMODE 64) with the new samples in *hlq.SCEESAMP* to determine whether you need to change the defaults. Then, update (and reinstall) your usermods with the following new options:
  - The new z/OS V1R10 runtime option HEAPPOOLS that was added to CELQDOPT. But note that this function, which is integrated in z/OS V1R10, was rolled back to prior releases (APAR PK49427 on z/OS V1R9 and APAR PK57579 on z/OS V1R8 and V1R7), so you might have already updated this usermod.
  - The new z/OS V1R9 CEEDUMP runtime option.



### **Use CEEPRMxx's CEEROPT instead of making copies of Language Environment initialization modules (Required-IF, as of R10)**

*Required if you have copied any Language Environment initialization module (CEEBINIT, CEEBINSS, CEEPIPI, or CEEBPICI) for the purpose of manually linking in a separate installation-wide runtime option CSECT CEEDOPT. **Note:** If you have used **only** SMP/E to maintain a customized CEEDOPT CSECT (by using a sample usermod CEEWD01), this migration action does not apply to you. Using the CEEWD01 usermod to specify your runtime options in CEEDOPT will still result in Language*

*Environment correctly using your specified runtime options. This migration action is applicable only if you have linked CEEDOPT outside of SMP/E into separate libraries that were not SMP/E-controlled (and were STEPLIBed to by the applications).*

Many installations have made copies of Language Environment's initialization modules CEEBINIT, CEEBINSS, CEEPIPI, and CEEBPICI (updating each copy with specific runtime options specified in the CEEDOPT CSECT), and then used STEPLIB to access those modules for different applications. This method allows applications to individually have their own runtime options while not requiring any updates to the applications themselves. However, this method has the disadvantages of being difficult to manage when service updates are made in Language Environment, having many copies of Language Environment modules to keep track of, and creating synchronization problems between the initialization modules and other Language Environment modules, which are hard to debug.


This method of customizing Language Environment's initialization modules no longer works as of z/OS V1R10. As of z/OS V1R10, the CEEDOPT CSECT is no longer part of the Language Environment's initialization modules CEEBINIT, CEEBINSS, CEEPIPI, and CEEBPICI. Therefore, any customization for runtime options added to these initialization modules will not be used by Language Environment for the application's runtime environment. To easily allow applications to continue to use customized runtime options, and to eliminate the need to make separate copies with updates to the initialization modules, support was added to the CEEPRMxx parmlib member. The CEEPRMxx CEEROPT statement (for non-AMODE 64 applications and CELQROPT for AMODE 64 applications) controls this support and CEEROPT(ALL) indicates to Language Environment that CEEROPT should be loaded for all application execution environments, and that the options found in that module should be used. Using CEEPRMxx's CEEROPT statement does not require any updates to the applications themselves. Because of these changes, you might have to perform migration actions.

**Migration action:**

1. CEEROPT processing can only be specified in your CEEPRMxx parmlib member (as CEEROPT is not an option in the CEEDOPT assembler language module). Turn on CEEROPT processing for all non-AMODE 64 environments by setting the keyword CEEROPT(ALL) in your active CEEPRMxx member. (CEEROPT(COMPAT) is the default, which indicates CEEROPT is to be loaded and used only for CICS and LRR environments. This is the behavior prior to z/OS V1R10.)
2. Create a separate CEEROPT module with runtime option overrides similar to those you are currently manually linking into a copy of one of the Language Environment initialization modules. You can use sample CEE.SCEESAMP(CEEWROPT) for the assembly and link-edit. Place each CEEROPT module in a separate data set.  
**Note:** You only need to set those runtime options that are different from the ones set in your installation defaults or system-wide defaults.
3. Place this CEEROPT data set into the STEPLIB concatenation for the application.
4. Review the other modules you have in the STEPLIB concatenation. If you had any Language Environment initialization modules in the concatenation for the sole purpose of customizing the CEEDOPT CSECT, you must remove those modules from the concatenation.

**Language Environment Migration Actions Post-First IPL**

<none>

z/OS 1.10: Migration Part 3


## SDSF Migration Actions for z/OS R10 from z/OS R8

### Migration Actions You Can Do Now:

- **Use an alternative to modify SDSF source modules and macros (Required-IF, as of R9)**
  - Many source modules and macros are no longer distributed. Review your modifications for possible alternatives.

### Migration Actions Pre-First IPL:

- **Use dynamic statements for ISFPARMS to avoid reassembly (Recommended)**
  - ISFPARMS dynamic statements require no migration action. If using assembler macros for ISFPARMS, then 1) convert to dynamic statements © or 2) reassemble each release.
- **Use the new default for column options in REXX execs (Recommended)**
  - Before R10: when SDSF panel was accessed in a REXX exec the default was to include only the columns from the primary field list. As of R10: the default is to include columns from both the primary and alternate field lists.
  - Decide if you want only the primary or alternate field lists accessed (or use the new behavior). Can specify new option to get prior behavior.
- **Find sample in new JES2 feature FMID (Required-IF, as of R10)**
  - Assembly-related samples have moved from SISFJCL/SISFSRC to SISFJCL1 / SISFSRC1
- **Use SAF instead of sample usermods ISFUM01 and ISFUM08 (Recommended)**
  - Sample ISFUM01 and ISFUM08 are removed from SDSF. Recommended to use SAF instead. No change to the function of SDSF security.

17
© 2008 IBM Corporation

### SDSF Migration Actions Between z/OS V1 R8 and z/OS V1 R10

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*.

#### SDSF Migration Actions Post-First IPL

##### **Use an alternative to modifying SDSF source modules and macros (Required-IF, as of R9)**

*Required if you have source modifications.*

Although modifying SDSF source modules and macros is not supported, some installations might have made such modifications. (The supported method to modify SDSF is to implement a user exit routine.) Beginning with z/OS V1R9, many SDSF source modules and macros that were distributed previously are no longer distributed. You should review any user modifications you have and assess alternatives.

**Migration action:** Review your modifications for possible alternatives:

- The new PROPLIST and PROPERTY statements in ISFPARMS
- User exit routines.

If you cannot use either of these to replace your modifications, you may want to submit a requirement through your IBM representative

#### SDSF Migration Actions Post-First IPL

##### **Reassemble user exit routines (Required-IF, as of R9)**

*Required if user exit routines are in use.*

If you have written user exit routines, review them to ensure they are still appropriate for the current environment. All user exit routines must be reassembled with the z/OS V1R10 level of the SDSF macro library.

**Migration action:** Reassemble user exit routines with the z/OS V1R10 level of the SDSF macro library.

**Note:** In z/OS V1R9 SDSF, a new target data set, ISF.SISFMAC, was introduced to hold macro parts that were moved from ISF.SISFSRC.



##### **Use dynamic statements for ISFPARMS to avoid reassembly (Recommended)**

*Recommended to avoid the migration action of reassembling your customized ISFPARMS for each z/OS release. (If you do not use dynamic statements for ISFPARMS, reassembly of your customized ISFPARMS is required on each release.)*

ISFPARMS in SDSF is used for specifying global options, the format of panels, and security for SDSF functions. SDSF provides two alternatives for ISFPARMS:

- Assembler macros that you define, assemble, and then link into the SDSF load library. This is the original format for defining ISFPARMS and it continues to be supported for compatibility.
- Dynamic statements, which are in parmlib member ISFPRMxx. Dynamic statements are the recommended format. They are easier to code and are more dynamic than the assembler macros; they can be updated without reassembling or link-editing. The statements are processed by an SDSF server, which is controlled by MVS operator commands.

**Migration action :** If you are already using dynamic statements for ISFPARMS, there is no migration action to perform.

If you are using assembler macros for ISFPARMS, do one of the following:

- Convert your existing ISFPARMS to dynamic statements by using a conversion utility that you invoke with the ISFACP command.
- Reassemble your customized ISFPARMS for use with z/OS V1R10. Reassembly must be done whenever you change your z/OS release. Before reassembling ISFPARMS, you might want to update it for new function. The assembler ISFPARMS cannot be shared with any other release of SDSF. Only use ISFPARMS for the release on which it is assembled.

**Note:** Sample job ISFPARME has been removed from the samples supplied with SDSF. This job contained SMP/E control statements to receive the sample assembler macro ISFPARMS as a user modification (USERMOD).

**Tip:** Use check SDSF\_ISFPARMS\_IN\_USE with IBM Health Checker for z/OS to verify that SDSF dynamic statements in ISFPRMxx are being used rather than the assembler macros. If the check determines that the assembler macro ISFPARMS is in use instead, and that it has been modified, the check generates an exception. If the assembler macro ISFPARMS is in use but it has not been modified, so that all defaults are in effect, the check does not generate an exception.

SDSF registers this check with the IBM Health Checker for z/OS infrastructure when the SDSF server address space is initialized. However, one of the items this check verifies is that the SDSF server itself is in use, so you have to manually add this check (particularly if you do not use the SDSF server) so that the IBM Health Checker for z/OS infrastructure will invoke the check. To add the check, put the following statement in your PROGxx parmlib member: EXIT ADD EXITNAME(HZSADDCHECK) MODNAME(ISFHCADC).

### **Stage SDSF with JES2 (Required-IF, as of R9)**

*Required if you use SDSF with JES2.*

The staging, or postponement, of migration to the new level of JES2 when you install a new level of z/OS can help you minimize the number of simultaneous changes that you make in an upgrade. When you stage the migration of JES2, you initially use your current, supported JES2 on your new z/OS system. Later you migrate to the new level of JES2. Before z/OS V1R9, you had to use the latest level of SDSF with your currently installed JES2. For example, when migrating to z/OS V1R8, you needed to migrate to the z/OS V1R8 level of SDSF, even if you postponed migrating to the z/OS V1R8 level of JES2.

Starting with z/OS V1R9, SDSF and JES2 are “coupled” in the sense that the release level of your SDSF and JES2 must be the same. This means that:

- When you migrate to the z/OS V1R10 level of JES2, you must migrate to the z/OS V1R10 level of SDSF.
- You can stage the migration of SDSF as follows:
  - If you are running the z/OS V1R9 level of JES2 with z/OS V1R10, you must run the z/OS V1R9 level of SDSF.
  - If you are running the z/OS V1R8 level of JES2 with z/OS V1R10, you must run the z/OS V1R8 level of SDSF.

A by-product of this change is a change to the UCLIN samples that are provided. Previously, to support the staging of JES2, SDSF provided UCLIN samples for each of the supported levels of JES2. As z/OS V1R9, SDSF only provides one UCLIN sample, which is for the level of JES2 that SDSF runs with. The UCLIN members for the lower levels of JES2 are no longer provided.

**Migration action :** Migrate SDSF to a level that is appropriate for the level of JES2.

### **SDSF Migration Actions Post-First IPL**

#### **Use the new default for column options in REXX execs (Recommended, as of R10)**

*Recommended because the new default eliminates the need to be aware of which field list a column is in and the need to specify the ALTERNATE option to include columns that are in the alternate list.*

Before z/OS V1R10, when an SDSF panel was accessed in a REXX exec, the default was to include only the columns from the primary field list. Beginning with z/OS V1R10, the default is to include columns from both the primary and alternate field lists.

**Migration action:**

- If you want the panels accessed in your execs to include columns from both the primary and alternate field lists, which is the new behavior, do nothing. (This assumes you are not currently using the ALTERNATE or ALTERNATE2 options, which continue to limit the columns to those in the alternate field lists.)
- If you want the panels accessed in your execs to include only the columns in the primary field lists, which is the prior behavior, update the ISFEXEC or ISFACT commands to use the new PRIMARY and PRIMARY2 options. If you want only the columns that are in the primary field list, add a PRIMARY option to the command that accesses the panel. If the panel is a secondary panel (accessed from another panel), use the PRIMARY2 option.
- If you want the panels accessed in your execs to include only the columns in the alternate field lists, and the ISFEXEC or ISFACT commands already have the ALTERNATE and ALTERNATE2 options, do nothing.

**Update customized field lists and REXX execs for a column name change on the Enclaves panel (Required-IF, as of R10)**

*Required if you have customized field lists or REXX execs that use the old column name.*

The column on the Enclaves panel that was previously named SCOPE is now named ESCOPE. (The name is how the column is known to SDSF; it is different from the column title that is displayed on the panel.) If you have a customized field list in ISFPARMS for the Enclaves panel that includes the column name SCOPE, update the field list by changing the column name to ESCOPE. Similarly, if you have REXX execs that use the column name SCOPE, update them to use ESCOPE.

**Migration action:** Update the field lists and REXX execs to use the new column name.

**Find samples in the new JES2 feature FMID (Required-IF, as of R10)**

*Required if you use sample job ISFISUCL or ISFASMP, or have SMP/E sysmods that modify ISFPARMS.*

Three samples supplied with SDSF are now shipped with SDSF's JES2 feature FMID (JJE775S) and are located in new data sets.

Sample	Description	Data set in z/OS V1R10	Data set in prior releases
ISFISUCL	Reassemble SDSF automatically when JES2 macros change	ISF.SISFJCL1	ISF.SISFJCL
ISFASMP	Reassemble SDSF	ISF.SISFJCL1	ISF.SISFJCL
ISFPARMS	Provides sample assembler macro for customization and security for SDSF in the JES2 environment	ISF.SISFSRC1	ISF.SISFSRC

**Migration action:**

- Use the samples, if needed, from their new locations.
- If you have an SMP/E usermod that specifies modifications to assembler macro ISFPARMS, change the usermod to indicate that module ISFPARMS is now owned by the SDSF JES2 feature FMID (JJE775S) and not the base SDSF FMID (HQX7750). The correct SMP/E syntax is ++VER(Z038) FMID(JJE775S), not ++VER(Z038) FMID(HQX7750). Note that you can eliminate such a usermod; see "Use dynamic statements for ISFPARMS to avoid reassembly".

**Use SAF instead of sample usermods ISFUM01 and ISFUM08 (Recommended, as of R10)**

*Recommended if you have previously evaluated the samples and planned to implement one because SAF provides better granularity and auditability, and is more dynamic than using ISFPARMS for SDSF security.*

Sample user modifications ISFUM01 and ISFUM08 have been removed from the samples supplied with SDSF. The security provided by these samples is available through the system authorization facility (SAF). There is no change to the function of SDSF security with the removal of these samples. However, if you had previously evaluated the samples and planned to implement one, you should consider using SAF instead. Using SAF for SDSF security provides better granularity and auditability, and is more dynamic, than using ISFPARMS.

**Migration action:** Use SAF to implement the function in ISFUM01 and ISFUM08.

**Review batch jobs or procedures that use the O command with the @ parameter (Req-IF, as of R9)**

*Required if you have batch jobs or procedures that use this command syntax and expect the prior behaviors.*

The special character, @, on the O command, requests output destined for another node. Before z/OS V1R9 SDSF, if you combined @ with output classes on the O command, you saw all output destined for another node (regardless of class) and all output in the specified classes. With z/OS V1R9 (and later), combining the @ with output classes displays only output that is both destined for another node and in the specified classes.

**Migration action:** Review your batch jobs and procedures. If you use the O command with the @ and other output classes, and expect the result to be all output destined for another node, you need to make adjustments. You might want to issue the O command twice, once with the @ parameter and once with the specific output classes.

**Customize SDSF if you do not want zAAP and zIIP columns on the DA panel to always be shown (Required-IF, as of R9)**

*Required if you do not want zAAP and zIIP columns on the DA panel to always be shown.*


Before z/OS V1R9 SDSF, columns that show information about the System z Application Assist Processor (zAAP) and System z9 Integrated Information Processor (zIIP) were dynamically included on or omitted from the DA panel based on whether the associated special processor was configured for a system that was within the scope of the systems being shown on the panel. With z/OS V1R9 (and later) SDSF, the columns are always shown. You can customize SDSF to act as it did in prior releases and show the columns only when the special processor is present.

**Migration action:** Customize SDSF to specify that the columns showing information about the zAAP and zIIP should be included on the DA panel only when the appropriate special processor is present. Use either:



- The new PROPLIST and PROPERTY statements in ISFPARMS. The property is Panel.DA.DynamicZAAPCols.
- Flag UPROFLG1.UPRO1DYZ in a user exit routine

z/OS 1.10: Migration Part 3

**z/OS UNIX Migration Actions for z/OS R10 from z/OS R8**



**Migration Actions You Can Do Now:**

- 
**• Migrate from HFS to zFS file systems (Recommended)**
  - HFS has been stabilized, and zFS is strategic.
  - Convert you file systems from HFS to zFS.
- 
**• Use ServerPac (during install, for the shipped file systems), BPXWH2Z (for after install, and other file systems you have), and look at R10's F OMVS,NEWROOT= support for the sysplex root.**


**Migration Actions Pre-First IPL:**

- Update z/OS UNIX configuration files (Required-IF, as of R10)**
  - /etc/rc was changed to clear man page cache for R10. /etc/magic was changed for R9. /etc/ohelp.ENU changes every release.
- Increase dump space to accommodate dumps of the zFS address space (Required-IF, as of R9)**
  - Before R10: system dumps didn't include dumps from the zFS address space.
  - As of R9: zFS data is captured, as long as the address space is active and the dump cause is related to a file system problem.
  - Can be larger, may need to increase your dump data sets.

18 © 2008 IBM Corporation

z/OS 1.10: Migration Part 3

**z/OS UNIX Migration Actions for z/OS R10 from z/OS R8**



**Migration Actions Post-First IPL:**

- Use the Shell and Utilities version of the submit command (Recommended, as of R10)**
  - Before R10: submit command was a Tools and Toys download.
  - As of R10: it is part of the Shell and Utilities support in z/OS.
  - Start using the R10 submit command, remove the "old" submit.
- Update In invocations to accommodate changes made to the In shell command (Required-IF, as of R10)**
  - Before R10: on the In command, the -e option with -s, -r, or -R had invalid results.
  - As of R10: -e is mutually exclusive with -s, -r, or -R. If you try to specify -e with those options or with an existing target directory, you will receive an error.
  - Remove any -e option from In invocations, so it is not used with -s, -r, or -R.

19 © 2008 IBM Corporation



## z/OS UNIX Migration Actions Between z/OS V1 R8 and z/OS V1 R10

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



#### **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.

#### **Migration action:**

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

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

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

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

You can use the JCL sample ISPBTCH in SYS1.SAMPLIB to invoke BPXWH2Z as an ISPF batch job. Before you run the job, be sure to read the Notes section. When you run BPXWH2Z on your z/OS V1R10 system, it uses the z/OS V1R10 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 V1R10. You can now 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 *z/OS Migration*.

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.

### **Use the new default of 10 for the automount delay keyword (Recommended, as of R9)**

*Recommended to prevent the system from hanging as it continually tries to unmount file systems and fails.*

Before z/OS V1R9, the default for the delay keyword for the **automount** shell command was 0. In z/OS V1R9, the default has been changed to 10. This change brings the default in line with the already-established best practice. A delay value lower than 10 minutes can cause the system to hang as it continually tries to unmount file systems and fails.

**Migration action:** If you are using the default of 0 on your pre-z/OS V1R9 system, determine whether the default of 10 is acceptable on your z/OS V1R9 system. If it is, you do not need to update the delay field in the MapName file. However, if the default of 10 is not acceptable, you can override the value by changing the 10 to 0.

**Tip:** If your system has IBM Health Checker for z/OS set up, the USS\_AUTOMOUNT\_DELAY check looks at the delay configuration values in a sysplex to ensure they are at least 10 minutes.

### **Update BPXPRMxx if more than 50 MKDIR keywords are specified (Required-IF, as of R9)**

*Required if more than 50 MKDIR keywords are specified on a ROOT or MOUNT statement in BPXPRMxx.*

Before z/OS V1R9, you could specify more than 50 MKDIR keywords on each ROOT or MOUNT statement in the BPXPRMxx member of SYS1.PARMLIB. In z/OS V1R9, the maximum number of MKDIR keywords allowed for each ROOT or MOUNT statement is 50. If you specify more than 50, syntax errors occur and initialization stops when parsing completes. The syntax error message that is issued is BPXI075I ERROR IN PARMLIB MEMBER=member ON LINE xxxxx. TOO MANY MKDIR STATEMENTS.

**Migration action:** Make sure that no more than 50 MKDIR keywords are specified on ROOT and MOUNT statements in BPXPRMxx

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



### **Update z/OS UNIX configuration files changed by IBM (Required)**

Some utilities provided by z/OS UNIX require the use of certain configuration files. You are responsible for providing these files if you expect to use the utilities. IBM provides default configuration files as samples in the /samples directory. Before the first use of any of these utilities, you must copy these IBM-provided samples to the /etc directory (in most cases). You can further customize these files to include installation-dependent information. An example is setting up the /etc/rc file by copying the sample file from /samples/rc to /etc/rc and then customizing it for the installation.

Whenever IBM ships a changed sample configuration file, you must copy the changed sample to its target location and, if you customized the previous version of the sample, incorporate the customization into the new version.

**Migration action:** For each of the files listed in the table below, copy the IBM-provided sample file to the listed target location. If you added installation-dependent customization to any of the IBM-provided files, make the same changes in the new versions of the files.

Utility	IBM-provided sample file	Target location	What changed and when
file	/samples/magic	/etc/magic	In z/OS V1R9, the system default magic file was changed to support the SUSv3 format. The V1R9 file command supports old format magic files without change. The new /samples/magic file has been changed to comply with SUSv3.
initialization	/samples/rc	/etc/rc	In z/OS V1R10, the initialization script was changed to clear out the man page cache. This eliminates the migration action "Update the date and time of book files used for man pages".
OHELP	/samples/ohelp.ENU	/etc/ohelp.ENU	Every release, bookshelf and book data set names are updated. See "Install books and a bookshelf for the OHELP command" on page 306.



### **Ensure that UNIX03 is not set in system profiles (Required-IF, as of R9)**

*Required if you want shell commands to maintain compatibility with prior releases.*

Before z/OS V1R9, the `_UNIX03` variable may have been set in system profiles. The variable was introduced in z/OS V1R8 as a means of controlling whether certain shell commands conform to Single UNIX Specification, Version 3 (SUSv3). `_UNIX03=YES` is only needed when an option or behavior conflicts with an existing z/OS implementation, and the SUSv3 behavior is desired. In z/OS V1R9, additional shell commands have behavior conflicts controlled by the `_UNIX03` variable. If `_UNIX03` is not set (or is set to "NO"), then the z/OS V1R9 commands will maintain compatibility with prior releases.

**Migration action:** To maintain shell command compatibility with prior releases, ensure that the `_UNIX03` variable is not set (or is set to "NO") in system-wide profiles or setup scripts, including the following:

- `/etc/profile`
- `/etc/csh.cshrc`
- `/etc/csh.login`
- `/etc/rc`

If you have set `_UNIX03=YES` in a system profile, then changing that setting may affect users. If users want behavior that conforms to SUSv3, they can set `_UNIX03=YES` in their profile files or in specific scripts or command invocations where it is needed.

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



### **Install books and a bookshelf for the OHELP command (Required-IF)**

*Required if you use the OHELP command.*

The TSO/E OHELP command uses the BookManager READ base element to display online reference information about shell commands, TSO/E commands, C functions, callable services, and messages issued by the shell and dbx. Before this online help facility can be used, you must set it up by installing books and a bookshelf.

**Migration action:** Copy the IBM-supplied sample file `/samples/ohelp.ENU` to `/etc/ohelp.ENU`. If you choose not to use the IBM-supplied default file, you can create your own file called `/etc/ohelp.language_id`, where `language_id` is the TSO/E user's primary language code.

### **Use the Shell and Utilities version of the submit command (Recommended, as of R10)**

*Recommended because the new supported version of the submit command is enhanced compared to the Tools and Toys version.*

Before z/OS V1R10, the **submit** command was available from the Tools and Toys section of the z/OS UNIX Web site. As of z/OS V1R10, Shell and Utilities support of the **submit** command has been added. The supported version differs from the Tools and Toys version in a number of ways:

1. Multiple files can be used as input.
2. After the **submit** command completes, a message that indicates success or failure is sent. If the command succeeds, the resulting message contains the job IDs of the submitted jobs.
3. Formatting for submitting jobs from a partitioned data set (PDS) is different. Because the supported **submit** command is now consistent with the other subcommands, the `//` characters are required as a prefix to the PDS name. For example: `"/'full.pds.name(member)'"`
4. The Tools and Toys version of **submit** had the following conventions, which have been changed:
  - `?` was used as a way to display usage but is now treated as a valid file name.
  - `-` was used as a file name but is now used to specify standard input.

**Migration action:**

1. If you are using the Tools and Toys version of the **submit** command, delete it or move it to a location that is not in the system's search path. If you leave it in the search path, it might be picked up instead of the supported version that is in the `/bin` directory.
2. Begin using the supported version in the `/bin` directory.

### **Update cksum and sum invocations (Required-IF, as of R10)**

*Required if any automated tasks rely on cksum or sum exiting when an automatic conversion error occurs.*

Before z/OS V1R10, the **cksum** and **sum** utilities exited if they could not turn off automatic conversion on a file stream. As of z/OS V1R10, if this error is encountered, the file stream is skipped and the remaining files are processed. The message associated with this error is FSUMF182. This change should not affect most users.

**Migration action:** Modify any **sum** or **cksum** invocations (for example, in shell scripts) that rely on the utility exiting when automatic conversion errors occur.

**Update In invocations to accommodate changes made to the In shell command (Req-IF, as of R10)**

Required if `In -e` is used in either of the following ways: 1) in conjunction with the `-s`, `-r`, or `-R` option, or 2) with an existing target directory.

Before z/OS V1R10, if you specified the `-e` option with the `-s`, `-r`, or `R` option when issuing the `In` shell command, the results were not valid.

- If `source` was a directory containing symbolic links or external links and `target` was an existing directory, and you issued `In -e` with `-s`, `-r`, or `-R`, one of the following events occurred:
  - Message FSUM6245 was issued.
  - One or more incorrect symbolic links was created.
  - One or more incorrect external links was created.
- If `source` was a file or a PDS or PDSE member, `-e` was specified with `-s`, and `target` was not a directory, `In` created a symbolic link.
- If `-e` was specified with an existing target directory, `In` created an incorrect external link.

As of z/OS V1R10, the `-e` option is mutually exclusive with the `-s`, `-r`, and `-R` options. You will receive an error message if you use `-e` in conjunction with `-s`, `-r`, or `-R`, or with an existing target directory.

**Migration action:**

- Modify any `In` invocations (for example, in shell scripts) so that `-e` is not used in conjunction with the `-s`, `-r`, and `-R` options.

If you want to create . . .	Use . . .
A symbolic link	<code>-s</code>
An external link	<code>-e</code>
Symbolic links or external links recursively	<code>-s</code> with <code>-r</code> or <code>-R</code>

- Modify `In` invocations so that `-e` is not used if the target specified on the command line is an existing directory.

**Update scripts to accommodate changes made to the locale shell command (Req-IF, as of R10)**

Required if automated scripts evaluate strings based on error messages FSUM6701 or the default locale output.

Before z/OS V1R10, the locale shell command did not support the `LC_TOD` category and associated keywords. The default locale command (specified without any options or operands) did not display the `LC_TOD` environment variable, and using `LC_TOD` (or associated keywords) with or without options generated error message FSUM6701. Beginning with z/OS V1R10, the default locale command displays `LC_TOD` information along with other `LC_*` environment variables. Also, using `LC_TOD` or associated `LC_TOD` keywords is honored.

**Migration action:** Modify automated scripts if they evaluate strings based on error message FSUM6701 or the default locale output.

**Update automation that handles message BPXF034I (Required-IF, as of R10)**

Required if any automated actions are issued for message BPXF034I.

Before z/OS V1R10, message BPXF034I was issued if a file system was quiesced for more than 10 minutes. As of z/OS V1R10, message BPXF083I is issued instead. The new message includes the file system name, quiescing system, job name, and process ID (PID).

**Migration action:** Modify automated actions for message BPXF034I so they now work with message BPXF083I.

**Update automation and operational procedures for file systems in a shared file system configuration (Required-IF, as of R9)**

Required if you have automation or operational procedures that use `MOUNT` statements that specify `NOAUTOMOVE`, `UNMOUNT`, or a system list.

Before z/OS V1R9, if you mounted a file system in a sysplex-aware mount mode with the `NOAUTOMOVE` setting, the `NOAUTOMOVE` setting was changed to `AUTOMOVE(YES)` and message BPXF234I was issued. Starting in z/OS V1R9, the conversion to `AUTOMOVE(YES)` will not occur for sysplex-aware file systems. Instead, the `AUTOMOVE` setting (`UNMOUNT`, `NOAUTOMOVE`, `AUTOMOVE`, or a system list) is now honored by z/OS V1R9 systems, independent of the sysplex-aware or sysplex-unaware capabilities for the mount mode. (The mount mode can be either read-only or read/write.)

The changes to the `AUTOMOVE` processing affect z/OS V1R9 systems only. If you have a mixed-release configuration, you may see the original behavior on systems prior to z/OS V1R9. For example:

- Before z/OS V1R9, if you mounted a file system in a sysplex-aware mount mode with the `NOAUTOMOVE` setting, the `NOAUTOMOVE` setting was changed to `AUTOMOVE(YES)` and message BPXF234I was issued. On a z/OS V1R9 system, if you mount a file system in a sysplex-aware mount mode with `NOAUTOMOVE`, that setting is honored. However, if the file system ownership is moved to a system earlier

than z/OS V1R9, the NOAUTOMOVE setting is changed to AUTOMOVE(YES). File systems that are mounted with a system list are also changed to AUTOMOVE(YES).

- If you mount a file system in a sysplex-aware mount mode with the NOAUTOMOVE setting, all z/OS V1R9 systems will honor the setting but systems earlier than z/OS V1R9 will not. For example, if a z/OS V1R9 owner system fails and there is a z/OS V1R8 system in the shared file system configuration, the z/OS V1R8 system will attempt to take over the file system and become the new owner.

**Migration action:** Review MOUNT statements that specify NOAUTOMOVE, UNMOUNT, or a system list and determine whether they should be updated. Notify operators about the changed mount behavior, if needed.

**Tip:** To find out if the file system is sysplex-aware, use the D OMVS,F command to display each mounted file system.

- Those file systems for which this system is not the owner, and for which it says Client=N, are *sysplex-aware*. They are not clients and are locally mounted to the PFS.
- Those file systems for which this system is not the owner, and for which it says Client=Y, are *sysplex-unaware*.

### **Increase dump space to accommodate dumps of the zFS address space (Required-IF, as of R9)**

*Required if the existing dump data space is not large enough to accommodate additional dumps.*

Before z/OS V1R9, system dumps did not include dumps from the zFS address space. In z/OS V1R9 and later, system dumps capture data from the zFS address space, as long as the address space is active and the cause of the dump is related to a file system problem. Consequently, a system dump can now be larger than before z/OS V1R9.

**Migration action:** Determine if you need to increase your dump data sets to include the additional space needed for data dumped from the zFS address space. The amount of increase depends on the amount of zFS activity and the configuration parameters. For example, the `trace_table_size` parameter specification may greatly affect the amount of storage that needs to be captured in the dump data set.

### **Update awk invocations to accommodate changes made to the awk utility (Required-IF, as of R9)**

*Required if either (or both) of the following situations is present:*

- If **awk** invocations use command-line arguments that have an “=” but do not follow the rules for a variable name. For example, `./x=y`.
- An **awk** program contains a `break` or `continue` statement outside a loop.

Before z/OS V1R9, the **awk** utility did not comply with Single UNIX Specification, Version 3 (SUSv3). In z/OS V1R9, the **awk** utility has been enhanced to conform with SUSv3.

- Certain arguments that were treated as variable assignments are now interpreted as path names. Prior to z/OS V1R9, **awk** recognized command-line arguments with an “=” character as a variable assignment. Starting in z/OS V1R9, **awk** will only recognize an argument as a variable assignment if it consists of the following characters (in the order listed):
  1. An underscore or alphabetic character
  2. A sequence of underscores, digits, and alphabetic characters
  3. The “=” character

If the argument does not follow that format, **awk** interprets it as a path name. Most **awk** commands will not have arguments that break the variable name rules.

- If a `break` or `continue` statement occurs outside a loop, **awk** either exits with an error message or, with the PTF for APAR OA22253 installed, displays a warning message, exits the current pattern or function, and continues processing. Previously, **awk** exited the current pattern or function without an error message and continued processing. Most **awk** programs will not have a `break` or `continue` statement that occurs outside a loop.

**Migration action:** Modify any **awk** invocations (for example, in shell scripts) that have command-line arguments with an “=” but break the rules for a variable name.

If you have an **awk** program that contains a `break` or `continue` statement outside a loop (which is reported by an error message or, with the PTF for APAR OA22253 installed, a warning message from **awk**), correct the program.

### **Use the new default of 64000 for the MAXFILEPROC statement (Recommended, as of R9)**


*Not required, but recommended if you use IBM Health Checker for z/OS to check whether the value of MAXFILEPROC is too low. (The check is USS\_MAXSOCKETS\_MAXFILEPROC.) If the value is 256, an exception message will be issued.*

Before z/OS V1R9, the default for the MAXFILEPROC statement in the BPXPRMxx parmlib member was 256. In z/OS V1R9, the default is 64000.




**Migration action:** The recommended value is 64000 or higher for the MAXFILEPROC statement in BPXPRMxx. Either use the default of 64000 or specify a higher number.

z/OS 1.10: Migration Part 3

## Important Actions for z/OS R10 from z/OS R8



**Migration Actions You Should NOT Overlook:**


-  •Redefine existing VSAM data sets that contain the IMBED, REPLICATE, and KEYRANGE attributes (Recommended)
-  •Migration from zFS multi-file system aggregates to compatibility mode aggregates (Recommended)
- Accommodate JES2 dynamic exits function (Required-IF, as of OA23561/OA21346)
- Use CEEPRMxx's CEEROPT instead of making copies of Language Environment initialization routines (Required-IF, as of R10)
-  •Migrate from HFS to zFS file systems (Recommended)

*...(and others from Part 2!!)*




20 © 2008 IBM Corporation

z/OS 1.10: Migration Part 3


## Recap of Important Actions from Part 2



**Migration Actions You Should NOT Overlook:**

- Add or change volumes to keep file systems in a single data set (Recommended)
- Update automation to accommodate new responses to the DISPLAY CONSOLES command (Required-IF, as of R10)
- Provide an appropriate HZSPDATA data set on fallback (Req-IF, as of R10)
- Reallocate the WLM couple data set (Required-IF, as of R10)
-  •Update callers of CNZMXURF that do not hold appropriate serialization (Required-IF, as of R10)
- Remove your *run-time* dependency on the C/C++ IBM Open Class Library (Required-IF, as of R9)
-  •IP Services: Migrate to the TN3270E Telnet server that runs in its own address space (Required-IF, as of R9)
- SNA Services: Ensure compatible levels of VTAM for HPR sessions (Required-IF, as of R10)
-  •ICSF HCR7750: Increase the size of your PKDS (Required, as of HCR7750)
- Integrated Security Server LDAP: Migration from LDAP Server to IBM TDS (Recommended)

21 © 2008 IBM Corporation

z/OS 1.10: Migration Part 3


## z/OS V1 R10 Install-Related Enhancements

**f Documentation Enhancements for z/OS Migration:**

- Customized books contain all supported migration paths to z/OS R10!  
*z/OS R8 to z/OS R10, and z/OS R9 to z/OS R10,*
- “Tips” have been added to appropriate migration actions for related IBM Health Checker checks (and Health Checks that are for migration).
- How to use the IBM Health Checker for migration checking is explained.

**f ServerPac Deliverable:**

- **ServerPac in z/OS R10:**
  - ★ **ACCEPT RSU\*:** now, only RSU\* and Integration Tested Service will be ACCEPTed, to make it easier to RESTORE some PTFs.
  - ★ **SYSUT\* and SMPWRK\* values in SMP/E CSI and jobs:** updated to be more “appropriate” for large (and electronic) installs.
  - ★ **SDSF for JES3 support:** SDSF will be in a separate zone, and merged when JES merge is selected. (You may merge both or neither JESes into the z/OS zone.) Removal of unused JES is unaffected.
  - ★ **CEA and SYSREXX Setup:** Security customization is performed.
- (As of R9: **ServerPac: Installing Your Order** document is available in Adobe PDF format, in addition to the existing BOOK format. The obsolete LIST1403 and LIST3820 formats are removed.)...**and lots of other goodies from prior releases!**

22
© 2008 IBM Corporation

## z/OS R10 Installation Enhancements in ServerPac

### ACCEPT RSU\*

For ServerPac orders, service is integrated with product code according to the following timeline:

- ServerPac is refreshed every month to pick up the most current RSU.
- All products incorporate high impact or pervasive (HIPER) and PTF-in-error (PE) fixes that are available up to the day your order is built. Because these PTFs are still fairly new at the time your order is built, they are not SMP/E ACCEPTed. They have been APPLYed only, therefore allowing you to back them off of your system using the SMP/E RESTORE command, if necessary. Your order also includes (in its SMP/E global zone) the latest unintegrated service that was available at the time the order was created. In addition, the PTFs that were SMP/E APPLYed but not ACCEPTed are also in the SMP/E global zone.
- All z/OS elements and features will incorporate additional service that has been through platform (integration) testing (SOURCEID ZOSV1Rn, where n is the release number).

Prior to z/OS R10, all PTFs that were APPLYed in ServerPac, were also ACCEPTed. As of z/OS R10, only the latest RSU (SOURCEID RSU\*) and the PTFs with SOURCEID ZOSV1Rn are ACCEPTed.

The publication *ServerPac: Installing Your Order*, which comes with your order, specifies the integrated service level applicable to your order. Your ServerPac order will also include unintegrated service that was available when your order was built.

### SYSUT\* and SMPWRK\* values in SMP/E CSI and jobs

The values for SMP/E’s SYSUT and SMPWRK data sets have (in some cases) been too small for installing large PTFs or large electronic orders. To accommodate this, the SYSUT and SMPWRK values have been increased in the ServerPac deliverable. These are the values that will be provided in ServerPac (in 3390 tracks):

SMPWRK1	PDS FB 80	150
SMPWRK2	PDS FB 80	150
SMPWRK3	PDS FB 80	9500
SMPWRK4	PDS FB 80	132
SMPWRK6	PDS FB 80	20000
SYSUT1	SEQ -- --	7500
SYSUT2	SEQ -- --	1500
SYSUT3	SEQ -- --	960
SYSUT4	SEQ -- --	3000

### **SDSF for JES3 Support**

With z/OS R10 SDSF, support for JES3 has been added. To accommodate this in ServerPac, the “JES Element Selection” panel will do some enhanced processing.

On this panel, select at least one JES element for installation: JES2 (and SDSF) or JES3 (and SDSF). If your installation requires both JES elements to be installed, you can select BOTH.

For each JES element you select, specify whether the dialog is to merge the SMP/E zones of the JES element with the Base Control Program (BCP) zones. When you merge a JES element with the BCP zones, the SDSF zone is automatically merged with the BCP zones:

- If you specify the JES2 zone to be merged with the BCP zones, both JES2 and SDSF zones are merged with the BCP zones. Similarly, if you specify JES3 to be merged with the BCP zones, both JES3 and SDSF zones are merged with the BCP zones.
- If you specify both JES elements to be merged with the BCP zones, the JES2, JES3, and SDSF zones are merged with the BCP zones.


**Note:** If you install both JES elements, you cannot merge only one of the JES elements with the BCP zones. The error message CPPP0601011E will appear to indicate that the action is not allowed.

If you migrate JES2 or JES3 together with the rest of z/OS, IBM recommends that you specify **Y** on this panel to merge the selected JES elements into the BCP zone. However, if you plan to stage your z/OS and JES migrations separately, do not merge zones. The dialog loads the JES element’s zones, but does not merge them.

### **CEA and SYSREXX Setup**

Security customization (provided in ServerPac job RACFTGT) now includes the necessary security customization for CEA and System REXX.



z/OS 1.10: Migration Part 3 


z/OSR10 Install-Related Enhancements

### Summary of Enhancements of System Programmer Interest


- ☺ **ISPF: Multiple targets for MOVE/COPY, Enhanced screen swapping, and Block commands in 3.4**
- ☺ **Language Environment: Syntax checking tool for CEEPRMxx parmlib members**
- ☺ **DFSORT: Using PARMLIB members to set installation options**

R9 goodies you also get in R10:

- ❖ **zFS: IOEAGFMT removes SUPERUSER.FILESYS.PFSCNTL requirement to format a zFS in batch.**
- ❖ **SDSF: REXX interface to SDSF**
- ❖ **ISPF: Support for editing and browsing z/OS UNIX and ASCII files**



23 © 2008 IBM Corporation

z/OS 1.10: Migration Part 3


z/OSR10 Instal-Related Enhancements

## ISPF: Various Enhancements

- **Multiple targets for ISPF Editor MOVE and COPY**
  - Nice when you want to put the same line into multiple places in one fell swoop!
  - ★ **Exploitation:** Use AK (or BK) on the M or C line command, then A (or B) when done.
  
- **ISPF enhanced screen swapping**
  - View all the logical screens in your ISPF session and navigate more easily between them. Just point-and-shoot to switch!
  - ★ **Exploitation:** Use command SWAPBAR ON.
  
- **Block commands allowed in ISPF Data Set List Utility**
  - You can now enter line commands in blocks using the ISPF Data Set List Utility (option 3.4)
  - ★ **Exploitation:** Use //x and // to mark blocks and do the command (where 'x' is your line command).

24
© 2008 IBM Corporation

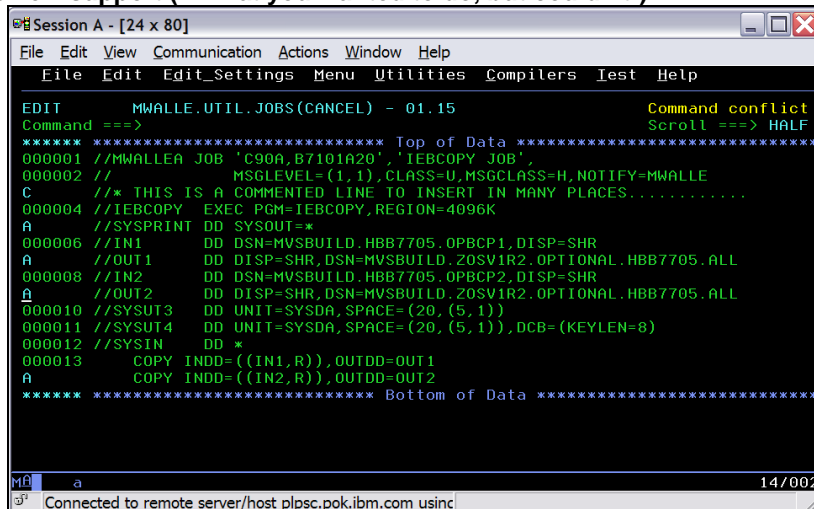
## ISPF Various Enhancements

### Multiple targets for ISPF Editor MOVE and COPY

The enhanced ISPF Editor MOVE and COPY primary commands lets you to move or copy data to many locations within the edit session in one operation, reducing the time you spend editing files. Previously, you would have had to move or copy data to one location at a time.

When data is to be moved or copied, the A (after) line command specifies the line after which the data is to be placed. When data is to be moved or copied to multiple destinations, the A (after) line command specifies the final destination line after which the data is to be placed. When data is to be moved or copied to multiple destinations, the AK (after, multiple targets) line command specifies each multiple destination line (apart from the final destination line) after which the data is to be placed

**Example, without this new support (...what you wanted to do, but couldn't!):**



```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT          MWALLE_UTIL_JOBS(CANCEL) - 01.15          Command conflict
Command ==>                                         Scroll ==> HALF
***** ***** Top of Data *****
000001 //MWALLE JOB 'C90A,B7101A20','IEBCOPY JOB',
000002 //          MSGLEVEL=(1,1),CLASS=U,MSGCLASS=H,NOTIFY=MWALLE
C      /** THIS IS A COMMENTED LINE TO INSERT IN MANY PLACES.....
000004 //IEBCOPY EXEC PGM=IEBCOPY,REGION=4096K
A      //SYSPRINT DD SYSOUT=*
000006 //IN1 DD DSN=MVSBUILD.HBB7705.OPBCP1,DISP=SHR
A      //OUT1 DD DISP=SHR,DSN=MVSBUILD.ZOSV1R2.OPTIONAL.HBB7705.ALL
000008 //IN2 DD DSN=MVSBUILD.HBB7705.OPBCP2,DISP=SHR
A      //OUT2 DD DISP=SHR,DSN=MVSBUILD.ZOSV1R2.OPTIONAL.HBB7705.ALL
000010 //SYSUT3 DD UNIT=SYSDA,SPACE=(20,(5,1))
000011 //SYSUT4 DD UNIT=SYSDA,SPACE=(20,(5,1)),DCB=(KEYLEN=8)
000012 //SYSIN DD *
000013 COPY INDD=((IN1,R)),OUTDD=OUT1
A      COPY INDD=((IN2,R)),OUTDD=OUT2
***** ***** Bottom of Data *****

MWALLE a 14/002
Connected to remote server/host plpsc.pok.ibm.com using

```

## Example with this new support, as of R10:

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT MMALLE.UTIL.JOBS(CANCEL) - 01.15 Columns 00001 00072
Command ==> Scroll ==> HALF
***** Top of Data *****
000001 //MMALLEA JOB 'C90A,B7101A20', 'IEBCOPY JOB',
000002 // MSGLEVEL=(1,1),CLASS=U,MSGCLASS=H,NOTIFY=MMALLE
000003 /* THIS IS A COMMENTED LINE TO INSERT IN MANY PLACES.....
000004 //IEBCOPY EXEC PGM=IEBCOPY,REGION=4096K
000005 //SYSPRINT DD SYSOUT=*
000006 /* THIS IS A COMMENTED LINE TO INSERT IN MANY PLACES.....
000007 //IN1 DD DSN=MVSBU1LD.HBB7705.OPBCP1,DISP=SHR
000008 //OUT1 DD DISP=SHR,DSN=MVSBU1LD.ZOSVIR2.OPTIONAL.HBB7705.ALL
000009 //IN2 DD DSN=MVSBU1LD.HBB7705.OPBCP2,DISP=SHR
000010 //OUT2 DD DISP=SHR,DSN=MVSBU1LD.ZOSVIR2.OPTIONAL.HBB7705.ALL
000011 //SYSUT3 DD UNIT=SYSDA,SPACE=(20,(5,1))
000012 //SYSUT4 DD UNIT=SYSDA,SPACE=(20,(5,1)),DCB=(KEYLEN=8)
000013 //SYSIN DD *
000014 COPY INDD=((IN1,R)),OUTDD=OUT1
000015 COPY INDD=((IN2,R)),OUTDD=OUT2
***** Bottom of Data *****
19/003
Connected to remote server/host.plpsc.pok.ibm.com using

Session A - [24 x 80]
File Edit View Communication Actions Window Help
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT MMALLE.UTIL.JOBS(CANCEL) - 01.15 Columns 00001 00072
Command ==> Scroll ==> HALF
***** Top of Data *****
000001 //MMALLEA JOB 'C90A,B7101A20', 'IEBCOPY JOB',
000002 // MSGLEVEL=(1,1),CLASS=U,MSGCLASS=H,NOTIFY=MMALLE
000003 /* THIS IS A COMMENTED LINE TO INSERT IN MANY PLACES.....
000004 //IEBCOPY EXEC PGM=IEBCOPY,REGION=4096K
000005 //SYSPRINT DD SYSOUT=*
000006 /* THIS IS A COMMENTED LINE TO INSERT IN MANY PLACES.....
000007 //IN1 DD DSN=MVSBU1LD.HBB7705.OPBCP1,DISP=SHR
000008 //OUT1 DD DISP=SHR,DSN=MVSBU1LD.ZOSVIR2.OPTIONAL.HBB7705.ALL
000009 //IN2 DD DSN=MVSBU1LD.HBB7705.OPBCP2,DISP=SHR
000010 //OUT2 DD DISP=SHR,DSN=MVSBU1LD.ZOSVIR2.OPTIONAL.HBB7705.ALL
000011 //SYSUT3 DD UNIT=SYSDA,SPACE=(20,(5,1))
000012 //SYSUT4 DD UNIT=SYSDA,SPACE=(20,(5,1)),DCB=(KEYLEN=8)
000013 //SYSIN DD *
000014 COPY INDD=((IN1,R)),OUTDD=OUT1
000015 COPY INDD=((IN2,R)),OUTDD=OUT2
000016 /* THIS IS A COMMENTED LINE TO INSERT IN MANY PLACES.....
***** Bottom of Data *****
23/002
Connected to remote server/host.plpsc.pok.ibm.com using

```

## Reference information: z/OS ISPF Edit and Edit Macros

## ISPF enhanced screen swapping

This enhancement allows you to view all the logical screens in your ISPF session and navigate more easily between them. ISPF now provides a new system command, SWAPBAR, which displays a line at the bottom of the physical screen that contains point-and-shoot fields showing the logical screens for your ISPF session. You can put the cursor on the point-and-shoot field for a logical screen, press Enter, and the logical screen becomes the active screen.

The list of logical sessions can be activated by entering the SWAPBAR or SWAPBAR ON command on the command line. The list is displayed on the last line of the physical screen.

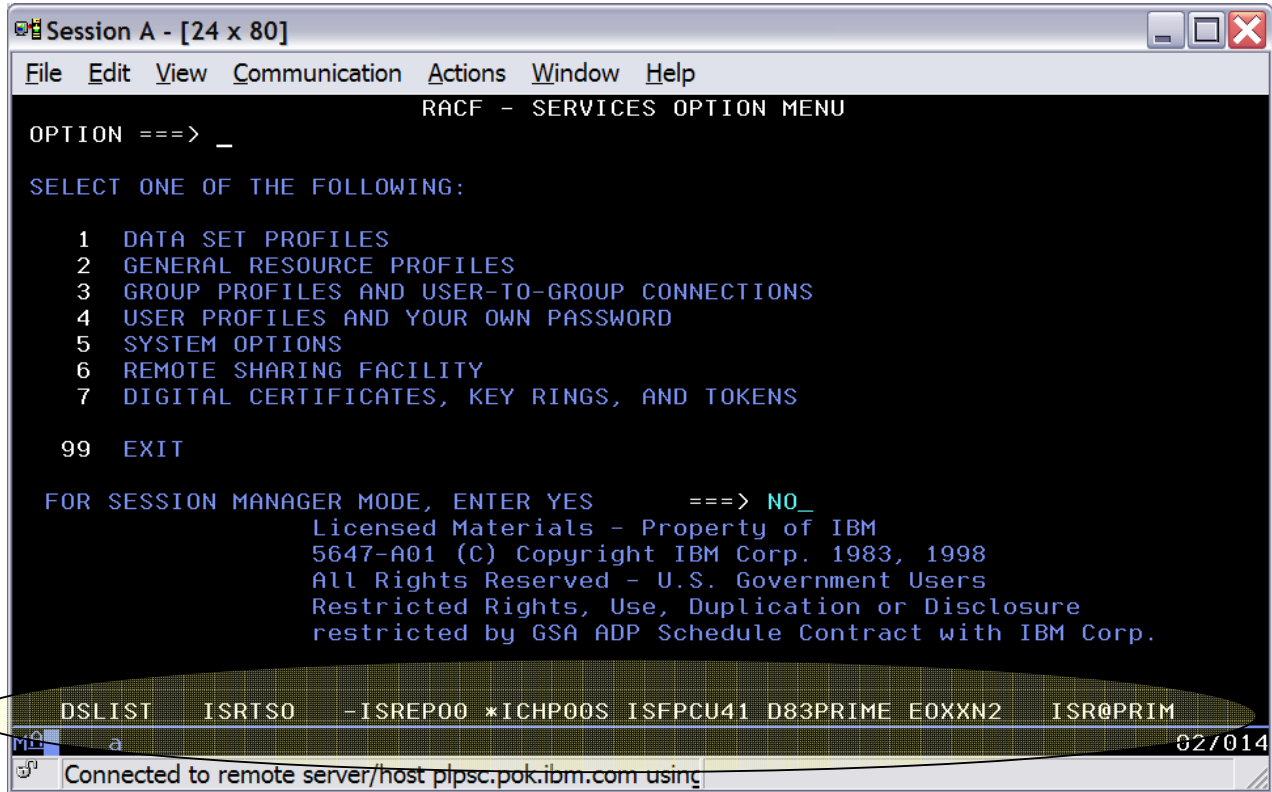
The entry for each logical session is the screen name if assigned or, if a screen name is not available, it is the panel name of the current panel displayed for the logical session. The entry for the active logical session has an asterisk (\*) in the first character position and, if the name is 8 bytes long, the last character is not displayed. Also, the alternate logical session has a "-" in the first position and the 8th character is not displayed.

The list remains active until you enter the SWAPBAR or SWAPBAR OFF command. The setting for the SWAPBAR is maintained in the system profile member and applies across logons.

If the list is longer than the width of the screen, a ">" appears at the right of the list to indicate there are more entries and you can scroll right by either positioning the cursor on the ">" and pressing Enter, or by positioning the cursor on an entry and pressing PF11 which scrolls to the entry indicated. When the start of the list is not displayed, a "<" is displayed at the left of the visible part of the list to indicate that you can scroll left on the list by positioning the cursor on the "<" and Pressing Enter or by positioning the cursor on an entry and pressing PF10.

The active logical session can be changed by positioning the cursor on an entry and pressing Enter. This logical session then becomes the active logical session. The SWAPBAR entries use the same physical attribute as the action bar choices and, if in the options settings **Tab to action bar choices** is selected, then tabbing to swapbar entries also occurs.

If the cursor is positioned on the swapbar entry for the currently active session and Enter is pressed, this is treated the same as Enter being pressed within the active logical session panel. When the SWAPBAR is activated, the **Always show split line** option is deactivated and you are not able to reactivate it until the SWAPBAR is deactivated.



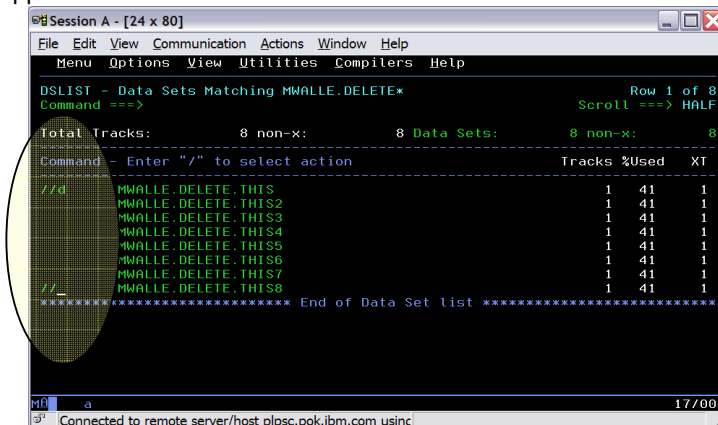
Reference information: z/OS ISPF User's Guide Vol I, and z/OS ISPF Dialog Developer's Guide and Reference

**Block commands allowed in ISPF Data Set List Utility**

You can now enter line commands in blocks using the ISPF Data Set List Utility (option 3.4). This greatly reduces the number of line commands you are required to enter on the data set list utility when you need to process a number of ISPF data sets grouped together in the list. For example, if you want to delete all the data sets displayed in the data set list, you are only required to enter the delete block line command against the first and last data sets in the list.

You mark the block by typing a "/" at the beginning of a block of rows and another "/" at the end of the block of rows. You must type the line command either immediately after the // on the first row of the block, or immediately after the // on the last row of the block.

You can enter several blocks of commands at the same time, but you cannot nest them. Single line commands are not allowed within a block command. You can execute all line commands, including TSO commands, Clists and REXX execs as block commands. If you have selected the DSLISL settings option **Execute Block Commands for excluded Data Sets**, all applicable excluded rows are unexcluded before the block commands are executed.



z/OS 1.10: Migration Part 3

z/OSR10 Install-Related Enhancements

## LE: Syntax checking tool for CEEPRMxx parmlib members

- **In R10, you can check the syntax of new/changed CEEPRMxx parmlib members from batch or from TSO.**
  - Parmlib member must be in a FB80 data set. Can specify concatenated members, which is nice for verification of what will be used.
  - For TSO, SCEECLST must be in SYSPROC or SYSEXEC concatenation.
  - When no errors are found, you get a nice report of the run-time options for each group and “Last Where Set”!
  - When errors are found, you get messages for what is wrong and where.

★ **Exploitation:**

★ **For batch:**

```
//CEEPRMCK EXEC PGM=CEEPRMCC ,
// PARM='CEE=(MW) '
//CEEPRMCK DD DSN=MWALLE.CEE.PARMLIB,DISP=SHR
```

★ **For TSO:**

```
====> CEEPRMCK MEMBERS(MW) DSN('MWALLE.CEE.PARMLIB')
```

25
© 2008 IBM Corporation

### Syntax checking tool for CEEPRMxx parmlib members

Language Environment is enhanced to provide a tool for syntax checking of one or more CEEPRMxx members of PARMLIB before attempting to activate them. CEEPRMxx members can be syntactically checked for errors under z/OS batch or TSO/E. Before calling the syntax checker, a CEEPRMxx member must be created and placed in a PDS or PDSE.

#### Under z/OS batch

The CEEPRMCC program reads and parses the CEEPRMxx member for syntax errors. The CEEPRMCC program displays the run-time options report if no errors are found. The run-time options report only displays options that are specified inside the CEEPRMxx members. The CEEPRMCC expects the following inputs:

- The PARM parameter Use the PARM parameter of the EXEC job control statement to select one or more CEEPRMxx members.  

```
// PARM='CEE=(xx,yy,...,nn)'
```

The two alphanumeric characters, xx,yy,...,nn, are the suffix of the CEEPRMxx members to be checked. Embedded blanks are not allowed within the PARM.
- Optional CEEPRMCK DD Statement Use DD statement to specify the data set where CEEPRMxx members are located. //CEEPRMCK DD DSN=MEENAK.SYSTEM.PARMLIB,DISP=SHR If no DD is specified, the CEEPRMCC program uses the default data set SYS1.PARMLIB.

#### Example:

```
//CEEPRMCK EXEC PGM=CEEPRMCC ,
// PARM='CEE=(MW) '
//CEEPRMCK DD DSN=MWALLE.CEE.PARMLIB,DISP=SHR
```

**Usage notes** An input data set must be a fixed record format and a record length of 80.

#### Under TSO/E

Under TSO/E, CEEPRMCK reads and parses the CEEPRMxx member for syntax errors. CEEPRMCK displays the run-time options report if no errors are found. The run-time options report only displays options that are specified inside the CEEPRMxx members. The syntax of the CEEPRMCK invocation is as follows:

**CEEPRMCK MEMBERS( xx ) DSN ( *data-set-name* )**

- **xx** The two alphanumeric characters that are the suffix of the CEEPRMxx members to be checked. The **MEMBERS** keyword parameter must always be specified.
- **data-set-name** The data set name that contains the specified CEEPRMxx member. The fully qualified data set name must be enclosed in single quotes if a TSO/E prefix is not required. The **DSN/DSNAME** keyword parameter is optional. If both DD is allocated and DSN or DSNAME is specified, the CEEPRMCK program uses the DD and the DSN/DSNAME is ignored. DD allocation overrides DSN/DSNAME specification. If no DD is allocated and no DSN or DSNAME is specified, the CEEPRMCK program uses the default data set SYS1.PARMLIB.

**Example:** CEEPRMCK MEMBERS(MW) DSN('MWALLE.CEE.PARMLIB')

#### Usage notes:

- An input data set must be a fixed record format and a record length of 80.
- To invoke CEEPRMCK by using the documented syntax, SCEELST must be allocated to a system file (SYSPROC or SYSEXEC). See *z/OS TSO/E REXX User's Guide* for more information about setting up and using REXX execs.

#### Sample Output

If there are syntax errors, no run-time options report will be displayed. Error messages will be written to the Language Environment message file. CEE3761I will be followed by other existing error messages related to syntax errors in CEEPRMxx members and end with CEE3762I.

Here is the partial output, when no syntax errors are found:

```
CEE3762I The Language Environment Parmlib checker has completed.
```

```
CEE3745I 16.38.52 Display CEEDOPT
```

```
CEE=(MW)
```

LAST WHERE SET	OPTION
PARMLIB(CEEPRMMW)	ABPERC(NONE)
PARMLIB(CEEPRMMW)	ABTERMENC(ABEND)
PARMLIB(CEEPRMMW)	NOAIXBLD
PARMLIB(CEEPRMMW)	ALL31(ON)
PARMLIB(CEEPRMMW)	ANYHEAP(16384,8192,ANYWHERE,FREE)
PARMLIB(CEEPRMMW)	NOAUTOTASK
PARMLIB(CEEPRMMW)	BELOWHEAP(8192,4096,FREE)
PARMLIB(CEEPRMMW)	CBLOPTS(ON)
PARMLIB(CEEPRMMW)	CBLPSHPOP(ON)
PARMLIB(CEEPRMMW)	CBLQDA(OFF)
PARMLIB(CEEPRMMW)	CEEDUMP(60,SYSOUT=*,FREE=END,
...	

Here is the complete output, when syntax errors are found. (Where “WEIRD” were the violations that were being detected):

```
CEE3761I The following messages pertain to the call to the Language Environment Parmlib checker.
```

```
CEE3731I The following messages pertain to the system default run-time options in the CEEDOPT in CEEPRMMX
```

```
CEE3616I The string 'WEIRD' was not a valid or supported suboption of the run-time option RPTOPTS in this release.
```

```
CEE3731I The following messages pertain to the system default run-time options in the CEEDOPT in CEEPRMMX.
```

```
CEE3614I An invalid character occurred in the numeric string 'WEIRD' of the run-time option DEPTHCONDLMT.
```


```
CEE3731I The following messages pertain to the system default run-time options in the CELQDOPT in CEEPRMMX.
```

```
CEE3616I The string 'WEIRD' was not a valid or supported suboption of the run-time option DEPTHCONDLMT.
```

```
CEE3749I The value specified for keyword CEEROPT found in CEEPRMMX is not valid.
```

```
CEE3762I The Language Environment Parmlib checker has completed.
```

**Reference information:** For more information on the tool for checking the syntax of CEEPRMxx members, see: *z/OS Language Environment Customization*.

z/OS 1.10: Migration Part 3



z/OSR10 Install-Related Enhancements

## DFSORT: Using ICEPRMxx to set installation options

- Each ICEPRMxx member can contain options to be changed for any or all of DFSORT's eight installation environments.
- Up to ten ICEPRMxx members can be activated by a START ICEOPT started task command. The options in the activated members will be merged with the ICEMAC defaults at run-time.
- Old way, ICEMAC macro and usermods, are still supported, but this new way is much better!

★ **Exploitation:**

- ★ **Read *DFSORT Installation and Customization...basically you:***
  - ★ **code up your desired ICEPRMxx values**
  - ★ **activate them with command `start iceopt,iceprm=xx`**
  - ★ **run an ICETOOL DEFAULTS report to verify that your installation defaults are as expected**



26
© 2008 IBM Corporation

### Using ICEPRMxx parmlib members to set installation options

ICEPRMxx members in concatenated PARMLIB can now be used to specify changes to DFSORT's installation options. Each ICEPRMxx member can contain options to be changed for any or all of DFSORT's eight installation environments (JCL, INV, TSO, TSOINV and TD1-TD4). Up to ten ICEPRMxx members can be activated by a START ICEOPT started task command. The options in the activated members will be merged with the ICEMAC defaults at run-time. A different ICEPRMxx member, or combination of ICEPRMxx members, for different LPARs can be activated at IPL time by including a START ICEOPT command in an appropriate COMMNDxx member in PARMLIB, or at any time by issuing a START ICEOPT command from the console.

**Note:** You can use MVS console security to control who can issue the START ICEOPT command.

ICEPRMxx members are now the recommended way to change DFSORT installation defaults since they are easier to use and more flexible than the old method using the ICEMAC macro and usermods. However, the old method is still supported. The ICETOOL DEFAULTS operator can be used at any time to produce a report showing the merged PARMLIB/ICEMAC installation default values for each environment that will be used at run-time, as well as the active ICEPRMxx and ICEMAC values.

### Coding Options in ICEPRMxx members

You can specify installation options for any or all of the eight environments in an ICEPRMxx member of concatenated PARMLIB. You must precede each group of options with one of the following keywords to indicate the installation **Installation Defaults**. See the *DFSORT Installation and Customization* book for syntax requirements.

### Activating ICEPRMxx Members

To change the installation defaults with one ICEPRMxx member in PARMLIB, issue a START ICEOPT command to activate that member as follows: `start iceopt,iceprm=xx` where xx is the suffix of the ICEPRMxx member you want to use. For example, to activate ICEPRM01, use:

```
start iceopt,iceprm=01
```

To change the installation defaults with more than one ICEPRMxx member in PARMLIB, issue a START ICEOPT command to activate those members as follows:

```
start iceopt,iceprm=(x1,...,xn)
```

where x1,...,xn are the suffixes of the ICEPRMxx members you want to use.

For example, to activate ICEPRMMN and ICEPRMS3, type the following on the console:

```
start iceopt,iceprm=(mn,s3)
```

You can use a START ICEOPT command at any time to change your installation defaults by activating up to 10 ICEPRMxx members. You can issue a START ICEOPT command from the console, or include it in an appropriate COMMNDxx member in PARMLIB so it will be executed automatically at IPL time. However, we recommend that you test your ICEPRMxx members by activating them from the console before activating them automatically with a COMMNDxx member. ICEOPT issues messages to the console to indicate the status of your installation default changes as well as any errors it detects. Be sure to review these console messages and the return code from ICEOPT.

After you execute a START ICEOPT command, you can run an ICETOOL DEFAULTS report to verify that your installation defaults are as expected.

**Reference information:** *z/OS DFSORT Installation and Customization*



**z/OS 1.10: Migration – Part 3 Summary**

- **DFSMS Migration Actions:**
  - Control data set ID for RMM, changes for EAV support, change in CA sizes assigned
  - DEFRAG command in DSS, new facility class names for (H)BACKDSNEWNAME command.
- **DFS Migration Actions:** Fix syntax errors for SMB startup.
- **JES2 Migration Actions:** Automation for some \$D commands.
- **JES3 Migration Actions:** IATUX63 changes.
- **LE Migration Actions:** New runtime options (no change in behavior, though).
- **SDSF Migration Action:** Use dynamic statements for ISFPARMS sample locations have moved.
- **z/OS UNIX Migration Action:**
  - /etc files updated, increase dump space for zFS, submit command, ln changes.
- **z/OS R10 Installation Enhancements:** for z/OS Migration and for Server Pac
- **z/OS R10 SysProg Enhancements:** various ISPF, CEEPRMxx checker, ICEPRMxx support

**Also, don't forget to review the Important Migration Actions!**