



Professor Kimura explains z/OS migration to you

BY SHIGEKI KIMURA

Professor "z/OS migration" has expertise in and deep knowledge of z/OS release-to-release migration. Customers who need help in z/OS migration ask him for advice, and he writes a prescription to resolve the pain. This article contains a series of his practical advice in the following areas.

- Is there any migration assistance to help identify the impact of the change?
- Is there any enhancement or solution in a new release to help with the customer's request?
- Is there any compatibility option in a new release to handle the old behavior?

Migration assistance for the change of DISP=UNCATLG processing

From the customer:

Dear Mr. Migration:

After the transition to z/OS V1R10, we experienced an error situation in DISP=UNCATLG processing as follows. How can we easily identify the affected JCL statements to be updated?

```
//UNCAT      DD  DSN=BEANS.ZOS.HOT.TOPICS.SAMPLE1,
//           UNIT=3390,DISP=(OLD,UNCATLG),VOL=SER=WRKI02
//DEL       DD  DSN=BEANS.ZOS.HOT.TOPICS.SAMPLE1,
//           UNIT=3390,DISP=(OLD,DELETE),VOL=SER=WRKI02
```

```
IEF142I BEANSZZ STEP1 - STEP WAS EXECUTED - COND CODE 0000
IEF287I  BEANS.ZOS.HOT.TOPICS.SAMPLE1                NOT
UNCTLGD 13
IEF287I  VOL SER NOS= WRKI02.
IEF285I  BEANS.ZOS.HOT.TOPICS.SAMPLE1
DELETED
IEF285I  VOL SER NOS= WRKI02.
```

Professor's advice:

Thank you very much for your inquiry. Your observation is correct because the behavior in z/OS V1R10 now only allows a data set to be uncataloged when the data set information is retrieved from the catalog at allocation time to prevent accidental data loss.

I suggest that you make use of the new function shipped by APAR OA27917. This APAR allows you to avoid a migration action if the changes required for it are too disruptive (for example, if you need to change a lot of production jobs). It introduced a new option for the ALLOCxx parmlib member to provide migration assistance for existing JCL that might be incorrectly coded and it takes advantage of the pre-V1R10 behavior. You can request the pre-V1R10 behavior by tracking the job and program information with the Tracking Facility (using the SYSTEM VERIFY_UNCAT(TRACK) option) and optionally with message IEF384I in the job log to indicate that the incorrect data set might have been uncataloged (using the SYSTEM VERIFY_UNCAT(MSGTRACK) option).

This is something that is valuable even after you have migrated to z/OS V1R10, because you can still use the tracker to indicate what is out there that you might still want to identify or update. This item should also be useful for other customers who migrate to z/OS V1R11 from z/OS V1R9.

While APAR OA27917 provided new function to assist in migrating existing JCL that might be incorrectly coded and provided the SYSTEM VERIFY_UNCAT settings of FAIL, TRACK, and MSGTRACK, a second APAR OA32843 extends this migration function with a new SYSTEM VERIFY_UNCAT setting of LOGTRACK. When LOGTRACK is in effect, the installation can request the pre-V1R10 behavior, with tracking of the job and program information in the Tracking Facility. In such cases, the system issues message MSGIEF384I as a hardcopy-only WTO message, so you can find it in either SYSLOG or OPERLOG and make the necessary automation updates for your installation. LOGTRACK is similar to MSGTRACK, with the only difference that MSGIEF384I is not issued as a WTO message when MSGTRACK is in effect.

You can find more information about the Tracking Facility at the following Web site:

http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/IEA2G3A0/A.0?SHELF=EZ2ZBK0H&DT=20090603221244

Here is the sample output from your job with the MSGTRACK option activated using the SETCON TRACKING=ON command.

Output from the Tracking Facility:

```
D O,TRACKING - INTERNALLY ISSUED BY CNZKSETC. FACILITY
TURNED OFF
CNZ1001I 14.27.08 TRACKING DISPLAY 984
STATUS=ON          NUM=1          MAX=1000 MEM=n/a EXCL=0
REJECT=0
----TRACKING INFORMATION----- -VALUE-- JOBNAME  PROGRAM+OFF-
- ASID NUM
IEFALC 01: STEP1      UNCAT              00 BEANSZZ  IEFIIC
00  2D  1
-----
```

JESYSMSG of JOBLOG:

```
IEF142I BEANSZZ STEP1 - STEP WAS EXECUTED - COND CODE 0000
IEF384I BEANSZZ STEP1 UNCAT - WARNING: VOLUME NOT RETRIEVED
FROM CATALOG
IEF285I BEANS.ZOS.HOT.TOPICS.SAMPLE1
UNCATALOGED
IEF285I VOL SER NOS= WORK01.
IEF285I BEANS.ZOS.HOT.TOPICS.SAMPLE1
DELETED
IEF285I VOL SER NOS= WORK01.
```

Solution for the duplicate temporary data set name

From the customer:

Dear Professor Migration:

When a temporary data set is specified as DSN=&&tempname, the system-generated qualified name for the temporary data set is not unique under two conditions. The first is if the system created data sets for multiple tasks or APPC transactions having the same job names within the same system clock second. The second is if the tasks or transactions contain DD statements that specify the same temporary data set names. To help avoid the problem and suppress the JCL error, we need to remove the DSN=&&tempname specification and update the rest of the JCL to use refer-back notation for the data set. Do you have any solution to avoid this time-consuming work?

Professor's advice:

Yes, a new function in z/OS V1R12 will help you. Beginning with z/OS V1R12, you can use a new ALLOCxx parmlib option, SYSTEM TEMPDSFORMAT(UNIQUE), to specify that the system uses the data set naming convention for unnamed temporary data set instead. This substantially reduces the probability of this JCL error without the need to change JCL. This solution involves generating a unique name for a temporary data set even when the DSN=&&tempname is specified; consequently, the resulting data set name looks like the one generated when no DSN=&&tempname is specified.

Solution for the overflow of SMF30SRV field

From the customer:

Dear Professor Migration:

There are 4-byte fields (SMF30xxx: xxx=SRV,CSU,SRB,IO,MSO,ESU) in the Performance Section of the SMF type 30 record. We always need to get the correct values because the service units (SMF30SRV) are used as a base for customer billings. However, this field grows to 'FFFFFFFF'x and then wraps back to zero and continues growing, so it results in erroneous resource consumption

of data, especially for the long running jobs. Our shop is now running z/OS V1R10; How can we resolve this issue?

Professor's advice:

You are so lucky because this usability issue can be resolved by new expanded sized service unit fields in the SMF type 30 records. When you have migrated to z/OS V1R11, or if you have applied PTF UA42968 for APAR OA25540 (SRM) and PTF UA47865 for APAR OA26832 (SMF) in z/OS V1R10, you can use the following new functions.

- New 8-byte fields (SMF30xxx_L: xxx=SRV,CSU,SRB,IO,MSO,ESU) are added to the Performance Section of the SMF Type 30 record. These fields are equivalent to the corresponding 4-byte fields and continue to be valid after the 4-byte fields wrap.
- A new flag byte SMF30INV is also added to the Performance Section. It provides individual bits that are set when each 4-byte field exceeds its maximum capacity. For example, once wrapping occurs for SMF30SRV, SMF30SRV_INV will be set to **on**.

Enhancement of writing an EOF mark for non-SMS managed sequential data set

From the customer:

Dear Professor Migration:

A guy who is new to z/OS in our shop asked me why we run the in-house procedure to open output and close the newly allocated non-SMS managed sequential data set. I know the simple answer for him, but it makes me ask you another question. How can we resolve this issue without our own effort?

Professor's advice:

Now, I have good news for you. Beginning with z/OS V1R11, when the system allocates a new data set with DSORG=PS or no DSORG, regardless of whether it is SMS-managed or not, it writes a file mark at the beginning any data set for which space is allocated. With this enhancement, a program can safely read the data set before data has been written in it.

Here is a simple scenario to demonstrate this new feature. Note that ISPF OPT3.2 Allocation writes an end-of-file mark for new non-SMS managed sequential data set even before z/OS V1R11.

```
//STEP1 EXEC PGM=IEFBR14
//DD1 DD
DSN=BEANS.ZOS.HOT.TOPICS.SAMPLE2,DISP=(NEW,CATLG),
// UNIT=3390,VOL=SER=WRKI02,SPACE=(TRK,(1,0)),
// RECFM=FB,LRECL=80,BLKSIZE=0,DSORG=PS
//STEP2 EXEC PGM=ADRDSSU,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
```

```
PAGE 0001      5695-DF175  DFSMSDSS V1R11.0 DATA SET SERVICES
2010.095 14:46
  PRINT DATASET(BEANS.ZOS.HOT.TOPICS.SAMPLE2)  INDYNAM(WRKI02)
00061002
ADR006I (001)-STEND(01), 2010.095 14:46:28 EXECUTION BEGINS
*** TRACK(CCHH) 00070008          R0 DATA 0000000000000000
      COUNT 0007000801000000
ADR006I (001)-STEND(02), 2010.095 14:46:28 EXECUTION ENDS
```

Consideration of new STORAGENSWDP option in IEAOPTxx parmlib member

From the customer:

Dear Professor Migration:

We recently experienced an auxiliary storage shortage (IRA200E) in our production LPAR. Then, the system issued the following messages:

```
*IRA206I ANTMAIN ASID 000B FRAMES 0000613386 SLOTS
0000281553 % OF AUX 93.2
*IRA210E NETVIEW ASID 0085 SET NON DISPATCHABLE
FRAMES+SLOTS 0000005539 RATE 000011
```

and the important address spaces with the non-swappable attribute, such as the message automation subsystem, were unexpectedly set to non-dispatchable. How can we avoid this problem situation?

Professor's advice:

Thanks a lot for your e-mail. Beginning with z/OS V1R10, when an auxiliary storage shortage occurred, the system identifies the address space with the largest increase in the amount of allocated central plus auxiliary storage and the address space is set to non-dispatchable regardless of whether it is swappable or non-swappable until the shortage is relieved.

You can control this default behavior with the new STORAGENSWDP parameter, introduced in z/OS V1R10, in the IEAOPTxx parmlib member. STORAGENSWDP=YES (default) specifies that the system should also select non-swappable address spaces to resolve the storage shortage except for the address spaces in service class SYSTEM. To avoid your problem situation and expect the same behavior as previous releases, setting the STORAGENSWDP=NO parameter can be an answer for you. Good luck!

Consideration of HIBFREXT and LOBFREXT options in TSOKEYxx parmlib member

From the customer:

Dear Professor Migration:

After migrating to z/OS V1R10, some TSO/E users reported that the Forward (PF8) operation in ISPF/PDF (OPT1 VIEW or OPT2 EDIT) hung. When they pressed the ESC key, the next screen came up. Is this a known problem?

Professor's advice:

I would suggest that you check the HIBFREXT and LOBFREXT values in your TSOKEYxx parmlib member. I do not recommend using any lower values than the defaults, HIBFREXT=48000 and LOBFREXT=24000, and this is very important once you migrate to z/OS V1R10. There have been quite a few instances of customers hanging at z/OS V1R10 because of low settings for these parameters. To avoid this situation, both HIBFREXT and LOBFREXT should be set, at least, to the default values.

While you might have used the TSOKEY00 parmlib member shipped by z/OS V1R10 ServerPac which contained HIBFREXT=6600 and LOBFREXT=3300, the good news is that z/OS V1R11 ServerPac has already been enhanced to specify the default values. This should help.

Moving along

I hope these questions and answers can help you when you start to migrate to z/OS V1R12. This is Professor Kimura signing off for now!