IBM Education Assistance for z/OS V2R2

Item: Tape Performance and Recovery New Function
Conversion of Tape Installation Exits to Dynamic Exits
RAS Enhancements

Element/Component: DFSMSdfp Open/Close/End of Volume

Material current as of May 2015

Agenda

- Trademarks
- Presentation Objectives
- Overview
- Usage & Invocation
- Interactions & Dependencies
- Migration & Coexistence Considerations
- Presentation Summary
- Appendix

Presentation Objectives

- Cover these new enhancements:
- Tape Performance and Recovery New Function
- Conversion of Tape Installation Exits to Dynamic Exits
- RAS Enhancements:
  - IEAVNP16 and IFG0DEVS enhancements associated with processing parmlib member: DEVSUPxx
  - Include above the line CSECTs in IFG019RB in OCE in-core module table
  - Eliminate A13-18 while ensuring the initial mount is for the correct volume
  - Add JOBID and SYSPLEXID to SMF14/15 records

Overview – Tape Performance and Recovery New Function

- Problem Statement / Need Addressed
  - Tape drive Data Synchronization ensures the actual medium is synchronized with the drive cache via moving the cached data to tape. After a synchronization has occurred, tape drive movement comes to a halt. When more data needs to written to the medium, the drive must regain servo track acquisition. This typically involves some backward motion which gives the drive sufficient time and distance to achieve recording speed. This backward motion and reacquisition of servo tracks can significantly degrade tape write performance.
  - Recovery of a multi-file synchronization failure requires rewriting all files since there is no way to determine what files were safely written to the tape medium.
- Solution
  - This new function provides a mechanism at the application level to:
    - Control synchronization events
    - Determine the specific compromised files in the event of a synchronization failure
- Benefit / Value
  - Better performance and Recovery

Usage & Invocation – Tape Performance and Recovery New Function

- Set a new DCBE parameter SYNCH=(NUMFILES,nn)
  - Should be positive and 32000 is the maximum
  - Value 'nn' is stored in new field DCBENMFL
- Example:
  If OPEN processing for tape is passed a DCBE with DCBENMFL value 100, up to a hundred consecutive files can be written specifying JCL PASS RETAIN or CLOSE LEAVE before an explicit synchronize channel program is issued. (These options leave the tape positioned at end of file just written.)

```
DCBAD DCB   DDNAME=DD1,DSORG=PS,MACRF=PM,BLKSIZE=80,RECFM=F,DCBE=DCBEB
DCBEB DCBE RMODE31=NONE,SYNC=(NUMFILES,100)
```

Usage & Invocation – Tape Performance and Recovery New Function

- The system service keeps track of each file, as well as the number of blocks within each file, on a file boundary basis. A file boundary in this case being a tape mark. All data, including tape marks are buffered in the device cache. After 'n' files have been written to cache, a single explicit synchronize will be done in order to move the files from the cache to the tape medium. In the event of a synchronization failure, the following message will be written to the system and job logs. The job will then be abended.

```
IEC999I jobname,stepname,volser LOST BLOCKS START: FILE 'n'
```

- Recovery rewrite should start at the file sequence in IEC999I message. The associated data set name can be found in IEC205I message previously written during CLOSE output processing and in the abend message (joblog/syslog).

Overview – Conversion of Tape Installation Exits to Dynamic Exits

- Problem Statement / Need Addressed
  - Current design of DFSMS Open/Close/EOV Installation Tape exits requires IPL for changes to be put into effect. There is no way to add multiple exit routines for the exit.
- Solution
  - All DFSMS Open/Close/EOV Installation Tape exits converted to be dynamic. All exits are AMODE 31, reentrant and persistent until the next IPL.
- Benefit / Value
  - Useability:
    - No need to IPL to manage changes with Tape Installation exits
    - Multiple exit routines might be associated with each exit

Usage & Invocation – Conversion of Tape Installation Exits to Dynamic Exits

Exit name	Default Exit Routine
OCE_VOLUMEMOUNT	IFG019VM
OCE_FILESTART	IFG019FS
OCE_FILEVALIDATE	IFG019FV
OCE_FILEEND	IFG055FE
OCE_LABELANOMALY	IFG019LA

- Permanent exit routines to be associated with exits during IPL are defined in PROGxx parmlib member:

```
EXIT ADD
     EXITNAME(OCE_VOLUMEMOUNT)
     MODNAME(MYEXIT1)
     STATE(ACTIVE)
     DSNAME(MY.PDS)
     FIRST
EXIT ADD
     EX(OCE_FILESTART)
     MOD(MYEXIT2)
     LAST
EXIT DELETE
     EX(OCE_FILEEND)
     MOD(IFG055FE)
```

Usage & Invocation – Conversion of Tape Installation Exits to Dynamic Exits - continued

- To display Open/Close/EOV tape installation exits:
  ```
  D PROG,EXIT,EXITNAME=OCE_*
  ```
- To display diagnostic data including exit routines for Volume Mount:
  ```
  D PROG,EXIT,EX=OCE_VOLUMEMOUNT,DIAG
  ```
- To add MYEXIT1 exit routine resided in MY.PDSE data set to File Start exit:
  ```
  SETPROG EXIT,ADD,EX=OCE_FILESTART,MODNAME=MYEXIT1,DSN=MY.PDSE
  ```
- To add and make MYEXIT2 exit routine to be called first every time once File End exit gets control:
  ```
  SETPROG EXIT,ADD,EX=OCE_FILEEND,MOD=MYEXIT2,FIRST
  ```
- To make MYEXIT3 exit routine inactive for File Validate exit:
  ```
  SETPROG EXIT,MODIFY,EX=OCE_FILEVALIDATE,MOD=MYEXIT3,STATE=INACTIVE
  ```
- To delete MYEXIT4 exit routine from File End exit:
  ```
  SETPROG EXIT,DELETE,EX=OCE_FILEEND,MODNAME=MYEXIT4
  ```

Usage & Invocation – Conversion of Tape Installation Exits to Dynamic Exits - continued

Rules for parameter list and execution

- Exit routines are called in unpredictable order unless they were defined with FIRST or LAST options.
- If multiple exit routines are defined then every exit routine is eligible to make changes in parameter list. Every subsequent exit routine will get the parameter list with possible changes from the previous one.
- In all cases parameter list contains last changes from the last called exit routine, no matter which one was called and which RC it returned.
- If the exit routine ABENDed then it is made inactive immediately and stay inactive until it is activated manually or the system is Re-IPL'ed.

Usage & Invocation – Conversion of Tape Installation Exits to Dynamic Exits - continued

Return code decisions rules when multiple exits called

- Label Anomaly exit:
  - If any exit routine returns RC4 (accept the volume) then return RC4 and the volume is accepted.
  - Else if any exit routine returns RC8 (reject the volume) then return RC8 and the volume is rejected.
  - Else return RC12 and an appropriate ABEND will be issued.
- Volume Mount and File Validation exits:
  - If any exit routine returns RC0 (accept the volume) then return RC0 and the volume is accepted.
  - Else if any exit routine returns RC8 (reject the volume) then return RC8 and the volume is rejected.
  - Else return RC4 and continue processing.
- File Start and File End exits:
  - Return RC0.

All other RCs are treated as incorrect and appropriate ABEND is issued in this case. All RCs are processed only after every exit routine associated with the exit gets control.

Overview – RAS Enhancements: IEAVNP16 and IFG0DEVS Enhancements

- Problem Statement / Need Addressed
  - Currently there is a subset of DEVSUPxx PARMLIB member keywords that are initially reset to their default values before processing the member. However, other keywords and their values are left intact. This creates an inconsistency in how all keywords are processed. When processing DEVSUPxx member during IPL or when updating a current DEVSUPxx member via system command: `SET DEVSUP=xx` this inconsistency forces the requirement that all keywords and their values be specified, rather than only the keywords that are to be changed.
- Solution
  - From now on all keywords will be treated in a consistent manner so that none will be initially reset to their default values both during IPL and with processing SET DEVSUP command. The `SET DEVSUP=xx` target can specify only the keyword(s) to be changed.
- Benefit / Value
  - Usability

Overview – RAS Enhancements: IEAVNP16 and IFG0DEVS Enhancements - continued

- Problem Statement / Need Addressed
  - Currently continued processing of the DEVSUP member stops at the point where a keyword error is detected and `IEA252E DEVSUP: INVALID SYNTAX. MEMBER PROCESSING TERMINATED ON LINE yyyy.` message is issued. There is no indication which DEVSUP member is processed.
- Solution
  - If syntax error is detected system continues processing member and IEA252E message is issued: `IEA252E DEVSUPxx: INVALID SYNTAX ON LINE yyyy. MEMBER PROCESSING CONTINUES.`
- Benefit / Value
  - Usability

Overview – RAS Enhancements: IEAVNP16 and IFG0DEVS Enhancements - continued

- Problem Statement / Need Addressed
  - SET DEVSUP MVS command can only support single value: `SET DEVSUP=xx`.
  - IEASYSxx PARMLIB member supports only single value for DEVSUP keyword in IEASYSxx member.
- Solution
  - MVS system command syntax enhanced with multiple values: `SET DEVSUP=(xx,yy,zz…)`
  - IEASYSxx PARMLIB member now supports multiple values for DEVSUP keyword: `DEVSUP=(xx,yy,zz…)`
- Benefit / Value
  - Usability

Overview – RAS Enhancements: IEAVNP16 and IFG0DEVS Enhancements - continued

- Problem Statement / Need Addressed
  - If `SET DEVSUP=xx` command is issued, but no DEVSUPxx member exists, then `IEE536I DEVSUP VALUE xx NOW IN EFFECT` message is issued.
- Solution
  - No `IEE536I DEVSUP VALUE xx NOW IN EFFECT` message is issued if no DEVSUPxx member exists.
- Benefit / Value
  - Usability

Usage & Invocation – RAS Enhancements: IEAVNP16 and IFG0DEVS Enhancements

- IEASYSxx PARMLIB member DEVSUP keyword new syntax support:
  `DEVSUP=(xx,yy,zz...)`
- Multiple DEVSUPxx PARMLIB members specification support in one MVS command:
  `SET DEVSUP=(xx,yy,zz...)`
- In case of syntax error in DEVSUPxx PARMLIB member:
  `NON_VSAM_XTIOT=TES,`
  The following messages are issued:

```
IEA251E DEVSUP8A: NON_VSAM_XTIOT PARAMETER 'TES' NOT
VALID.
IEA252E DEVSUP8A: INVALID SYNTAX ON LINE 0002. MEMBER
PROCESSING CONTINUES.
```

Overview – RAS Enhancements: Include above the line CSECTs in OCE in-core module table

- Problem Statement / Need Addressed
  - OCE maintains an in-core module table whose entries consist of: CSECT name, LPA address and PTF level. Currently none of 12 CSECTs in IFG019RB are included in this table.
- Solution
  - 12 above the line CSECTs of IFG019RB module will be included in in-core module table.
- Benefit / Value
  - Elimination further requests from Level 2 for PTF's levels.

Usage & Invocation - RAS Enhancements: Include above the line CSECTs in OCE in-core module table

A section of this table in storage follows:

```
C9C6C7F0   F2F0F0E5   E4C1F9F0   F6F0F540   | IFG0200VUA90605
00D36138   C9C6C7F0   F2F0F2D3   E4C1F6F5   | .L/.IFG0202LUA65
F9F8F240   00D40318   C9C6C7F0   F2F0F9D3   | 982 .M..IFG0209L
E4C1F6F5   F9F8F240   00D40318   C9C6C7F0   | UA65982 .M..IFG0
F1F9F4C1   E4C1F6F6   F8F9F440   00D61000   | 194AUA66894 .O..
C9C6C7F0   F1F9F4C3   D5D6D5C5   40404040   | IFG0194CNONE
00D63220   C9C6C7F0   F1F9F4C4   E4C1F6F0   | .O..IFG0194DUA60
F1F6F640   00D64180   C9C6C7F0   F1F9F4C5   | 166 .O..IFG0194E
D5D6D5C5   40404040   00D66520   C9C6C7F0   | NONE    .O..IFG0
F1F9F5C1   E4C1F6F3   F6F0F440   00D4B910   | 195AUA63604 .M..
```

Overview – RAS Enhancement: Eliminate A13-18 while ensuring correct mount

- Problem Statement / Need Addressed
  - An abendA13 RC18 is detected when an EOV1 label is read on the last SL or AL tape volume while forward spacing to the desired file or just before the desired file. Here are 2 possible conditions:
    - If opening to the end of the file, it could not be treated as the end of the data set because it was for a previous file sequence number. Probable user error.
    - If the user is trying to extend a data set that had abended during EOV, it is necessary to specify a volume count on the DISP=MOD DD statement. This volume count must exceed the number of known volumes for the data set.
  - Another scenario leading to this abend is when there are multiple DDs for tape data sets to be created in the same step. The first DD contains a specific volser and the subsequent DD's specify a VOLREF back to the first DD.
  - When one of the data sets being written overflows to another volume, that volume is not known to the subsequent DD's in the job step. This results in the subsequent detection of an ABENDA13 rc18.
- Solution
  - This item will ensure that during OPEN processing for one of the subsequent DD’s, that the new last volume is ‘discovered’ and then selected in the initial OPEN mount request. This will occur when stacking tape files (ie. writing multiple consecutive file sequence tape data sets while leaving the tape positioned and the end of each file as it is created).
- Benefit / Value
  - Usability

Usage & Invocation – RAS Enhancement: Eliminate A13-18 while ensuring correct mount

- Example:

```
//JOB
//STEP1 EXEC
//DD1  DD DSN=DATASET1,
          DISP=(NEW,PASS,KEEP),
          VOL=SER=111111,
          UNIT=920,
          LABEL=(1,SL)
//DD2  DD DSN=DATASET2,
          DISP=(NEW,PASS,KEEP),
          VOL=REF=*.STEP1.DD1,
          UNIT=920,
          LABEL=(2,SL)
```

DD1 spills to scratch volume 222222. When DD2 is processed open replaces 111111 with 222222 in the JFCB.

Overview – RAS Enhancement: Add JOBID & SYSPLEXID to SMF 14/15

- Problem Statement / Need Addressed
  - JCTJOBID and Sysplex should be added because it's often important to understand where a job ran for charge back or capacity planning.
  - The JCTJOBID is needed because it's possible to have duplicate job names with the same reader timestamp.
- Solution
  - JOBID and SYSPLEXID are now added to SMF 14/15.
- Benefit / Value
  - Usability

Usage & Invocation – RAS Enhancement: Add JOBID & SYSPLEXID to SMF 14/15

- None. When SMF14/15 records are requested the new fields are automatically included.

Migration & Coexistence Considerations

- None unless you have a message automation product that will be affected by IEA251E and IEA252E messages text changes.

Presentation Summary

- Tape Performance and Recovery can be improved
- Dynamic Tape Installation Exits can be used to avoid system's IPL
- The `SET DEVSUP=xx` target can specify only the keyword(s) to be changed
- If syntax error is detected system continues processing `DEVSUPxx` member and `IEA252E` message indicates the place of error:
  `IEA252E DEVSUPxx: INVALID SYNTAX ON LINE yyyy. CONTINUE MEMBER PROCESSING.`
- MVS system command syntax enhanced with multiple values:
  `SET DEVUP=(xx,yy,zz...)`
- IEASYSxx PARMLIB member now supports multiple values for DEVSUP keyword:
  `DEVSUP=(xx,yy,zz...)`
- 12 above the line CSECTs in IFG019RB are now included in OCE in-core table.
- A13-18 is now eliminated while ensured correct mount
- SMF 14/15 records now contain JOBID & SYSPLEXID

Appendix

- *z/OS DFSMS Macro Instructions for Data Sets, SC23-6852*
- *z/OS DFSMS Installation Exits, SC23-6850*
- *z/OS MVS System Commands, SA38-0666*
- *z/OS MVS Initialization and Tuning Reference, SA23-1380*
- *z/OS MVS System Messages, Vol 6 (GOS-IEA), SA38-0673*