




Small Enhancements You Might Have Missed in z/OS

Best of 2018!

Marna WALLE, mwalle@us.ibm.com
Member of the IBM Academy of Technology 
z/OS System Installation
IBM Z Systems, Poughkeepsie, NY, USA

Trademarks

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

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

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

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

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

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

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed.

Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

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

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

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

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

z/OS Small Enhancements - Best of 2018!

- **z/OS V2.3:**
 - **RACF: IRRPRMxx**
 - **z/OS UNIX: BPXWMIGF facility**
- **z/OS V2.2:**
 - **z/OS UNIX: zlsnf updates (with jsonprint)**
 - **SDSF: snapshot**
 - **SDSF: System Command Extension pop-up**
 - **BCP PROGxx: LPA Volser**
 - **BCP Dynamic APF: SMF 90-37**
- **z/OS V2.1:**
 - **DFSMSdfp and ICKDSF: Protection for initializing non-empty volumes**
 - **DFSMSdss: Renaming of VSAM physical data sets**
- **Older than the hills:**
 - **SMP/E: Automatic cross-zone requisite checking**



z/OS V2R3 Small Enhancements

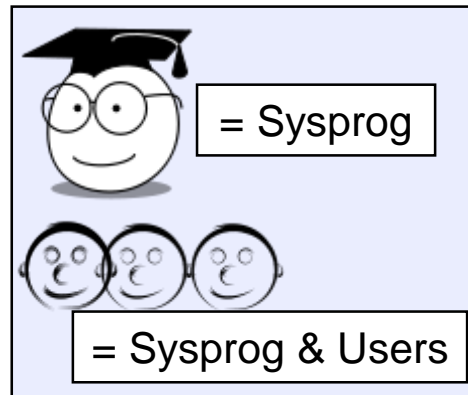


❖ RACF: IRRPRMxx

❖ Prepare for your first V2.3 IPL!



❖ z/OS UNIX: BPXWMIGF facility





z/OS V2.3:

RACF: IRRPRMxx

What: Parmlib member for RACF data set name table and range table specification!

- IEASYSxx RACF=yy, points to your IRRPRMyy member(s)
 - Each section can be in its own member, but not split over two. Maximum of 3 members.
- You might have one less usermod!
- Accompanying V2.3 TSO command, **RACPRMCK** to verify syntax.

Considerations:

- The **DSNT2PRM** tool can help you create a new IRRPRMxx parmli member.
 - Retrieve tool (and doc) from the RACF Downloads web site, <https://www-03.ibm.com/systems/z/os/zos/features/racf/downloads/dsn2prm.html>
 - Comments on tool should be directed to RACF-L mailing list.
- IRRPRMxx and RACPRMCK are available **on z/OS V2.3 and higher**.
 - DSNT2PRM + RACPRMCK = good practice.



z/OS V2.3:

RACF: IRRPRMxx

My trial run on z/OS V2.1...

Invocation of tool: `ex 'mwalle.clist(dsnt2prm)' 'mwalle.util.jobs2(dsnt2out)'`

DCU004I Generate PARMLIB data based on ICHRDSNT data.

of DS = 3

DCU005I Generate PARMLIB data based on ICHRRNG data.

Ranges = 3

DCU105I INFO: Verify the generated output using the RACPRMCK command.

DCU106W WARNING: DSNT2PRM running on V2R2 release or lower.

DCU104W WARNING: Using current in-storage Data Set Name Table values. These values may NOT match what you IPLed with.

DCU002W Successful execution of DSNT2PRM, with WARNINGS! Return code = 4

- I tried this on V2.1, just to see what my *possible* IRRPRMxx would look like for V2.3.
- In addition to in-memory, it can also take load module(s) as input!
- The “not matching” warning gives you a heads-up to any RVAR commands that might have been issued. Look carefully at the produced IRRPRMxx to make sure it is desirable.



z/OS V2.3:

RACF: IRRPRMxx

Parmlib Member Output:

```

/* -----
--
-- This PARMLIB member was generated on 02/05/18
-- by the DSNT2PRM utility on system ST6.
--
-- In-Storage version of ICHRDSNT & ICHRRNG were used
-- to generate this PARMLIB member.
--
----- */
DATABASE_OPTIONS
/* ----- */
SYSPLEX(DATASHARING)
DATASETNAMETABLE
ENTRY
  PRIMARYDSN('SYS1.RACFP01')
  BACKUPDSN('SYS1.RACFB01')
  UPDATEBACKUP(ALL)
  BUFFERS(255)
ENTRY
  PRIMARYDSN('SYS1.RACFP02')
  BACKUPDSN('SYS1.RACFB02')
  UPDATEBACKUP(ALL)
  BUFFERS(255)
ENTRY
  PRIMARYDSN('SYS1.RACFP03')
  BACKUPDSN('SYS1.RACFB03')
  UPDATEBACKUP(ALL)
  BUFFERS(255)

```

```

/* ----- */
RANGETABLE
  START('00' HEX)
    ENTRYNUMBER(1)
  START('U71' CHAR)
    ENTRYNUMBER(2)
  START('U80' CHAR)
    ENTRYNUMBER(3)

```



z/OS V2.3:

RACF: IRRPRMxx

Sanity check vs. my trial run on V2.1:

RVARY LIST

RACF DATABASE STATUS:

ACTIVE USE NUM VOLUME DATASET

YES PRIM 1 RACFS1 SYS1.RACFP01

YES BACK 1 RACFS1 SYS1.RACFB01

YES PRIM 2 RACFS1 SYS1.RACFP02

YES BACK 2 RACFS1 SYS1.RACFB02

YES PRIM 3 RACFS1 SYS1.RACFP03

YES BACK 3 RACFS1 SYS1.RACFB03

MEMBER ST6 IS SYSPLEX COMMUNICATIONS ENABLED & IN DATA SHARING MODE.

RVARY COMMAND HAS FINISHED PROCESSING.

Re-ran DSNT2PRM on V2.3, then final verification before use, on V2.3:

Enter TS0 or Workstation commands below:

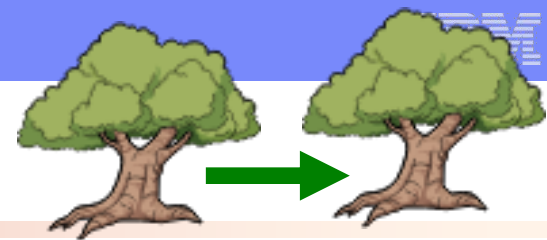
==> racprmck member(testracf)

IRRY301I No errors found in PARMLIB member(s).

This command runs using the contents of the current parmlib concatenation member you say.

z/OS V2.3:

z/OS UNIX: BPXWMIGF facility



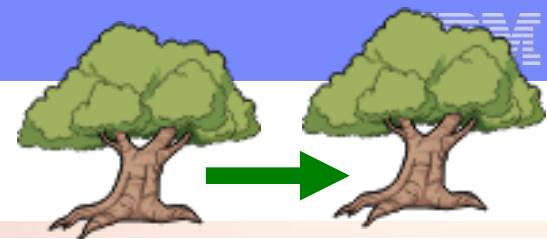
What: New tool for converting HFS to zFS for high availability file systems.

- Available from TSO, z/OS UNIX shell, and via SYSREXX (console)
- HFS does not need to be unmounted. Can be RO or RW.
- Two phases: 1) mirror data and maintain, 2) swap, when ready.

Many Considerations:

- all systems in OMVS group must be V2.3 – no downlevels,
- unmounting or moving ownership cancels migration,
- only HFS -> zFS, and only one migration at a time.
- Superuser or SUPERUSER.FILESYS.PFSCCTL auth,
- ~~zFS must not be in the OMVS address space.~~ Restriction removed with OA53128!
- **Extreme caution to ensure new zFS is mounted after a swap and not the old HFS,**
- ...

Read about them in *z/OS UNIX: Planning*, and *z/OS UNIX Command Reference*.



z/OS V2.3:

z/OS UNIX: BPXWMIGF facility

How to use, one scenario:

At my /busyfs mountpoint, my HFS needs high availability:

—	—/home/mwalle	OS390AT.ZFS.MWALLE	ZFS
—	—/home/mwalle/busyfs	MWALLE.BUSY.TESTFS	HFS

Where, the file “always.needed”:

```
Pathname . . . : /home/mwalle/busyfs/always.needed

General Data
File Type . . . : File
File Size . . . : 22063104
Links . . . . . : 1
Inode . . . . . : 4
File Format . . . : ----
Last Modified . . : 2017/07/07 15:07:11
Last Changed . . . : 2017/07/07 15:07:11
Last Accessed . . : 2017/07/07 15:07:11
Created . . . . . : 2017/07/07 15:07:11
CCSID . . . . . :
Text Convert . . . : NO

Mode Fields
Permissions . . . : 644
Set User ID . . . : NO
Set Group ID . . . : NO
Sticky Bit . . . . : NO

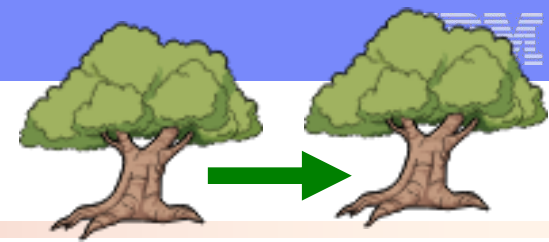
Extended Attributes
Shared AS . . . . : YES
APF Auth . . . . . : NO
Pgm Control . . . . : NO
Shared Lib . . . . . : NO

Audit
Auditor . . . . . : ---
User . . . . . : fff

Device Data
Device Number . . : 1EC7
Major Device . . . :
Minor Device . . . :
```

z/OS V2.3:

z/OS UNIX: BPXWMIGF facility



My zFS replacement is allocated, V5 formatted, is proper size, and is not mounted.

No migrations are ongoing for that HFS:

```
-F AXR,BPXWMIGF -QUERY  
BPXWMG017I no migrations found  
BPXWMG019I end of output
```

Example: as
SYSREXX
command

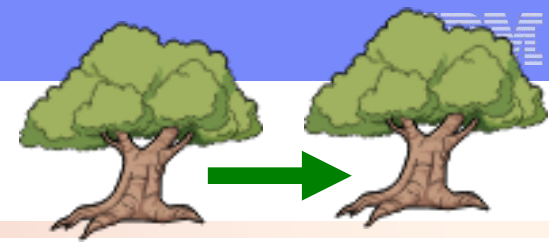
Try to migrate #1, without a swap when done. One system was downlevel.

```
# bpxwmigf -source mwalle.busy.testfs -target mwalle.busy.zfs-srename mwalle.bu  
y.old -trename mwalle.busy.new -noswap  
BPXWMG099I pfscctl error -1 79 11B30682  
BPXVFPCT 06/26/17  
JRMigDownLevel: A system in the sysplex is at a lower release level that does  
not support file system migration.  
  
Action: Retry the migration when the down-level system is not a member of the  
SYSBPX sysplex group.
```

Example: as shell
command

z/OS V2.3:

z/OS UNIX: BPXWMIGF facility

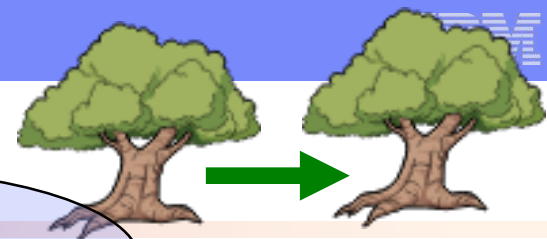


Query shows failure, and info on the attempted migration:

```
# bpxwmigf -query
MWALLE.BUSY.TESTFS
  status.....: failed at 22:15:35 07/06/2017 GMT
  failed reason: 11B30682 JRMigDownLevel: A system in the sysplex is at a lower
release level that does not support file system migration.
  started.....: 22:15:35 07/06/2017 GMT
  user.....:
  target name..:
  source rename: no
  rename target: no
  mount mode...: same
  mount parms..:

BPXWMG019I end of output
#
```

...(z/OS V2.2 system was then upgraded to V2.3.)



z/OS V2.3:

z/OS UNIX: BPXWMIGF facility

Try to migrate #2 :

Example: as
TSO/E command

ISPF Command Shell

Enter TSO or Workstation commands below:

```
==> BPXWMIGF -source mwalle.busy.testfs -target mwalle.busy.zfs -srename mwall  
e.busy.old -trename mwalle.busy.new -noswap -priority 1
```

```
BPXWMG099I pfscctl error -1 79 11B30668
```

```
BPXVFPCT 06/26/17
```

```
JRMigNotLocal: The source file system is not mounted locally
```

Action: Move the file system so that it is mounted locally or initiate the migration request from the owning system.

```
*** _
```

...logged onto the owning system. Try to migrate #3:

```
BPXWMG099I pfscctl error -1 79 11B30689
```

```
BPXVFPCT 06/26/17
```

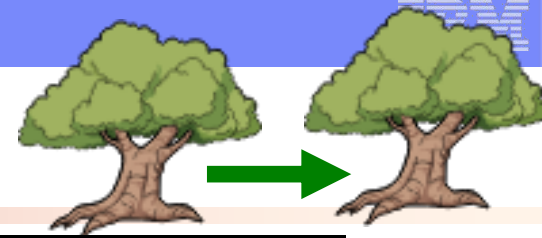
```
JRMigNotColony: The target physical file system must be running in a colony  
address space.
```

```
***
```

Advice: Make sure you have the PTF for OA53128 installed so that you can put zFS in the OMVS address space (V2.2), and use *bpxwmigf*.

z/OS V2.3:

z/OS UNIX: BPXWMIGF facility



Try to migrate #4 :

```
MWALLE.BUSY.TESTFS
status.....: mirroring 14% complete
started.....: 19:26:29 07/07/2017 GMT
user.....: MWALLE
```

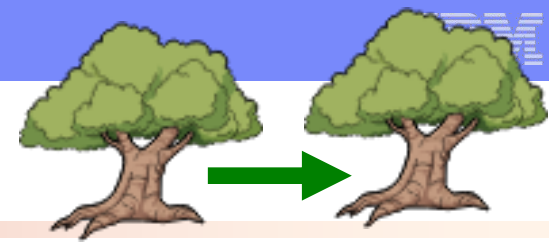
```
target name..: MWALLE.BUSY.ZFS
source rename: MWALLE.BUSY.OLD
rename target: MWALLE.BUSY.NEW
mount mode...: same
mount parms...:
auto-swap....: no
priority.....: 1
```

Example: as
TSO/E command

...finally a migration success! (Any error would have cancelled the migration)

```
-F AXR,BPXWMIGF -QUERY
MWALLE.BUSY.TESTFS
status.....: mirror complete at 19:26:29 07/07/2017 GMT
started.....: 19:26:29 07/07/2017 GMT
user.....: MWALLE
target name..: MWALLE.BUSY.ZFS
source rename: MWALLE.BUSY.OLD
rename target: MWALLE.BUSY.NEW
mount mode...: same
mount parms...:
auto-swap....: no
```

BPXWMG019I end of output



Swap:

```
# bpxwmigf -source mwalle.busy.testfs -swap
MWALLE.BUSY.TESTFS
status.....: swap initiated at 20:25:28 07/07/2017 GMT
started.....: 19:26:29 07/07/2017 GMT
user.....: MWALLE
target name..: MWALLE.BUSY.ZFS
source rename: MWALLE.BUSY.OLD
rename target: MWALLE.BUSY.NEW
mount mode...: same
mount parms..:

BPXWMG019I end of output
```

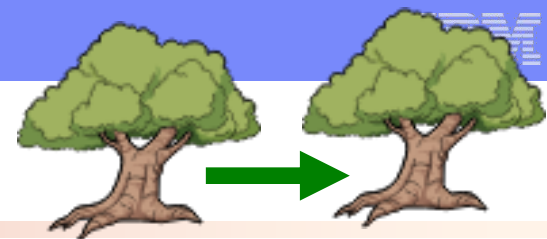
Access to the file system is very briefly quiesced during the swap, which is transparent to applications.

Now, I carefully verify that “rename target” data set is to mount correctly from now on in my BPXPRMxx or policies.

■ Use of `-srename` helped me avoid mounting the old one.

z/OS V2.3:

z/OS UNIX: BPXWMIGF facility



Final verification:

```
-F AXR,BPXWMIGF -QUERY
MWALLE.BUSY.TESTFS2
status.....: mirror complete at 19:41:34 07/07/2017 GMT
started.....: 19:41:33 07/07/2017 GMT
user.....: BPXROOT
target name..: MWALLE.BUSY.ZFS2
source rename: MWALLE.BUSY.OLD2
rename target: MWALLE.BUSY.NEW2
mount mode...: same
mount parms...:
auto-swap....: no

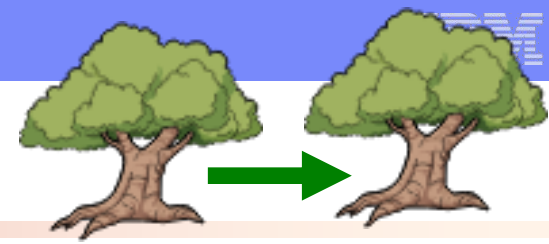
.
MWALLE.BUSY.TESTFS
status.....: completed at 20:25:30 07/07/2017 GMT
started.....: 19:26:29 07/07/2017 GMT
user.....: MWALLE
target name..: MWALLE.BUSY.ZFS
source rename: MWALLE.BUSY.OLD
rename target: MWALLE.BUSY.NEW
mount mode...: same
mount parms...:

.
BPXWMG019I end of output
```

Example of another swap to
do later

```
-MWALLE.BUSY.**
MWALLE.BUSY.NEW          /home/mwalle/busyfs
MWALLE.BUSY.TESTFS2      /home/mwalle/busyfs2
```

ZFS
HFS



And yet more verification:

```
Pathname . . . : /home/mwalle/busyfs/always.needed

General Data                                     Mode Fields
File Type . . . : File                         Permissions . . : 644
File Size . . . : 22063104                     Set User ID . . : NO
Links . . . . . : 1                           Set Group ID . . : NO
Inode . . . . . : 4                           Sticky Bit . . . : NO
File Format . . : ----
Last Modified . : 2017/07/07 15:07:11           Extended Attributes
Last Changed . . : 2017/07/07 15:07:11         Shared AS . . . : YES
Last Accessed . . : 2017/07/07 15:26:29        APF Auth . . . . : NO
Created . . . . . : 2017/07/07 15:26:28        Pgm Control . . : NO
CCSID . . . . . :                               Shared Lib . . . : NO
Text Convert . . : NO

Owner
File . . . . . : MWALLE(9268)
Group . . . . . : OPERATOR(0)

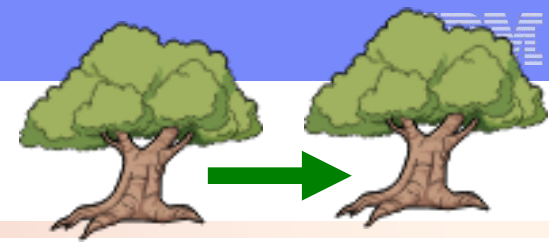
Audit
Auditor . . . . : ---
User . . . . . : fff

File Data
Number . . . . . : 1EC7
Device . . . . . :
Min . . . . . :
```

Only changes on target

z/OS V2.3:

z/OS UNIX: BPXWMIGF facility



Cancelling a migration, before swap is done:

```
Enter TSO or Workstation commands below:
```

```
==> BPXWMIGF -cancel mwalle.busy.testfs2
```

```
MWALLE.BUSY.TESTFS2
```

```
status.....: cancelled by MWALLE at 21:16:03 07/07/2017 GMT
```

```
started.....: 19:41:33 07/07/2017 GMT
```

```
user.....: BPXROOT
```

```
*** _
```

BPXWMIGF – query
indicates one cancelled, and one completed, as expected.

z/OS V2R2

Small Enhancements



❖ z/OS UNIX: zlsf updates (with jsonprint)



❖ SDSF: snapshot



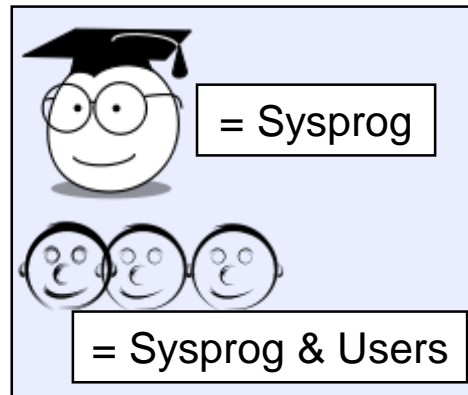
❖ SDSF: System Command Extension pop-up



❖ BCP PROGxx: LPA Volser



❖ BCP Dynamic APF: SMF 90-37





z/OS UNIX: zlsof updates

■ **What:**

- zlsof is a handy utility to look at open files, sockets, and pipes.
- Originally on the z/OS UNIX Tools and Toys website
- z/OS V2.1: moved into z/OS /bin and enhanced (for instance with lock holders and waiters when the byte range lock manager is used)
- z/OS V2.2 with OA55246: additional enhancements for extended processing information, and generate output in JSON format.

■ **How to use:**

- Install PTFs, and use new zlsof options `-x` and `-json`.



z/OS UNIX: zlsf updates

Without using new functions:

```
$ zlsf
zlsf version=180606
Searching for all file usage
Command      PID User   File System      Mountpoint      Inode/file
-sh          50594018 MWALLE OMYS.ZFS.COMBAT.SYSPLEX.ROOT /                r 1
              OS390AT.ZFS.MWALLE /home/mwalle    c 1
              OMYS.TFS.DEV.CB8B /CB8B/dev       8234 /dev/ttyp0001
              OMYS.TFS.DEV.CB8B /CB8B/dev       8234 /dev/ttyp0001
              OMYS.TFS.DEV.CB8B /CB8B/dev       8234 /dev/ttyp0001
              OMYS.TFS.DEV.CB8B /CB8B/dev       8234 /dev/ttyp0001
              OS390AT.ZFS.MWALLE /home/mwalle    32 /home/mwalle/.sh_history
OMYS         33816804 MWALLE OMYS.ZFS.COMBAT.SYSPLEX.ROOT /                r 1
              OS390AT.ZFS.MWALLE /home/mwalle    c 1
              OMYS41.ZFS.MYSBUILD.VERSION.CMRS41 /CMRS41         23642 /usr/lib/nls/msg/C/fsumucat.cat
              OMYS.TFS.DEV.CB8B /CB8B/dev       8233 /dev/ptyp0001

zlsf End of output
$
```

zlsf default output for an unauthorized invoker consists of open file information for processes that are associated with the user. If the invoker is authorized, the default output consists of open file information for all processes in the system.



z/OS V2R2 Enhancements

z/OS UNIX: zlsnf updates

Using new -x option:

```
zlsnf version=180606
Searching for all file usage
Command      PID User      File System      Mountpoint      Inode/file/process info
-sh          50594018 MWALLE(9268)      OMYS.ZFS.COMBAT.SYSPLEX.ROOT /                r 1
State: MULPROCESS                               OS390AT.ZFS.MWALLE /home/mwalle    c 1
                                                    OMYS.TFS.DEV.CB8B /CB8B/dev       rw 8234 /dev/tty0001
                                                    OMYS.TFS.DEV.CB8B /CB8B/dev       rw 8234 /dev/tty0001
                                                    OMYS.TFS.DEV.CB8B /CB8B/dev       rw 8234 /dev/tty0001
                                                    OMYS.TFS.DEV.CB8B /CB8B/dev       rw 8234 /dev/tty0001
                                                    OS390AT.ZFS.MWALLE /home/mwalle    32 /home/mwalle/.sh_history
OMYS        33816804 MWALLE(9268)      OMYS.ZFS.COMBAT.SYSPLEX.ROOT /                r 1
                                                    OS390AT.ZFS.MWALLE /home/mwalle    c 1
                                                    OMYS41.ZFS.MYSBUILD.VERSION.CHRS41 /CHRS41         rd 23642 /usr/lib/nls/msg/C/fsumucat.cat
                                                    OMYS.TFS.DEV.CB8B /CB8B/dev       rw 8233 /dev/pty0001

zlsnf End of output
```

Shows extended process information. The information includes UID with the user name, start time, elapsed time, CPU time, ppid, thread number, controlling TTY information, state of the process, and read/write open mode.

z/OS UNIX: zlsf updates



Using new `–json` option:

```
zlsf -json > myzlsf.json; cat myzlsf.json
{"utility":"zlsf","version":"180606","request":"Searching for all file usage","result":[{"command":"-sh","commandLine":"-sh","userId":"MWALLE","uid":"9268","job":"MWALLE","asid":"10F","pid":"50594018","ppid":"33816804","startTime":"1532382323","cpuTime":"13","contty":"\\dev\\tty0001","threads":"1","state":"MULPROCESS","files":[{"type":"root","openFlags":"0","devno":"1","fileSystem":"OMVS.ZFS.COMBAT.SYSPLEX.ROOT","mountPath":"\\","inum":"1","diagName":null,"pathName":null},{type":"cwd","openFlags":"0","devno":"350","fileSystem":"OS390AT.ZFS.MWALLE","mountPath":"\\home\\mwalle","inum":"1","diagName":null,"pathName":null},{type":"charSpec","openFlags":"35","devno":"203","fileSystem":"OMVS.TFS.DEV.CB8B","mountPath":"\\CB8B\\dev","inum":"8234","diagName":"\\dev\\tty0001","pathName":null},{type":"regularFile","openFlags":"145","devno":"350","fileSystem":"OS390AT.ZFS.MWALLE","mountPath":"\\home\\mwalle","inum":"63","diagName":"myzlsf.json","pathName":null},{type":"charSpec","openFlags":"35","devno":"203","fileSystem":"OMVS.TFS.DEV.CB8B","mountPath":"\\CB8B\\dev","inum":"8234","diagName":"\\dev\\tty0001","pathName":null},{type":"charSpec","openFlags":"35","devno":"203","fileSystem":"OMVS.TFS.DEV.CB8B","mountPath":"\\CB8B\\dev","inum":"8234","diagName":"\\dev\\tty0001","pathName":null},{type":"regularFile","openFlags":"139","devno":"350","fileSystem":"OS390AT.ZFS.MWALLE","mountPath":"\\home\\mwalle","inum":"32","diagName":"\\home\\mwalle\\.sh_history","pathName":null},{type":"charSpec","openFlags":"35","devno":"203","fileSystem":"OMVS.TFS.DEV.CB8B","mountPath":"\\CB8B\\dev","inum":"8234","diagName":"\\dev\\tty0001","pathName":null}]}],{"command":"OMVS","commandLine":"OMVS","userId":"MWALLE","uid":"9268","job":"MWALLE","asid":"10F","pid":"33816804","ppid":"1","startTime":"1532382323","cpuTime":"13","contty":null,"threads":"2","state":"MULPROCESS","files":[{"type":"root","openFlags":"0","devno":"1","fileSystem":"OMVS.ZFS.COMBAT.SYSPLEX.ROOT","mountPath":"\\","inum":"1","diagName":null,"pathName":null},{type":"cwd","openFlags":"0","devno":"350","fileSystem":"OS390AT.ZFS.MWALLE","mountPath":"\\home\\mwalle","inum":"1","diagName":null,"pathName":null},{type":"regularFile","openFlags":"2","devno":"9","fileSystem":"OMVS41.ZFS.MYSBUILD.VERSION.CMRS41","mountPath":"\\CMRS41","inum":"23642","diagName":"\\usr\\lib\\nls\\msg\\C\\fsu_mucacat.cat","pathName":null},{type":"charSpec","openFlags":"7","devno":"203","fileSystem":"OMVS.TFS.DEV.CB8B","mountPath":"\\CB8B\\dev","inum":"8233","diagName":"\\dev\\pty0001","pathName":null}]}]}
```

Shows data in JSON format.

Very good for programs, but I'm human...



z/OS UNIX: zlsf updates (with jsonprint)

Let's pair this with new "Client Web Enablement Toolkit" json pretty print capability in OA55438!

```
$ zlsf -json | /samples/jsonprint
{
  "utility"      : "zlsf"
  "version"      : "180606"
  "request"      : "Searching for all file usage",
  "result": [
    {
      "command"      : "ISRBRO"
      "commandLine"  : "ISRBRO  "
      "userId"       : "MWALLE"
      "uid"          : "9268"
      "job"          : "MWALLE  "
      "asid"         : "119"
      "pid"          : "50596768"
      "ppid"         : "1"
      "startTime"    : "1532453379"
      "cpuTime"      : "537"
      "contty"       : null,
      "threads"      : "3"
      "state"        : "MULPROCESS"
      "files": [
```



z/OS UNIX: zlsof updates

■ Considerations:

- Newer zlsof options opens up more opportunities to pull even more data into programs to help know who is using what, and what is in use by whom.
- zlsof can be invoked from shell, TSO/E, or as system REXX (F AXR, ZLSOF) command.
- Client Web Enablement Toolkit's json pretty print REXX is found:
 - SYS1.SAMPLIB(HWTJSPRT) for TSO/E.
 - /samples/jsonprint (which is /samples/IBM/HWTJSPRT) for shell.



SDSF: snapshot

■ **What:**

- SNAPSHOT allows you to display the data from an SDSF tabular panel in a browse or edit session.
- You can then use SDSF's Print function to print it, or ISPF functions to copy it to a data set.

■ **How to use, on any tabular panel:**

- Format: `SNAPSHOT | SNAP (S | SB | SE | SV)`

■ **Considerations:**

- Nice if you wanted to do “fancier” ISPF commands, rather than simple sorting from the CK panel.

Thanks to Mike Shorkend for this suggestion!



SDSF: snapshot

Scenario: Find all the inactive z/OS migration health checks.

■ 1) SNAPSHOT SE

```

Display  Filter  View  Print  Options  Search  Help
-----
SDSF HEALTH CHECKER DISPLAY S1                LINE 1-17 (219)
COMMAND INPUT ==> snapshot se                SCROLL ==> HAL
PREFIX=*  DEST={ALL}  OWNER=MWALLE  SORT=NAME/A  SYSNAME=

NP  NAME                                CheckOwner  State  Sta
  ALLOC_ALLC_OFFLN_POLICY              IBMALLOC    ACTIVE (ENABLED)  SUC
  ALLOC_SPEC_WAIT_POLICY               IBMALLOC    ACTIVE (ENABLED)  SUC
  ALLOC_TAPELIB_PREF                   IBMALLOC    ACTIVE (DISABLED) ENV
  ALLOC_TIOT_SIZE                      IBMALLOC    ACTIVE (ENABLED)  SUC
  ASM_LOCAL_SLOT_USAGE                 IBMASM      ACTIVE (ENABLED)  SUC
  ASM_NUMBER_LOCAL_DATASETS            IBMASM      ACTIVE (ENABLED)  SUC
  ASM_PAGE_ADD                         IBMASM      ACTIVE (ENABLED)  SUC
  ASM_PLPA_COMMON_SIZE                 IBMASM      ACTIVE (ENABLED)  SUC
  ASM_PLPA_COMMON_USAGE                IBMASM      ACTIVE (ENABLED)  SUC
  CATALOG_ATTRIBUTE_CHECK              IBMCATALOG  ACTIVE (ENABLED)  SUC
  CATALOG_IMBED_REPLICATE              IBMCATALOG  ACTIVE (ENABLED)  SUC
  CATALOG_RNLS                         IBMCATALOG  ACTIVE (ENABLED)  SUC
  CICS_CEDA_ACCESS                    IBMCICS     ACTIVE (ENABLED)  EXC

```



SDSF: snapshot

Scenario: Find all the inactive z/OS migration health checks.

■ 2) Search for migration checks; find those INACTIVE

```
SDSF EDIT      *SNAP                      Columns 00001 00072
Command ==> x all;f zosmig all;x ' ACTIVE(ENABLED)' all      Scroll ==> HALF
***** Top of Data *****
```

000001	NAME	CheckOwner	State
000002	ALLOC_ALLC_OFFLN_POLICY	IBMALLOC	ACTIVE(ENABLED)
000003	ALLOC_SPEC_WAIT_POLICY	IBMALLOC	ACTIVE(ENABLED)
000004	ALLOC_TAPELIB_PREF	IBMALLOC	ACTIVE(DISABLED)
000005	ALLOC_TIOT_SIZE	IBMALLOC	ACTIVE(ENABLED)
000006	ASM_LOCAL_SLOT_USAGE	IBMASM	ACTIVE(ENABLED)
000007	ASM_NUMBER_LOCAL_DATASETS	IBMASM	ACTIVE(ENABLED)
000008	ASM_PAGE_ADD	IBMASM	ACTIVE(ENABLED)
000009	ASM_PLPA_COMMON_SIZE	IBMASM	ACTIVE(ENABLED)
000010	ASM_PLPA_COMMON_USAGE	IBMASM	ACTIVE(ENABLED)
000011	CATALOG_ATTRIBUTE_CHECK	IBMCATALOG	ACTIVE(ENABLED)
000012	CATALOG_IMBED_REPLICATE	IBMCATALOG	ACTIVE(ENABLED)
000013	CATALOG_RNLS	IBMCATALOG	ACTIVE(ENABLED)
000014	CICS_CEDA_ACCESS	IBMCICS	ACTIVE(ENABLED)
000015	CICS_JOBSUB_SPOOL	IBMCICS	ACTIVE(ENABLED)
000016	CICS_JOBSUB_TDQINTRDR	IBMCICS	ACTIVE(ENABLED)
000017	CNZ_AMRF_EVENTUAL_ACTION_MSGS	IBMCNZ	ACTIVE(ENABLED)



SDSF: snapshot

Scenario: Find all the inactive z/OS migration health checks.

■ 3) Save results

```

SDSF EDIT      *SNAP                                     Columns 00001 00072
Command ==> create 'mwalle.sdsfsnap.inact.mighc'       Scroll ==> HALF
*****
***** Top of Data *****
- - - - - 216 Line(s) not Displayed
cc0217 ZOSMIGREC_ROOT_FS_SIZE          IBMUSS          INACTIVE(ENABLED)
00cc18 ZOSMIGREC_SUP_TIMER_INUSE       IBMTIMER        INACTIVE(ENABLED)
- - - - - 2 Line(s) not Displayed
*****
***** Bottom of Data *****

```



SDSF: System Command Extension pop-up

What: Lots more capability with the saved systems commands.

More commands: default is now 50 (from 20). Can be 2,000 if you use a PDSE for your ISFTABL.

Grouping: classify your useful commands together

Viewing and filtering: See them and sort them easier

In your words: your own comments on what a command does

How to use: / to use the system command option, then it's intuitive.

Considerations: To see if you are using ISFTABL, check on / for:

STORELIMIT : means that you don't have an ISFTABL

SDSF - System Command Extension

===> _____

STORELIMIT

Comment _____

NOPROFILE: means that commands are not stored in the ISPF profile or ISFTABL

"nothing": means that you are using an ISFTABL and are storing into it 😊

Ensure that on the Options pull-down you have **Set Store Command at Exit** to **ON**.

SDSF: System Command Extension pop-up



How to use: Grouping previous or new commands

Edit Options Help

System Command Extension

1. Type (or retrieve) command

```

====> E AXR,BPXWMIGF -QUERY
====>

```

2. Add human comment

```

Comment Show zFS to HFS migrations occuring

```

3. Give a group name, and show that group

```

Group MIG2ZFS Show MIG2ZFS

```

```

=> F AXR,BPXWMIGF -QUERY
=> F AXR,BPXWMIGF -cancel mwalle.busy.testfs
=> f axr,BPXWMIGF -source mwalle.busy.testfs -target mwalle.
=> f axr,BPXWMIGF -source omvs.hfs -swap
=> f axr,BPXWMIGF -source omvs.hfs -target omvs.zfs -s
=>
=>
=>

```

4. Either <enter> to issue and save command, or F10 to only save in that group

F5=FullScr F6=Details F7=Up F8=Down F10=Save F11=Clear F12=Cancel

PF11 will clear that command from the list.

Non-intuitive: PF11 hit *outside popup* to clear all commands and group.

SDSF: System Command Extension pop-up



How to use: Show all groups, to find a saved command (F4 near **Show**)

```

Edit  Options  Help
-----
Group Select          Row 1 to 12 of 12
Command ==> _____
Selection: 8_
1.
2.  APF
3.  DEVSERV
4.  DISPLAY
5.  EXIT
6.  GENERAL
7.  HEALTHCHECK
8.  LNKLST
9.  LPA
10. MIG2ZFS
11. PDSE
12. SYSPLEX
*****

Edit  Options  Help
-----
System Command Extension
==> _____
==> _____
Comment _____
Group  LNKLST          Show LNKLST (F4 for list)
More:
=> SETPROG LNKLST ADD NAME(      SSNAME(dsn) ATBOTTOM
=> SETPROG LNKLST ACTIVATE      (myglname)
=> DISPLAY PROG, LNKLST
=> SETPROG LNKLST DEFINE NAME(myname) COPYFROM(CURRENT)
=>
=>
=>
=>
F5=FullScr F6=Details F7=Up F8=Down F10=Save F11=Clear F12=Cancel
  
```

PF6 has many of the command details (comments, ...)

Edit -> Clear from the command list will also clear all commands and group.



- **What:** LPA statement in PROGxx (and SETPROG LPA and via CSVDYLPA) lets you identify the containing data set with volser
- **How to use:**
 - LPA ADD MOD (xxx) DSNAME (ddd) **VOLUME (vvv)**
 - SETPROG LPA, ADD, MOD=xxx, DSNAME=ddd, **VOLUME=vvv**
- **Considerations:** This appears STILL not to be documented. Sigh. Will try again to get that fixed.

Dynamic APF: SMF Record



- **What:** SMF record type 90 subtype 37 upon post-IPL APF update (ADD or DELETE)
- **How to use:**
 - PROGxx: APF ADD ... or APF DELETE ...
 - SETPROG APF,ADD,... or SETPROG APF,DELETE,...
 - SMFPRMxx: indicate to collect type 90 subtype 37 record
- **Information in the SMF record:**
 - Function: Add, Delete, DynFormat, StatFormat
 - Was the update via SETPROG, SET PROG, CSVAPF
 - Parmlib member suffix for the SET PROG case
 - Data set name
 - Volser
 - Time of update (STCK)
 - Jobname
 - Command Scheduling Control Block (CSCB)'s CHKEY field
 - Console ID of issuer (-1 for CSVAPF)
 - Utoken of issuer

Dynamic APF: SMF Record (cont)



■ z/OS V2R3 improvements:

- The RACF UTOKEN is stored in its “unencrypted format”
- The UserID within the UTOKEN is at offset x'98' in the data
- The console name is provided at offset x'A8'
- PROGxx supports **APFSMFALL**
 - When specified, the SMF record includes information about updates that are “already in the correct state”. Defaults to initial behavior of not placing “no change” cases in the SMF records
 - The record identifies this situation by a bit:
SMF90T37_AlreadyAsNeeded – the x'01' bit in byte SMF90T37Flags (offset 1)

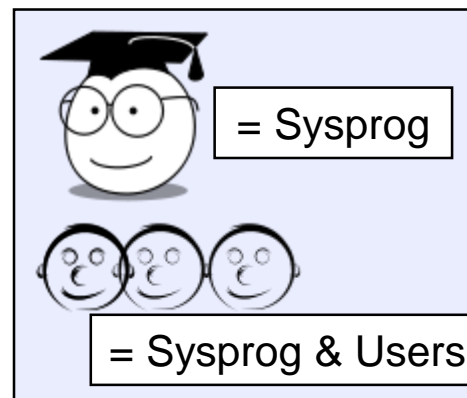
z/OS V2R1 Small Enhancements



❖ **DFSMSdfp and ICKDSF:** Protection for initializing non-empty volumes



❖ **DFSMSdss:** Renaming of VSAM physical data sets





z/OS V2.1 with OA51084 and PI67283 (Sept 2016) and higher: **DFSMSdfp and ICKDSF: Protection for initializing non-empty volumes**

■ What:

- ICKDSF had keyword `NODSEXIST` on the INIT command. This as a default was desired.
 - Means: if there are data sets on the volume (besides the index data set and VVDS), then you will not be allowed to initialize the volume.
- Now, DEVMAN support provides a system-wide value to be used for the ICKDSF default, via `DEVSUPxx`'s `ICKDSF_NODSEXIT=YES` or `NO`.
 - `YES` enables `NODSEXIST` to be defaulted for ICKDSF INIT.
 - Means: if device contains data sets, INIT is terminated. To override you have to now specify an ICKDSF `DSEXIST` keyword on the INIT.
 - `NO` disables `NODSEXIST` for ICKDSF INIT.
 - Means: if device contains data sets, INIT is not terminated.

■ Considerations:

- The `NODSEXIST` parameter will not be defaulted if an online INIT is attempted on a volume that has been initialized as a Data Facility Storage Management Subsystem (DFSMS) managed volume. If data sets other than the VTOC index data set or VVDS exist on a DFSMS managed volume, the command will be terminated.
- If `ICKDSF_NODSEXIST=NO` or `YES` is specified, an IEA253I message is logged at IPL or after a `SET DEVSUP=xx` command is issued. There is no `F DEVMAN` command for this setting.



z/OS V2.1 with OA51084 and PI67283 (Sept 2016) and higher:

DFSMSdfp and ICKDSF: Protection for initializing non-empty volumes

Use case #1:

I want to initialize a volume and want system-wide default protection in case someone has put data sets on there that really shouldn't be lost. I don't want to have to go to a point-in-time backup which might be out of date.

1. Edit my DEVSUPxx to add ICKDSF_NODSEXIST=YES

```
EDIT          SYS1.PARMLIB.POK(DEVSUPMW) -
Command ==> _
***** ***** To
000001 ICKDSF_NODSEXIST=YES
***** ***** Bot
```

1. SET DEVSUP=xx

```
SY1 set devsup=mw
SY1 IEA253I DEVSUP  ENABLED ICKDSF NODSEXIST PARAMETER DEFAULT
SY1 IEA253I DEVSUP  ISO/ANSI TAPE LABEL VERSION DEFAULT IS V3
SY1 IEA253I DEVSUP  TAPE OUTPUT DEFAULT BLOCK SIZE LIMIT IS 32760
SY1 IEA253I DEVSUP  COPYSDDB DEFAULT IS INPUT
SY1 IEA253I DEVSUP  STORAGE LIMIT FOR TAPE DDR SWAP DEFAULTED TO 1000M
SY1 IEA253I DEVSUP  PERFORM NORMAL EXPIRATION DATE PROCESSING
SY1 IEA253I DEVSUP  MULTIPLE INCREMENTAL FLASHCOPY: CHANGE RECORDING V2
SY1 IEE536I DEVSUP  VALUE MW NOW IN EFFECT
```




z/OS V2.1 with OA51084 and PI67283 (Sept 2016) and higher:

DFSMSdfp and ICKDSF: Protection for initializing non-empty volumes

3. Run ICKDSF to initialize a non-empty volume: **failure = success!**

```
//ICKDSFC EXEC PGM=ICKDSF,REGION=0K
//VOLDD DD UNIT=3390,VOL=SER=C96F14,DISP=SHR
//SYSABEND DD DUMMY
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
INIT DDNAME(VOLDD) VERIFY(C96F14) VTOC(0,1,29)
```

```
ICKDSF - MVS/ESA      DEVICE SUPPORT FACILITIES 17.0                TIME: 18:42

INIT DDNAME(VOLDD) VERIFY(C96F14) VTOC(0,1,29)                      00060
ICK00700I DEVICE INFORMATION FOR 0983 IS CURRENTLY AS FOLLOWS:
    PHYSICAL DEVICE = 3390
    STORAGE CONTROLLER = 2107
    STORAGE CONTROL DESCRIPTOR = E8
    DEVICE DESCRIPTOR = 0A
    ADDITIONAL DEVICE INFORMATION = 4A00003C
    TRKS/CYL = 15, # PRIMARY CYLS = 3339
ICK04000I DEVICE IS IN SIMPLEX STATE
ICK03091I EXISTING VOLUME SERIAL READ = C96F14
ICK03096I EXISTING VTOC IS LOCATED AT CCHH=X'0001 0000' AND IS      30 TRACKS.
ICK32179I DATA SETS EXIST ON VOLUME
ICK30003I FUNCTION TERMINATED. CONDITION CODE IS 12
    18:42:47      03/05/18

ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 12
```



z/OS V2.1 with OA51084 and PI67283 (Sept 2016) and higher:

DFSMSdfp and ICKDSF: Protection for initializing non-empty volumes

Use case #2:

I want to (system-wide setting) initialize a volume and I don't care what might be on the volume. Anything there is fine to delete.

1. Edit my DEVSUPxx to add ICKDSF NODSEXIST=NO

```
EDIT          SYS1.PARMLIB.POK(DEVSUPMW)
Command ==> _
*****
000001 ICKDSF_NODSEXIST=NO
***** B
```

1. SET DEVSUP=xx

```
SY1  set devsup=mw
SY1  IEA253I  DEVSUP  DISABLED ICKDSF NODSEXIST PARAMETER DEFAULT
SY1  IEA253I  DEVSUP  ISO/ANSI TAPE LABEL VERSION DEFAULT IS V3
SY1  IEA253I  DEVSUP  TAPE OUTPUT DEFAULT BLOCK SIZE LIMIT IS 32760
SY1  IEA253I  DEVSUP  COPYSDS DEFAULT IS INPUT
SY1  IEA253I  DEVSUP  STORAGE LIMIT FOR TAPE DDR SWAP DEFAULTED TO 1000M
SY1  IEA253I  DEVSUP  PERFORM NORMAL EXPIRATION DATE PROCESSING
SY1  IEA253I  DEVSUP  MULTIPLE INCREMENTAL FLASHCOPY: CHANGE RECORDING V2
SY1  IEE536I  DEVSUP  VALUE MW NOW IN EFFECT
```



z/OS V2.1 with OA51084 and PI67283 (Sept 2016) and higher:
DFSMSdfp and ICKDSF: Protection for initializing non-empty volumes

Use case:

3.Run ICKDSF to initialize a non-empty volume: **success = success!**

```
//ICKDSFC EXEC PGM=ICKDSF,REGION=0K
//VOLDD DD UNIT=3390,VOL=SER=C96F14,DISP=SHR
//SYSABEND DD DUMMY
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
INIT DDNAME(VOLDD) VERIFY(C96F14) VTOC(0,1,29)
```

```
SY1 $HASP373 IBMUSERN STARTED - INIT 1 - CLASS A - SYS SY1
SY1 ICK061I 0983 VTOC INDEX CREATION SUCCESSFUL: VOLUME IS IN INDEX
FORMAT
SY1 $HASP395 IBMUSERN ENDED
```

```
ICK10705I VOLUME SERIAL NUMBER FOR DEVICE 0983 IS C96F14
VTOC LOCATION MOVED FROM CCHH=X'0001 0000' TO X'0000 0001'
ICK00001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
18:54:26 03/05/18
ICKDSF - MVS/ESA DEVICE SUPPORT FACILITIES 17.0 TIME:
ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```



z/OS V2.1:

DFSMSdss: Renaming of VSAM physical data sets

- **What:** Ability to rename a VSAM physical data set on a **COPY** or **RESTORE**.
 - Use **RENAMEUNCONDITIONAL** keyword for these operations in this case (not **RENAME**)
 - Prior to z/OS V2.1, a rename could be done only on non-VSAM physical data sets.
 - Also, as of z/OS V2.1, **RESTORE** supports **REPLACEU**, just as **COPY** did before.
- **How to use:**
 - When **PHYSINDYNAM** or is **PHYSINDD** used, you can now use **RENAMEU** for renaming a VSAM data set.
 - **REPLACEUNCONDITIONAL** keyword on the **COPY** or **RESTORE** command now works for physical VSAM data sets are *not* cataloged during physical processing within SMS or non-SMS environments.
 - The **CATALOG** keyword is ignored for VSAM data sets during physical restore. Use **IDCAMS DEFINE RECATALOG** to catalog the data sets after the physical restore.



z/OS V2.1:

DFSMSdss: Renaming of VSAM physical data sets

Use case on z/OS R13:

I want to copy (overlay) a VSAM data set physically and rename it, on the same system.

```
//CPYCSI EXEC PGM=ADRDSSU,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  COPY DATASET( INC( PROD.ZOS113.** ) ) -
    PHYSINDYNAM ((PAGE08)) OUTDYNAM ((C96F1B)) -
    RENAMEU( (PROD.ZOS113.** , CLONE.ZOS113.** ) ) -
    ALLDATA(*)
/*
```

ADR332E (001)-PCVSM(01), CLUSTER PROD.ZOS113.CSI IN CATALOG
PAGE08.CATALOG NOT PROCESSED. PHYSICAL DATA SET OPERATION
DOES NOT SUPPORT RENAME OF VSAM DATA SETS



z/OS V2.1:

DFSMSdss: Renaming of VSAM physical data sets

Same use case on z/OS V2.1:

```
//CPYCSI EXEC PGM=ADRDSSU,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
    COPY DATASET(INC(PROD.ZOS113.**)) -
        PHYSINDYNAM ((PAGE08)) OUTDYNAM ((C96F1B)) -
        RENAMEU((PROD.ZOS113.**,CLONE.ZOS113.**)) -
        ALLDATA(*)
/*
```

ADR395I (001)-PCVSM(01), DATA SET PROD.ZOS113.CSI.DATA ALLOCATED WITH
NEWNAME CLONE.ZOS113.CSI.DATA, ON VOLUME(S): C96F1B

ADR395I (001)-PCVSM(02), DATA SET PROD.ZOS113.CSI.INDEX ALLOCATED WITH
NEWNAME CLONE.ZOS113.CSI.INDEX, ON VOLUME(S): C96F1B

ADR418I (001)-PCVSX(01), **THE FOLLOWING COMPONENTS FOR CLUSTER CLONE.ZOS113.CSI
ON C96F1B MAY HAVE TO BE CATALOGED IN CATALOG PAGE08.CATALOG**

COMPONENT CLONE.ZOS113.CSI.DATA

COMPONENT CLONE.ZOS113.CSI.INDEX

ADR454I (001)-DDDS (01), THE FOLLOWING DATA SETS WERE **SUCCESSFULLY PROCESSED**

CLUSTER NAME PROD.ZOS113.CSI

COMPONENT NAME PROD.ZOS113.CSI.DATA

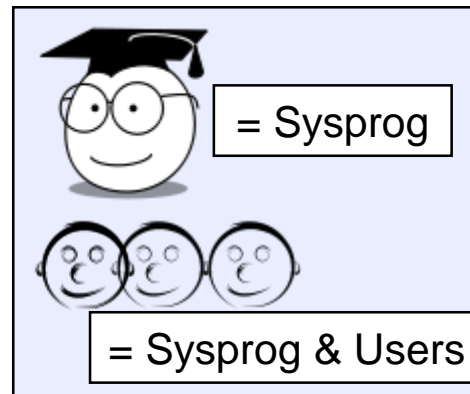
COMPONENT NAME PROD.ZOS113.CSI.INDEX

Older than the hills

Small Enhancements



❖ SMP/E: Automatic cross-zone requisite checking



Older than the hills:

SMP/E: Automatic cross-zone requisite checking

What: Sometimes a PTF or FMID might contain a ++IF REQ for a PTF in another zone. (BCP -> CICS, CICS -> DB2, ...). How do you easily know that that required PTF is installed in other zone? Do you always remember to REPORT CROSSZONE? You can let SMP/E do that verification for APPLY, ACCEPT, RESTORE processing automatically.

Consideration: Remember, you need to set this up in your GLOBAL zone, so it might be that you are not doing this after a ServerPac install when you get a new GLOBAL.

ServerPac's SMPREP does a "one time" REPORT CROSSZONE, though.

SMP/E commands will fail until the requisite is satisfied (in that other zone) 😊, or you BYPASS(XZIFREQ) ☹.

Older than the hills:

SMP/E: Automatic cross-zone requisite checking

How to use (there are variations of this):

Tell SMP/E which zones to use in a default zone group.

```
SET BDY (GLOBAL) .
```

```
UCLIN. /* assuming z/OS zones already defined in this GLOBAL */
```

```
  ADD GLOBALZONE ZONEINDEX (
    (cicstgt,cics.target.csi,TARGET)
    (db2tgt,db2.target.csi,TARGET)
    (cicsdlb,cics.dlib.csi,DLIB)
    (db2dlib,db2.dlib.csi,DLIB) ) .
```

```
ENDUCL.
```

```
ADD ZONESET (XZONE)
```

```
  ZONE (zostgt, zosdlb, cicstgt, db2tgt, cicsdlb, db2dlib)
```

```
  XZREQCHK (YES) .
```

```
ENDUCL.
```

XZREQCHK(YES) means the zones defined in the ZONESET “XZONE” are used as the default zone group any time an APPLY, ACCEPT, or RESTORE command is done in any of those zones .

Older than the hills:

SMP/E: Automatic cross-zone requisite checking

No ZONESETs in my CSI (1.1, GLOBAL, 7 ZONESET):

```

                                DEFINITION - ADD NEW ZONESET ENTRY
===>

No ZONESETS are currently defined for
GLOBAL      ZONE  GLOBAL.
Enter the name of the ZONESET to be created:

ZONESET NAME  ===> _
  
```

Run JCL to ADD ZONESET. Now, I have an automatic cross zone requisite set of zones!

```

                                DEFINITION - ZONESET
===> _

Verify or edit the ZONESET list. (Select ONE at a time.
Only the first of multiple selections will be processed.)

    A - on the command line to add an entry
    D - next to entry to delete an entry
    S - next to entry to update an entry
    T - Toggle XZREQCHK value

When the list is complete, enter END .

OPTION  NAME      XZREQCHK
      XZONE      YES
***** Bottom of data *****
  
```

Summary of What We Might Want to Share:

- **System Programmer & User Items:**

- z/OS UNIX (V2.2): zlsf and jsonprint
- SDSF (V2.2): snapshot
- SDFS (V2.2): System Command Extension pop-up
- DFSMSdss (V2.1): Renaming of VSAM physical data sets



- **System Programmers' Items:**

- RACF (V2.3): IRRPRMxx
- z/OS UNIX (V2.3): BPXWMIGF facility
- PROGxx (V2.2): LPA Volser
- Dynamic APF (V2.2): SMF 90-37
- DFSMSdfp and ICKDSF (V2.1): Protection for initializing non-empty volumes
- SMP/E: Automatic cross-zone requisite checking



z/OS Summary Enhancements – Best of 2018!

The year 2018 is displayed in a large, stylized font where each digit is composed of many small, colorful, geometric shapes (triangles, squares, etc.) in shades of green, blue, yellow, and red.

- **z/OS V2.3:**
 - ✓ **RACF:** IRRPRMxx Parmlib that specifies dsn and range tables.
 - ✓ **z/OS UNIX:** BPXWMIGF facility Your solution if appl avail is critical.
- **z/OS V2.2:**
 - ✓ **z/OS UNIX:** updates to zlsf, and adding jsonprint see more and use with programs
 - ✓ **SDSF:** snapshot on tabular panels, find information fast
 - ✓ **SDSF:** System Command Extension pop-up Set up your fav commands in groups
 - ✓ **PROGxx:** LPA Volser lets you identify the containing data set with volser!
 - ✓ **Dynamic APF:** SMF 90-37 upon post-IPL APF update
- **z/OS V2.1:**
 - ✓ **DFSMSdfp and ICKDSF:** Protection for initializing non-empty volumes Helpful.
 - ✓ **DFSMSdss:** Renaming of VSAM physical data sets Great rename ability now.
- **Older than the hills:**
 - ✓ **SMP/E:** Automatic cross-zone requisite checking Don't forget to set it up with your new Global!

z/OS Small Enhancements - A history

SlideShare | small

Home Explore Presentation Courses PowerPoint Courses by LinkedIn Learning

My Uploads My Comments Analytics

Most Recent Select all

small z/OS technical enhancements you might have missed: Edition 2018A
Marna WALLE, mwalle@us.ibm.com
Member of the IBM Academy of Technology
z/OS System Installation
IBM z Systems, Poughkeepsie, NY, USA
2018 IBM Systems Technical University
May 2017

z/OS small enhancements, episode 2...
2 minutes ago, 58 slides

2 0 0 0

Edit Preview

2016 z/OS enhancements over the past several releases
Marna WALLE, mwalle@us.ibm.com
Member of the IBM Academy of Technology
z/OS System Installation
IBM z Systems, Poughkeepsie NY, USA

2016 IBM Systems Technical Events
ibm.com/training/systems

Small enhancements - Edition 2016B
1 year ago, 66 slides

209 0 0 4

Edit Preview

small z/OS enhancements over the past several releases
Marna WALLE, mwalle@us.ibm.com
Member of the IBM Academy of Technology
z/OS System Installation
IBM z Systems, Poughkeepsie NY, USA

2016 IBM Systems Technical Events
ibm.com/training/systems

z/OS Small Enhancements - Episode...
2 years ago, 55 slides

619 1 0 18

Edit Preview

Small Stuff You May Want To Use in z/OS
Episode 2018A

Small Past IBM z/OS Releases -

You've Got It Good: Small z/OS

Eight older "Enhancements" presentations are on LinkedIn's SlideShare, going back to 2013 and covering other little gems.