

Bit Bucket X'27'

Ed Jaffe, edjaffe@phoenixsoftware.com
Sam Knutson, SKnutson@geico.com
Skip Robinson, robinsjo@sce.com



SHARE 115
Boston, MA
5 August 2010





VSAM? ICF? BCS? VVR? VVDS? WTF?

(Ed Jaffe)

VVDSFIX

- Anyone who maintains or upgrades z/OS systems has probably run into difficulties involving unsynchronized catalog/volume elements for VSAM data sets
 - Some ICF design issues and restrictions are “baggage” left over from the old VSAM catalog structure.
 - VTOC and VVDS are on the volume. BCS is in the catalog which can reside on another volume.
- For example, suppose you need to access a VSAM data set on a volume whose BCS catalog is no longer available.
- You can't just re-catalog into another BCS. *z/OS Access Method Services for Catalogs* says you must build a new BCS with the original name, then DEFINE RECATALOG the VSAM data set. This might not always be possible.
- Some ISVs have tools that provide relief for this and other ICF catalog restrictions.

VVDSFIX

- IBM provides an unsupported tool that intelligently reads, checks and modifies a VVDS to help get around some restrictions imposed by ICF design and other factors.
- You can read about and download this tool from this URL: <http://www.ibm.com/support/docview.wss?uid=isg3S1000618>
- You download a RECFM=FB LRECL=1024 sequential data set called VVDSFIX.VER14.TRSD which you must decompile, using AMATERSE (or TRSMMAIN), to a sequential data set with RECFM=FB and LRECL=80.
- The result is JCL that links the utility into SYS1.LINKLIB and executes it.
- The utility need not reside in SYS1.LINKLIB. You can use any APF-authorized library.

VVDSFIX

```
//VVDSFIX JOB (LG),VVDS.FIXER,MSGLEVEL=(1,1),MSGCLASS=H,
//      CLASS=A,TIME=1440,REGION=900K
//LKEDX0 EXEC PGM=HEWL,
//      PARM='NCAL,LIST,XREF,LET,SIZE=(950K,250K)'
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSLMOD DD DISP=SHR,DSN=SYS1.LINKLIB
//* *****
//*
//*          VVDSFIX JUN 22, 2009          Version 14          **
//*
//* VVDSFIX - A tool used to read, check and modify VVDSs.          **
//* This program must run authorized and expects the                **
//* following DD statement:                                          **
//*
//*          VVDS DD DSN=SYS1.VVDS.Vnnnnnn,DISP=SHR                **
//*
//* where nnnnnn is the VOLSER of the VVDS to be used.             **
//* GDGFIX and DELBCSR also require a DD card for a                **
//* Catalog. Example DD follows:                                     **
//*
//*          CATDD DD DSN=catalog name,DISP=SHR                    **
//*
//* Do not link VVDSFIX with the RENT attribute. (This             **
//* will cause an ABEND0C4 at offset x'26' in VVDSFIX.)           **
//*
...
      SETCODE AC(1)
      NAME VVDSFIX(R)
//VVDSGO EXEC PGM=VVDSFIX,COND=(0,NE)
//STEPLIB DD DSN=SYS1.LINKLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//*CATDD DD DSN=JACKCAT,DISP=SHR
//VVDS DD DSN=SYS1.VVDS.VXIDBB0,DISP=SHR,VOL=SER=XIDBB0,
//      UNIT=3390
/*
```

Note: the comment refers to **GDGFIX** which is an invalid command.

VVDSFIX

- The utility accepts either *SYSIN* or *WTOR* response.
 - *WTOR* response allows prompting and confirmation for certain actions. The *SYSIN* equivalent is to *fail* if a prompt is required.
- *BWOFIX*—zero the 3 fuzzy bits in a *VVR*.
- *CATNAME*—change or add *CATALOG* name in the *VVRs*.
- *CHECK*—check the *VVDS* for correctness.
- *CHECKX*—like *CHECK* but corrects certain errors (e.g., out of synch space maps).
- *DEFERD*—change all non-*VSAM* non-*GDGs* *NVRs* from deferred to non deferred status. It also updates the *BCS* record.
- *DELBCSR*—deletes a *BCS* record when special characters in the name prevent *IDCAMS DELETE* from working.

VVDSFIX

- DELVVR—delete a VVR.
- TIMESTAMP—add TIMESTAMP cells to VVRs.
- ZAPVVR—ZAP bytes in a VVR
- I set mine up with a started task JCL procedure.
- You'll probably want to run as a batch job for operations performed on a regular basis. (In that case, "canned" JCL could mean fewer typos.) Also, note that the DEFERD, DELBCSR and (invalid) GDGFIX commands will require a CATDD DD statement that points to the catalog.

```
//VVDSFIX  PROC V=,U='3390',SC='*'
//VVDSFIX  EXEC PGM=VVDSFIX
//SYSPRINT DD  SYSOUT=&SC
//VVDS     DD  DSN=SYS1.VVDS.V&V,DISP=SHR,VOL=SER=&V,UNIT=&U
```

VVDSFIX

- Utility output is via MLWTO with ROUTCDE=11.
- If you have a SPOOL browse interface that can watch data sets grow line-by-line, you can monitor the job/STC's JESMSG LG and JESYSMSG output.
- If you're on an MCS/EMCS/SMCS console, you'll need to enable to the console to receive route code 11 messages.
- To temporarily enable a console to receive route code 11 messages, issue D C,* to see the original ROUT= settings, issue V CN(*),AROUT=(11) to add route code 11 to the existing values, and restore the original settings when finished via V CN(*),DROUT=(11).
- If you're using a hardcopy browser (e.g., for OPERLOG) you will use V OPERLOG,HARDCPY,AROUT= and V OPERLOG,HARDCPY,DROUT= to set this.

VVDSFIX

```
S VVDSFIX,V=ZBSYS1
IAT6100 ( DEMSEL ) JOB VVDSFIX (JOB94932), PRTY=15, ID=SYSOPER
IEF695I START VVDSFIX WITH JOBNAME VVDSFIX IS ASSIGNED TO USER SYSOPER , GROUP SYS1
IEF403I VVDSFIX - STARTED - TIME=14.46.58
*0974 Enter command for VVDS fixer
974,CHECK
** VVDS SCAN AND CHECK COMPLETE ** 156
*****
*****
**** 0168 VVDS CI checked on VOLSER=ZBSYS1 ****
**** 0160 empty CIs in VVDS ****
**** 0000 VVRs with the IRF bit on ****
**** 0000 MAPs or VVRs incorrect ****
**** 0000 Problems found (see msg above) ****
**** 001E VVRs in the VVDS ****
*****
*****
*0975 Enter command for VVDS fixer
975,CATNAME
*0976 Enter old CATALOG name to be replaced.
976,CATALOG.MCATA.PHXHQ
*0977 Enter new catalog name to be put into VVRs.
977,CATALOG.MCATB.PHXHQ
0165 CI CHECKED, 001E VVRs FOUND, 0012 VVRs CHANGED
*0978 Enter command for VVDS fixer
978,END
IEF404I VVDSFIX - ENDED - TIME=15.49.47
```

Note: the count in red tells us these values are displayed in hex and not in decimal. Keep that in mind...



- The doc says you can specify the prompt answers separated by semi-colons i.e., 975,CATNAME;oldcatname;newcatname; up to 155 chars.
- I could not make that work in my environment because I have CMDDELIM(:) specified on my INIT statement in CONSOLxx.



Let's Get Wild
(Ed Jaffe)

Volume Name Mask on ISPF Data Set List Utility

- I must have been living in a cave somewhere as I only just recently learned that ISPF's Data Set List Utility (option 3.4) supports wildcard characters in volume names, e.g.:
 - * for all volumes
 - PRM* for all volumes beginning with PRM
 - M%C* for all volumes beginning with M followed by any single character, a C, and any other three characters.
- The resulting display is initially sorted by data set name. If you prefer to sort by volume, you can issue the SORT VOLUME command. To restore the original sort order, issue the SORT command with no operands.
- Remember, this is a VTOC search using CVAF. If your volume name mask includes many volumes, it can take a relatively long time for the request to be honored.

Volume Name Mask on ISPF Data Set List Utility

Menu RefList RefMode Utilities Help

Data Set List Utility

Option ==> _____

More: +

blank	Display data set list	P	Print data set list
V	Display VTOC information	PV	Print VTOC information

Enter one or both of the parameters below:

Dsname Level . . . ** . TRSD

Volume serial . . . MVS*

Data set list options

Initial View

- 1 1. Volume
- 2. Space
- 3. Attrib
- 4. Total

Enter "/" to select option

- / Confirm Data Set Delete
- / Confirm Member Delete
- / Include Additional Qualifiers
- / Display Catalog Name
- Display Total Tracks
- Prefix Dsname Level

When the data set list is displayed, enter either:

"/" on the data set list command field for the command prompt pop-up,
*DSLIS

Volume Name Mask on ISPF Data Set List Utility

Menu Options View Utilities Compilers Help

DSLISL - Data Sets on volume MVS*

Row 1 of 8

Command ==>

Scroll ==> CSR

Command - Enter "/" to select action

Message

Volume

EDJXADM.EJES4700.TRSD

MVSNV1

EDJXADM.EJES4800.TRSD

MVSNV1

EDJXADM.VVDSFIX.VER14.TRSD

MVSNV1

EJES.PRODGEN.PACKAGE.TRSD

MVSNV1

GILBERT.EJES4700.TRSD

MVSNV0

GILBERT.EJWS4700.TRSD

MVSNV1

JIMP2.PMR.X36973.Y227.Z000.TRACE.TRSD

MVSNV0

PSCOTT.EJES4700.TRSD

MVSNV1

***** End of Data Set list *****

*DSLISL



Use the FORCE, Luke!
(Ed Jaffe)

MODIFY, STOP, CANCEL, FORCE ARM, FORCE?

- Many of you have probably wondered about all of the different ways to take down an address space.
 - You can issue a subsystem command with something like "TERM".
 - You can respond to a WTOR with something like "TERM".
 - You can issue the MODIFY command with something like "TERM".
 - You can issue the STOP command.
 - You can issue the CANCEL command.
 - You can issue the FORCE ARM command.
 - You can issue the FORCE command.
- Unfortunately, the answer to which commands are appropriate for any given address space depends upon what the programmer had in mind when the code was designed.
- And what about the ARM keyword? Does that have something to do with MVS Automatic Restart Manager?

ARM Has Nothing To Do With ARM!

- The ARM keyword on the FORCE commands has absolutely nothing to do with the MVS Automatic Restart Manager or ARM. (Talk about overloading acronyms!)
- The ARMRESTART keyword, which can be specified on the CANCEL and FORCE commands (as well as JES CANCEL-type commands) says to restart the address space if it is registered with the MVS Automatic Restart Manager. If you don't specify ARMRESTART, the address space will not be restarted after CANCEL or FORCE.
- What's really confusing here is that both the ARM and ARMRESTART keywords can be specified together i.e., FORCE myspace,ARM,ARMRESTART
- For this discussion, forget about ARM (the MVS Automatic Restart Manager) and focus on ARM!

Terminate By Issuing a Subsystem Command

- Programs defined as a subsystem in IEFSSNxx can receive commands via subsystem command routing.
- Such programs might implement some sort of shutdown request i.e., subsystemprefixTERM
- This would constitute a normal shutdown of the address space. The program would have to be operating normally—at least well enough to receive and process the subsystem command.
- An example of such a program is IBM's RACF.
 - We set the subsystem command prefix to "RACF" via the INITPARM parameter in IEFSSNxx passed to IRRSSI00.

RACF STOP

```
IRRB069I (RACF) RACF SUBSYSTEM STARTING SHUTDOWN PROCESSING.
IRRB005I (RACF) RACF SUBSYSTEM TERMINATION IS COMPLETE.
$HASP250 RACF PURGED -- (JOB KEY WAS C64A6C52)
IEF404I RACF - ENDED - TIME=17.44.52
IEF196I IEF142I RACF RACF - STEP WAS EXECUTED - COND CODE 0000
IEF352I ADDRESS SPACE UNAVAILABLE
IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=0031.
```

Terminate By Responding to WTOR

- Some programs hang a WTOR as a way to get commands from the operator.
- Such programs might implement some sort of shutdown request i.e., R 1234,TERM
- This would constitute a normal shutdown of the address space. The program would have to be operating normally—at least well enough to wake up when the WTOR ECB is posted and process the shutdown request.
- An example of such a program is IBM's NVAS.

```
*0994 EMS0990A NVAS      READY FOR COMMANDS.  
994,TERMI  
IEE600I REPLY TO 0994 IS;TERMI  
EMS0999E NV/AS HAS BEEN TERMINATED - RETURN CODE = 0.  
IEF404I NVAS - ENDED - TIME=17.31.15  
IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=009D.  
IAT7450 JOB NVAS      (JOB94991) PURGED
```

Terminate By Issuing the MODIFY Command

- Some programs use the EXTRACT/QEDIT interface as a way to get commands and operands from the operator.
 - START, STOP, MODIFY.
- Such programs might implement some sort of shutdown request i.e., MODIFY program, TERM
- This would constitute a normal shutdown of the address space. The program would have to be operating normally—at least well enough to wake up when the COM ECB is posted and process the MODIFY CIB.
- An example of such a program is OMVS.

F OMVS, SHUTDOWN

BPXI055I OMVS SHUTDOWN REQUEST ACCEPTED

GFSC701I z/OS NETWORK FILE SYSTEM CLIENT SHUTDOWN BEGINS.

*BPXF229I PHYSICAL FILE SYSTEM NFS IS NOW RECYCLING

GFSC500I CLIENT LOG DATA SET, NFS.CLIENT.LOG1, FLUSHED.

GFSC702I z/OS NETWORK FILE SYSTEM CLIENT SHUTDOWN COMPLETE.

IEF404I NFSC - ENDED - TIME=19.37.33

...

BPXN002I UNIX SYSTEM SERVICES PARTITION CLEANUP COMPLETE 853

FOR SYSTEM MVSA0

IEA989I SLIP TRAP ID=X13E MATCHED. JOBNAME=OMVS , ASID=000F.

*BPXI056E OMVS SHUTDOWN REQUEST HAS COMPLETED SUCCESSFULLY

Terminate By Issuing the STOP Command

- Some programs use the EXTRACT/QEDIT interface as a way to get commands and operands from the operator.
 - START, STOP, MODIFY (optional).
- Such programs might respond to a STOP command.
- This would constitute a normal shutdown of the address space. The program would have to be operating normally—at least well enough to wake up when the COM ECB is posted and process the STOP CIB.
- An example of such a program is VLF.

P VLF

```
IEA989I SLIP TRAP ID=X13E MATCHED. JOBNAME=VLF , ASID=001F.  
IEA989I SLIP TRAP ID=X13E MATCHED. JOBNAME=VLF , ASID=001F.  
COF033I VLF HAS TERMINATED BECAUSE OF AN OPERATOR STOP REQUEST.  
IEF404I VLF - ENDED - TIME=11.02.07  
IEA989I SLIP TRAP ID=X33E MATCHED. JOBNAME=*UNAVAIL, ASID=001F.
```

Terminate By Issuing the CANCEL Command

- CANCEL is often used to terminate programs that are misbehaving. But, it is also be used to terminate programs that have no other operator communication vehicle i.e., subsystem, WTOR, MODIFY/STOP.
- CANCEL is accepted for programs with no PPT entry or whose PPT entry defaults to or explicitly specifies CANCEL. If the PPT entry contains NOCANCEL, you get IEE838I NON-CANCELABLE - ISSUE FORCE ARM.
- CANCEL causes the job step TCB to be terminated with a x222 or x122 (CANCEL w/DUMP) completion code via CALLRTM TYPE=ABTERM. Any TCBs not taken down and DETACHED abend with x13E but x222 or x122 in SDWA.
- ESTAE/ESTAI recovery routines created w/TERM=YES are given control by RTM. Retry from recovery routines is disallowed (SDWACLUP is ON).
- TCB- and ASCB-level resource managers also get control.

Terminate By Issuing the CANCEL Command

- Programs that expect CANCEL for normal termination don't have to worry about supporting dual-path shutdown.
- An example of such a program is APPC.

C APPC

```
IEE301I APPC                CANCEL COMMAND ACCEPTED
IEA989I SLIP TRAP ID=X222 MATCHED.  JOBNAME=APPC      , ASID=0022.
ATB010I APPC IS TERMINATING DUE TO OPERATOR CANCEL
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=APPC      , ASID=0022.
IST804I CLOSE IN PROGRESS FOR AA0RRSF OPENED BY APPC
IST400I TERMINATION IN PROGRESS FOR APPLID AA0RRSF
IST805I VTAM CLOSE COMPLETE FOR IOASERV
IST805I VTAM CLOSE COMPLETE FOR AA0RRSF
IST804I CLOSE IN PROGRESS FOR AA0EJWS OPENED BY APPC
IST400I TERMINATION IN PROGRESS FOR APPLID AA0EJWS
IST805I VTAM CLOSE COMPLETE FOR AA0EJWS
IST804I CLOSE IN PROGRESS FOR AA0APPCB OPENED BY APPC
IST400I TERMINATION IN PROGRESS FOR APPLID AA0APPCB
IST805I VTAM CLOSE COMPLETE FOR AA0APPCB
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=APPC      , ASID=0022.
...
IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=APPC      , ASID=0022.
IEF450I APPC APPC - ABEND=S222 U0000 REASON=00000000 763
      TIME=11.13.25
IEF196I IEF472I APPC APPC - COMPLETION CODE - SYSTEM=222 USER=0000
IEF196I REASON=00000000
IEF352I ADDRESS SPACE UNAVAILABLE
IEF196I IEF352I ADDRESS SPACE UNAVAILABLE
IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=0022.
ATB002I APPC HAS TERMINATED.
```

Terminate By Issuing the FORCE ARM Command

- FORCE ARM is often used to terminate programs that are misbehaving. But, it is also be used to terminate programs that have no other operator communication vehicle i.e., subsystem, WTOR, MODIFY/STOP.
- FORCE ARM is accepted for programs whose PPT entry explicitly specifies NOCANCEL. Otherwise, you get IEE838I CANCELABLE - ISSUE CANCEL BEFORE FORCE.
- FORCE ARM causes the job step TCB to be terminated with a xA22 completion code via CALLRTM TYPE=ABTERM. Any TCBs not taken down and DETACHED abend with x13E but xA22 in SDWA.
- ESTAE/ESTAI recovery routines created w/TERM=YES are given control by RTM. Retry from recovery routines is disallowed (SDWACLUP is ON).
- TCB- and ASCB-level resource managers also get control.

Terminate By Issuing the FORCE ARM Command

- Programs that expect FORCE ARM for normal termination don't have to worry about supporting dual-path shutdown.
- An example of such a program is AXR (System Rexx).

FORCE AXR,ARM

```
IEE301I AXR                FORCE ARM COMMAND ACCEPTED
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=AXR      , ASID=0018.
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=AXR      , ASID=0018.
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=AXR      , ASID=0018.
IEA631I OPERATOR *AXR01A0 NOW INACTIVE, SYSTEM=MVSA0  , LU=AXREMCS
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=AXR      , ASID=0018.
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=AXR      , ASID=0018.
...
IEA631I OPERATOR *AXR04A0 NOW INACTIVE, SYSTEM=MVSA0  , LU=AXREMCS
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=AXR      , ASID=0018.
IEF450I IEESYSAS AXR - ABEND=SA22 U0000 REASON=00000000 845
        TIME=11.37.52
IEF196I IEF472I IEESYSAS AXR - COMPLETION CODE - SYSTEM=A22 USER=0000
IEF196I REASON=00000000
IEF352I ADDRESS SPACE UNAVAILABLE
IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=0018.
AXR0103I SYSTEM REXX HAS ENDED
```

Terminate By Issuing the FORCE Command

- FORCE is used only to terminate programs that are misbehaving and not responding to repeated CANCEL or FORCE ARM requests.
- FORCE is accepted only after CANCEL or FORCE ARM—as appropriate—has already been tried. Otherwise, you get IEE838I CANCELABLE - ISSUE CANCEL BEFORE FORCE or IEE838I NON-CANCELABLE - ISSUE FORCE ARM.
- FORCE causes the address space to be terminated with a xA22 completion code via CALLRTM TYPE=MEMTERM.
- This form of address space termination bypasses all TCB-level recovery and resource manager routines! It should be used only as a last resort. IPL might be your next step!
- ASCB-level resource managers are still invoked.

Terminate By Issuing the FORCE Command

- If you FORCE a batch job, it might get restarted immediately, depending upon whether the job was successfully journaling and/or what the JES failure restart options are.
- The example below FORCEs a job doing assembly/link. It is in a job class with JOURNAL=NO and RESTART=YES.

```
FORCE EJES21,A=0093
```

```
IEE301I EJES21                FORCE COMMAND ACCEPTED
IAT6808 JOB EJES21      (JOB94037) - RE-ENQUEUED AND HELD BY END OF MEMORY
IAT2006 PREMATURE JOB TERM - JOB EJES21      (JOB94037) - RESTARTED      - MVS60
IAT2001 OS MVS WAITING FOR WORK - MVS60
IEF743I INIT          FORCED - CODE SA22 - IN ADDRESS SPACE 0093
IEF743I EJES21      FORCED - CODE SA22 - IN ADDRESS SPACE 0093
IAT2000 JOB EJES21      (JOB94037) SELECTED MVS60      SRVCLASS=BATCH_LO NET-ID=EJESGENR
ICH70001I EDJX2      LAST ACCESS AT 15:44:38 ON WEDNESDAY, JULY 21, 2010
IEF403I EJES21 - STARTED - TIME=15.55.41
```

The HALT Command

- Incredibly, there is still yet *another* command z/OS supports for shutting things down—HALT.
- To the best of my knowledge, there have only ever been three possible uses of HALT:
 - HALT TP (was for TCAM—no longer supported)
 - HALT NET (still used for VTAM)
 - HALT EOD
- HALT is not a documented programming interface. So, unless IBM or a clever ISV writes a new address space that responds to HALT, or IBM supports shutting down VTAM through its MODIFY interface, we will be stuck with the lone VTAM exception forever.
- Note: I don't consider HALT EOD to be an exception. It is a special command, with a special purpose, that doesn't shut down any address space. VTAM is the "odd ball" here.



Excessive FORCE
(Sam Knutson)

The FORCE Command

- FORCE normally works and removes an address space from the system. It is unwise to assume that its use is even "Mostly" harmless
- FORCE is documented z/OS V1R11.0 MVS System Commands z/OS V1R11.0 SA22-7627-22
- "After issuing FORCE, you might have to re-IPL"
- "If you need a dump, you must issue a DUMP command before you issue FORCE"
- We take a dump if a the first or sometimes first few CANCEL attempts do not succeed before beginning to attempt further to remove the work from the system
- CANCEL may need to be issued more than once. I provide a ROT to our team of 10x pausing at least 10 seconds between each attempt

The FORCE Command

- Practice makes perfect
 - Development and Test systems afford more opportunities to hone skills and procedures for handling these situations
- Consult the experts
 - IBM or your ISV product support teams should be consulted whenever possible before using abnormal procedures to terminate software and always after to attempt root cause resolution
- Some simple things you might check before using FORCE
 - Unexpected WTOR?
 - Latch held (D GRS, ANALYSIS, LATCH or V PDSE, ANALYSIS)?
 - Check for DFSMSHsm recalls in progress may prevent CANCEL from proceeding without any message
 - Review LOGREC/EREP and SYSLOG to see what occurred prior
 - Check you MVS Monitor or RMF III to see what is delaying this job

The FORCE Command

- A DBMS may be protecting your data integrity
- IMS may decline to allow a CANCEL but at least issues a message DFS0802I CANCEL COMMAND REJECTED BY IMS, USE /STOP REGION COMMAND TO STOP IMS DEPENDENT REGION, JOBNAME=zzzzzzzz
- DB2 may be backing out work i.e. DSNR048I -DB21 DSNRBMON UR BACKOUT PROCESSING LOG RECORD AT RBA 0A694D7B6072 TO RBA 0A669A7956A3 FOR CORRELATION NAME = XXD6CXU

Who you gonna call? CALLRTM

- FORCE is a big hammer but sometimes the only reason a address space won't CANCEL is because one task is not terminated and is preventing address space termination
- For years Level-2 has provided a small CALLRTM assembler program as a service aid that customers can assemble and use to terminate a TCB using the CALLRTM macro. See item [RTA000142394](#) for this program.
- ISV tools like TMON and SYSVIEW provide the same facility
- CALLRTM — Call Recovery Termination Manager macro documented in z/OS V1R11.0 MVS Programming Authorized Assembler Services Reference Vol 1 (ALESERV-DYNALLOC) z/OS V1R10.0-V1R11.0 SA22-7609-10
- SHARE requirement SSMVSE09005 Provide new or updated (CANCEL) Operator command to CANCEL a TCB is **Recognized**



Standalone Dump (Skip Robinson)

It's SAD, Oh So SAD

- Standalone dump is seldom needed—thankfully
- If you need it, it had better work right the first time
- The boss doesn't want to hear that there's no doc for the system crash
- Setting up SAD involves two processes
 - 1. Define/initialize the SYS1.SADMP data set (once only)
 - 2a. Create/replace the PAGEDUMP data set (after a PTF)
 - 2b. Write (SAD) IPL text on the PAGEDUMP volume after 2a
- Historically two methods to create/replace PAGEDUMP
 - 'One-step' and 'two-step'
 - 'One-step' runs program AMDSAOSG in a single job step
- OA31077 changed 'one-step' to two actual job steps
- Hence it's now called 'one-stage'

Bad Boy OA31077

- Two problems with OA31077
- -- It's an 'integrity' APAR, so you cannot read it (!)
- -- PTF chains for R10 and R11 went PE
- Result: many people don't know about the change
- PE chain resolved just this week (8/4) with GA PTFs
 - UA55675 for R10
 - UA55676 for R11
- For full doc on new 'one-stage' process, see [OA33162](#)
- 1st step: PGM=AMDSAOSG; 2nd step: PGM=ICKDSF
- Add some temp DSs in both steps, such as TRKOTEXT
 - Formerly allocated dynamically by 'one-step' AMDSAOSG
 - Job fails without all required DDs

Acknowledgements Both Knowing and Unknowing



- Jim Mulder, IBM
- Jerry Ng, IBM
- Martin Packer, IBM
- Peter Relson, IBM
- David Tidy, Dow Benelux B.V.
- Marna Walle, IBM