Introduction to Reading Dumps

Special Notices

Information in this document is correct to the best of our knowledge at the time of this writing. Please send feedback by fax to "AIXServ Information" at (512) 823-4009.

Please use this information with care. IBM will not be responsible for damages of any kind resulting from its use. The use of this information is the sole responsibility of the customer and depends on the customer's ability to evaluate and integrate this information into the customer's operational environment.

Note: The information in this document has **not** been verified for AIX 4.1.

About this Document

This document describes five types of dumps/crashes in AIX 3.2 on the RISC System/6000. It includes how to tell which type of dump has occurred and, in some cases, how to locate the cause.

Running crash Subcommands

The instructions in the following sections often refer to subcommands of the "crash" command. Do the following to run crash subcommands:

1. Verify the dump device with the following command:

   ```
   sysdumpdev -l
   ```

   Note the primary dump device name, and use it where you see "/dev/hd#" in the following steps.

2. Start the crash command:

   ```
   crash /dev/hd#
   ```

   To verify the usability of the dump information, look for the following output:

   ```
   Using /unix as the default namelist file
   Reading in symbols.........
   ```

3. Crash subcommands (such as "trace" and "od", which are used in this document) can now be entered. Enter "q" to exit the crash command.

Types of Dumps

There are five basic types of dumps:

Trap — An assert statement in the code caused the system to crash because it was not true.

Invalid Operation — There is probably a wild branch, and the instruction to be executed is not valid.

DSI — (Data Storage Interrupt) This means that there was an addressing exception on a data fetch.

ISI — (Instruction Storage Interrupt) This means that there was an addressing exception on an instruction fetch.

Hang — The system is hung and a system dump must be forced by the user.

The types of dumps can be differentiated by the following symptoms:

888 102 700 0c0	This LED sequence indicates a trap or invalid operation (differentiated by errlog entry).
888 102 300 0c0	This LED sequence indicates a DSI.
888 102 400 0c0	This LED sequence indicates a ISI.
No process can proceed and no external interrupts are accepted.	This indicates a hang.

Traps

A trap is the cause of the dump if the system crashed with LED sequence "888 102 700 0c0" and if the errlog (checked with "errpt -a") contains an entry with "ERROR LABEL: PROGRAM_INT" and a "Machine Status Save/Restore Register 1" of "0002 0000".

An example of the errlog entry:

```
ERROR LABEL: PROGRAM_INT
...
Detail Data
Segment Register, SEGREG
0000 0000
Machine Status Save/Restore Register 0
0009 AC50
Machine Status Save/Restore Register 1
0002 0000
Machine State Register, MSR
0002 90B0
```

Please note the "Machine Status/Restore Register 0" (9AC50 in the above example). This is the address at which the trap occurred.

If you have a version of the crash command which accepts the -m option on the trace subcommand (trace -m), you will be able to easily identify where the trap occurred.

Here's an example (input is shown with the ">" crash command prompt):

```
> trace -m
MST STACK TRACE:
    0x1f8154 (excpt=0:0:0:0:0) (intpri=0)
        IAR:        00007148  .i_enable + ac:       bcr   14,0
        *LR:        013d87a4  [tokdd:tokoflv] + 294c
        00000000:   00000000  <invalid>
        ...

    0x2ff98000 (excpt=0:40000000:618f:2ff95508:106) (intpri=b)
        IAR:        0009ac50  .freeiblk + 54:        ti   4,r7,0x0
        *LR:        0009ac38  .freeiblk + 3c
        *2ff97288:  00000000  <invalid>
        2ff97368:   0009b084  .ifreeind + 3c
        ...
```

Find the IAR line that contains the "Machine Status Save/Restore Register 0" that was found in the errlog. The "ti" on the "IAR:" line indicates that there was a trap instruction in ".freeiblk + 54".

Invalid Operation

An invalid operation is the cause of the dump if the system crashed with LED sequence "888 102 700 0c0" and if the errlog (checked with "errpt -a") contains an entry with "ERROR LABEL: PROGRAM_INT" and a "Machine Status Save/Restore Register 1" of "0008 0000".

An example of the errlog entry:

```
ERROR LABEL: PROGRAM_INT
...
Detail Data
Segment Register, SEGREG
0000 0000
Machine Status Save/Restore Register 0
000B E4F8
Machine Status Save/Restore Register 1
0008 0000
Machine State Register, MSR
0002 90B0
```

Please note the "Machine Status/Restore Register 0" (BE4F8 in the above example). This is the address at which the invalid operation occurred.

If you have a version of crash which accepts the -m option on the trace sub-command (trace -m), you will be able to easily identify an invalid operation.

Here's an example (input is shown with the ">" crash command prompt):

```
> trace -m
MST STACK TRACE:
    0x1f8154 (excpt=0:0:0:0:0) (intpri=0)
        IAR:        00007148  .i_enable + ac:       bcr   14,0
        *LR:        013d87a4  [tokdd:tokoflv] + 294c
        00000000:   00000000  <invalid>
        ...

    0x2ff98000 (excpt=0:42000000:40001164:2ff7fffc:106) (intpri=b)
        IAR:        000be4f8  <invalid>:  ???    (0x9872c)
        *LR:        000356ec  .soo_ioctl + 634
        *2ff979f8:  00000000  <invalid>
        2ff97a58:   000a4d88  .fp_ioctl + 68
        2ff97ab8:   01d00b90   sna_sysx:luxgosp  + 9b90
        2ff97d88:   01cfdde4   sna_sysx:luxioctl  + 6de4
        2ff97de8:   00082ba8  .rdevioctl + b0
        2ff97e28:   00084054  .spec_ioctl + 20
        2ff97ed8:   0009891c  .vno_ioctl + 110
        2ff97fa8:   000a4c84  .kioctl + e8
```

Find the IAR line that contains the "Machine Status Save/Restore Register 0" that was found in the errlog.

The IAR is listed as "invalid" in the trace output; however, the rest of the stack is valid information.

Data Storage Interrupts

A DSI (Data Storage Interrupt) is the cause of the dump if the system crashed with LED sequence "888 102 300 0c0" and if the errlog (checked with "errpt -a") contains an entry with "ERROR LABEL: DSI_PROC".

An example of the errlog entry:

```
ERROR LABEL:    DSI_PROC
...
Detail Data
Data Storage Interrupt Status Register
4000 0000
Data Storage Interrupt Address Register
007F FFFF
Segment Register, SEGREG
632E 6108
EXVAL
0000 000E
```

If you have a version of the crash command which accepts the -m option on the trace subcommand (trace -m), you will be able to easily identify where the DSI occurred.

Here's an example (input is shown with the ">" crash command prompt):

```
> trace -m
MST STACK TRACE:
  0x211db0 (excpt=0:0:0:0:0) (intpri=0)
     IAR:        00009e00  .v_copypage_pwr + 58:      dclz   r7,r6
     *LR:        0005be64  .getvmpage + 128
     *00211bb8:  00000000  <invalid>
IAR not in kernel segment.
  0x2ff98000 (excpt=632e6108:40000000:7fffff:632e6108:106) (intpri=b)
     IAR:        0147ded4  [smt_load:smconnect] + cd4: l  r0,0x8(r5)
     *LR:        0147de10  [smt_load:smconnect] + c10
     *2ff97fa8:  00000000  <invalid>
     00000000:   000036dc  <invalid>
```

You can identify the correct IAR stanza by the first three values following the "excpt=" on the line above "IAR". In the above example, those three numbers are "632e6108:40000000:7fffff". Each of these three numbers can be found in the DSI_PROC error log entry: the first number is the "Segment Register, SEGREG", the second number is the "Data Storage Interrupt Status Register", and the third number is the "Data Storage Interrupt Address Register".

In the case of an ISI_PROC error log entry (see the section on Instruction Storage Interrupts): the first number is the "Segment Register, SEGREG", the second number in the "ISISR", and the third number is the "ISIR0".

A DSI (or ISI) also shows up in the vmmerrlog. The vmmerrlog information can be seen in the detail data section of the DSI_PROC (or ISI_PROC) error report entry:

```
EXVAL
0000 000E
```

You can also view the information with crash subcommands. Here's an example (input is shown with the ">" crash command prompt):

```
> od *vmmerrlog 9 a
00056a20: 20faed7f 53595356 4d4d2000 00000000  | .."SYSVMM .....|
00056a30: 00000000 40000000 007fffff 632e6108  |....@...."..c.a.|
00056a40: 0000000e                             |....|
```

In this case the return code from vmm is 0000000e. The DSISR in the above example is 40000000.

VMM Return Codes and Meanings

For all of the following codes except 00000005, there is nothing the user can do to fix the problem; the crash information must be analyzed.

For information on support available from the AIX Support Family and Program Services (IBM's base support for code-related problems), request these faxes from 1-800-IBM-4FAX:

1537 Overview of AIX Support
1760 Using Program Services
2464 The AIX Support Family

0000000e - EFAULT

This is an efault. It comes from errno.h (14) and is returned if you attempt to store to an invalid address.

fffffffa - Invalid Address Not in Memory

This is usually the result of a page fault. This will be returned if you try to access something that is paged out while interrupts are disabled.

00000005 - I/O Error

This is a hardware problem. An I/O error occurred when you tried to page in/out, or you tried to access a memory mapped file and could not do it.

Check the error log for disk or SCSI errors. If there are disk or SCSI errors for disks that are not part of rootvg and do not contain paging space, contact AIX defect support and send in the crash information.

Otherwise, contact your CE to run diagnostics.

00000086 - Protection Exception
This means that you tried to store to a location that is protected. This is usually caused by low kernel memory.

0000001c - No Paging Space
This means that the system has exhausted its paging space.

DSISR — Data Storage Interrupt Status Register

The values for the DSISR are in /usr/include/sys/machine.h. In the above example, the DSISR is 40000000 which indicates a page fault.

Instruction Storage Interrupts

An ISI (Instruction Storage Interrupt) is the cause of the dump if the system crashed with LED sequence "888 102 400 0c0" and if the errlog (checked with "errpt -a") contains an entry with "ERROR LABEL: ISI_PROC".

An example of the errlog entry:

```
ERROR LABEL:  ISI_PROC
...
Detail Data
ISISR
4000 0000
ISIR0
007F FFFF
Segment Register, SEGREG
632E 6108
EXVAL
0000 000E
```

Refer to section “Data Storage Interrupts” on page 5 of this document. The correct IAR stanza for an ISI can be found in the same way as for a DSI.

Hangs

The system is hung if no process can proceed and no external interrupts are accepted. If the system can be pinged from another node on the network, the system is not hung; instead, the application or screen may be locked up.

If the system is not truly hung, do not force a system dump. If the system is truly hung, and you have not already forced a system dump, then force the dump by turning the key to the service position and pressing the yellow button. You will see LED 0c2 and then either 0c0 or 0c4, indicating that the dump is finished.

Note

If you work with AIX support on your problem, it is useful to describe in detail the conditions and events that led to the hang. For example, "After running xyz application for two hours, there is no response to the keyboard input. I cannot get a response at the system console or at any terminals directly connected to the system unit, and I cannot rlogin or ping the system."

For information on support available from the AIX Support Family and Program Services (IBM's base support for code-related problems), request these faxes from 1-800-IBM-4FAX:

1537 Overview of AIX Support
1760 Using Program Services
2464 The AIX Support Family

Locks are one thing to look at in a dump that was forced because of a hang. A few locks are:

proc_lock
kernel_lock
net_lock

You can view lock information with crash subcommands. The following example uses proc_lock; the same format can be used for kernel_lock or net_lock. (Input is shown with the ">" crash command prompt.)

```
> od *proc_lock
0002a728: ffffffff
```

If you get all "f"s, no process is holding that lock. Otherwise, the process ID of the process holding the lock will be in the field where "ffffffff" is in the example above.

You can take a look at the locks to see if there is a deadlock situation. If there is **not** a dead-lock situation, you can look at the kernel stack trace of any process or of the running process to attempt to determine what caused the system to hang.

Viewing the Kernel Stack Trace of Any Process

1. Run the "p" crash subcommand to map a process ID with its process slot number and its name. In the example below, note that the process in SLT number 12 has a PID of cf8 and a NAME of cdpg.

   You may look at the "NAME" column to find a process in which you are interested. You may use "trace -k <process slot number>" to display its kernel stack trace, if any exists.

   For example:

```
> p
SLT ST    PID   PPID   PGRP   UID  EUID  PRI   CPU   EVENT  NAME
  0 s        0     0     0     0     0    16   120          swapper
        FLAGS: swapped_in no_swap fixed_pri kproc wake/sig
  1 s        1     0     0     0     0    60     0          init
        FLAGS: swapped_in no_swap wake/sig locks
  2 r      202     0     0     0     0   127   120          wait
        FLAGS: swapped_in no_swap wake/sig locks
...
 12 s      cf8     1   662     0     0    60     0 014d6280 cdpg
        FLAGS: swapped_in kproc orphanpgrp
...
```

2. Run "trace -k <process_slot_number>" to get the kernel stack trace information for the process. For example:

```
> trace -k 12
STACK TRACE:
  3320 (excpt=04fa5654:40000000:00000000:04fa5654:00000106) (intpri=0)
         IAR:      .e_wait+15c (0003f384):    cror 15,cr15,cr15
         LR:       .e_wait+15c (0003f384)
         2ff7fec0: .e_sleep+120 (0003f674)
         2ff7ff20: .[cfs.ext:cdr_pager]+ac (014d5720)
         2ff7ff70: .procentry+1c (00032374)
         2ff7ffc0: INVALID (00000000)
```

Viewing the Kernel Stack Trace of the Running Process

1. Run the "pcb" crash subcommand. Here is an example:

```
> pcb
        USER AREA FOR X (ProcTable Address 0xe3003200)
SAVED MACHINE STATE
curid:0x00003236  m/q:0x00040000  iar:0x014973dc  cr:0x48844084
msr:0x000090b0  lr:0x0149712c  xer:0x00000004
ctr:0x0008bf8c  *bus:0x04fc13c0
*prevmst:0x00000000  *stackfix:0x00000000  intpri:0x0000000b
backtrace:0x00  tid:0x00000000  fpeu:0x01  ecr:0x00000087
.... rest of pcb output deleted ...
```

In the example above, note that the "curid" value (on the third line of output) is 0x00003236. The process slot number is obtained from the "curid" by shifting the curid eight bits to the right. In this case, the process slot number is 0x32. Convert that value from hexadecimal to decimal; in this case: 0x32 (hexadecimal) = 50 (decimal). The process slot number is 50 (decimal).

2. Run "trace -k <process_slot_number>" to get the kernel stack trace information for the running process. For example:

```
> trace -k 50
```

Reader's Comments

Please fax this form to (512) 823-4009, attention "AIXServ Information". You may also e-mail comments to: elizabet@austin.ibm.com. These comments should include the same customer information requested below.

Use this form to tell us what you think about this document. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

If you need technical assistance, contact your local branch office, point of sale, or 1-800-CALL-AIX (for information about support offerings). These services may be billable. Faxes on a variety of subjects may be ordered free of charge from 1-800-IBM-4FAX. Outside the U.S. call 415-855-4329 using a fax machine phone.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Note: If you have a problem report or item number, supplying that number may help us determine why a procedure did or did not work in your specific situation.

Problem Report or Item #: Branch Office or Customer #:

Be sure to print your name and fax number below if you would like a reply:

Name: Fax Number:

03/15/95

END OF DOCUMENT (crash.help.krn, 4FAX# 1828)