11/16/94

Crash Command Information for TTY/Printer Subsystems

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 explains how to use the crash command and interpret its output for tty/printer subsystems.

About the crash Command 1
Sample crash Output 3
  Sample Output for a TTY on a 64-Port Concentrator 3
  Some Fields from Output on a Hung TTY 3
Explanation of crash Output 4
    Possible Values for the "port status" Field 5
    Specific Fields to Examine 5
Troubleshooting 6
CLEARTTY.c Program 7
Reader's Comments 9

The last section includes a small C program which, when all else fails, may help to revive your port so that you don't need to reboot the system. This program is NOT supported by IBM and it is NOT guaranteed.

This document applies to AIX versions 3.2.3 and later. (AIX 3.2.5 is the latest version at the time of this writing.)

About the crash Command

The "crash" command is an interactive utility for examining an operating system image, a core image, the running kernel image, or an lp or tty device image. The "crash" command facility interprets and formats control structures in the system and certain miscellaneous functions for examining dump or device values. You must have "root" authority to run this command.

In this document, we will be using crash to take a "snapshot" of the current port settings. This, at the very least, can show the current "state" or "condition" of the port and can lead to an appropriate course of action if necessary.

Crash output may seem confusing but don't be alarmed by its appearance. Most users can find what they are looking for by examining only a few fields or lines. You should read the entire fax first; then go to -- Heading 'CREXAM' unknown -- to diagnose your port problems.

Note: To make interpreting output easier, "pdisable" ttys before running the crash command. This helps because "share", "delay", "enable" and "disable" each can give a different "port status" reading (as you will see later). To disable a port enter "pdisable tty#" on the command line, where # is the number of your tty.

The following are three examples of how you can invoke the crash command. The "/0" is only necessary for a 64-port. In the second example, edit the file "crash.out" after exiting the crash command.

```
o      crash  <enter>
       > tty tty64/0
       > q

o      crash | tee crash.out   <enter>
       > tty tty64/0
       > q

o      crash <enter>
       > tty lp0/0
       > q
```

Note: If you are running crash from an ASCII terminal, the information may scroll off the screen. You can stop the scroll by quickly pressing Ctrl-S. Press Ctrl-Q to start scrolling again.

Sample crash Output

Sample Output for a TTY on a 64-Port Concentrator

```
            tty17/0: tp=0x058d8780, dev:(26,258) chan:(0x0)
sid:0x00000000 group:0x00000000 tsm:0x00000000 id:8
port status: ccnt=0
ctl=0x04f83c80 lctl=0x05885800 hptr=0x058d1130 evt=0xffffffff lck=0xffffffff
rbuf: cc=0 ``''
tbuf: cc=0 ``''
raw queue: cc=0, actual=0: ``''
can queue: cc=0, actual=0: ``''
out queue: cc=0, actual=0: ``''
ospeed=9600 ispeed=9600 stops=0 parity=0 csize=3
char map=>0x013f4458, ihog=0, ohog=0
lion discipline => I O S
dtropen discipline =>     S
xon discipline =>     S
posix discipline =>     S
erase:^H kill:^U intrc:^C quitc:^\ startc:^Q stopc:^S eofc:^D eol:^@
eol2:^@ suspc:^Z dsuspc:^Y rprntc:^R flushc:^O werasc:^W lnextc:^V
input:   brkint icrnl ixon ixoff
output:  opost onlcr nl0 cr0 tab3 bs0 vt0 ff0
cntrl:   b9600/ b9600 cs8 cread hupcl
displn:  isig icanon echo echoe echok echoctl echoke iexten
negotiated input:
negotiated output:  nl0 cr0 tab0 bs0 vt0 ff0
negotiated displn:
select:
bits: pinned
col = 0, rocount = 0, rocol = 0, ecount = 0
lead = 0, curlead = 0, curcount = 0, outleft = 0
lead queue: cc=0, actual=0: ``''
canned quote queue: cc=0, actual=0: ``''
raw quote queue: cc=0, actual=0: ``''
dlytmr = 0x058e5b80, rawtmr = 0x058e5bc0, vptr = 0x00000000
```

Some Fields from Output on a Hung TTY

The following might appear in the crash output of a hung tty (not all fields shown):

```
[Port is "ENABLED"]

sid:0x00002eb4
port status:  isopen  stop  iaslp  iow   ccnt=8
tbuf:  cc=1  ``ato^M^D^D^D^D\MATQE1^M\''
```

Explanation of crash Output

Note: Reference to many crash variables can be found in "/usr/include/sys/tty.h". Once you know what the fields are, you can use them to help solve problems.

The following is a partial listing of output fields.

tp	Pointer to tty structure.
dev	Pointer to tty structure. It's filled in by the hardware discipline at configuration or open time and is not directly used by any of the disciplines. It contains the device MAJOR and MINOR numbers.
chan	The channel number for this tty if defined for multiplex devices.
sid	The session ID for this tty. The session ID is the process ID of the session leader. It's used by tty common code and the line disciplines to implement job control.
group	The process group is the process ID of the group leader.
tsm	The process ID of the terminal state manager for this tty.
id	The tty ID for this tty structure. Used by the kernel when querying the trustworthiness of the tty.
port status	See "Possible Values for the "port status" Field."
tbuf	Transmit buffer as passed down by the higher disciplines to the hardware discipline.
ispeed	Current baud rate for the input.
ospeed	Current baud rate for the output.
stops	Current number of stop bits for the port.
parity	Current parity of the port.
csize	Current character size for the port.
char map	Points to the character length and display width vector for the native language used.
seg	Address of the segment register for normal access.
port	Adapter offset.
iseg	Address of segment register for ID access.
slih queue	Second level interrupt queue.
rtrig	Receive trigger.
tbc	transmit buffer count.

Last Modem Status Contains different values based on which signals are currently active: DTR, RTS, CTS, DSR, RI, or CD.

Possible Values for the "port status" Field

wtopen At least one process is currently opening the tty port.

isopen At least one process has completely opened the tty port.

iclose At least one process is currently closing the tty port.

busy The hardware discipline sets this bit whenever output is actively moving out of the port.

stop The hardware sets this bit whenever it receives a "T_SUSPEND" command and clears it when it receives a "T_RESUME" command.

iaslp At least one process is sleeping or waiting for input.

oaslp At least one process is sleeping or waiting for output to fall below the low-water mark in the output queue.

iow At least one process is sleeping or waiting for output to completely drain.

ccnt This count is used by "ttyclose" to determine when the last close is complete. This field is **normally** zero — any other value may indicate a problem on this tty.

Specific Fields to Examine

port status: This field is one of the most important and can quickly show the current state of the port. With the port "disabled", this line should appear as:

```
        port status: iclose  ccnt=0
or      port status: ccnt=0
```

For other values for "port status", see “Possible Values for the "port status" Field.”

sid: This field contains the process ID for the port. An unused port should appear with the following entry:

```
sid:0x00000000
```

An entry like `sid:0x00002be1` is showing `2be1` as the process ID. To see the process name itself, issue a "p" command while in crash and look for `2be1` in the PID column. On the far right, you will see the process name. Careful, you may need to issue a Ctrl-S to freeze the screen! (Use Ctrl-Q to start scrolling again.)

rts discipline =>: This line will appear **only** if RTS (hardware handshaking) has been added to the port. If your device requires use of RTS, look for this line.

ospeed=38400 ispeed=38400 stops=0 parity=0: Examine this line when you are experiencing trouble with baud rates.

Troubleshooting

If you encounter the following error:

```
Port is locked up and users cannot log in to terminal or modem
```

try the following suggestions. (Substitute "34" with the appropriate number for your tty.)

1. Try issuing the command "pstart 34"; check to see if this helped.
2. Try issuing commands "pdisable 34" and "penable 34" — can they log in?
3. Are you receiving any errors on the console?

   An error such as "init: command respawning too rapidly" can mean that the system is receiving line noise on the port. This can be due to improper cabling, loose cabling, inadaquate shielding on cables, or incorrect settings on modems.

   The "respawn" error means that the system is saving CPU time by shutting the port (getty) down and not wasting valuable interrupts to a problem port. The message is given when the system senses about five errors within 3 minutes 45 seconds. If left alone, the system will revive the port after about 19 minutes. If the problem continues, the system will leave the port in a down state and it will take a reboot to reset it (or a program similar to the CLEARTTY.c program listed later).

   If using IBM cabling, check to see that all cables are tightly connected. If you still have problems, try substituting an IBM serial cable to the device.
4. If possible, move the terminal or modem to another port. Did the problem follow the device to the new port? If so, the problem may be hardware.
5. For this type of problem always run an "errpt [-a]" to see if there are any messages indicating a problem on an adapter or device.
6. Is there a "getty" running on the port?
7. Look at the PID on the getty. Does it change when you issue a "pdisable tty#", then "penable tty#" command?
8. If the device is a modem and you cannot dial into it, check the following:
   - Cycle power on the modem to reset it, and try again.
   - With the modem on, check to see if any respawn errors are on the console. If so, check that "clocal" is not set in the tty "runtime attributes" — if it is, the modem should not be set to echo back commands to the system. This will cause a respawn.
   - Verify cabling again and keep it away from high line-noise sources.
   - Verify that the modem does not have "carrier" strapped high (&C0). This will allow you to dial out but no one can dial in.
   - Look for a "lock" file in /etc/locks. If there, remove it.

- Verify the modem's "carrier" and "dtr" settings (normally &C1 and &D3).

CLEARTTY.c Program

The following C program may help bring a tty port back without a reboot. This program is supplied "AS IS" and is intended as an aid only. It is NOT supported by IBM and IBM is not responsible for its use or misuse.

Note: In the following code, some characters may appear incorrectly. If the characters in the following list do not match their descriptions, be sure to change them in the code.

[left bracket
] right bracket

```
/*                              CLEARTTY.c
        Program to attempt a reset of tty port without rebooting.
        To test the effectiveness of this program, do the following
        both before and after running cleartty:
                    pdisable tty0
                    echo "tty tty0" | crash > crash.out

 NOTE:  If tty0 is on a 16-port concentrator off a 64-port Async
        Adapter, you would replace "tty0" with "tty0/0".

        Edit the "crash.out" file created above and look at the
        "port status" line.  A normal, "disabled" port should have
        a status of "iclose ccnt=0".
_____________________________________________________________________

  To create program:
              o "vi cleartty.c"
              o  enter the following C program and save the file

  To compile program:   "cc -o cleartty  cleartty.c"
  To execute program:   "cleartty /dev/tty0"
_____________________________________________________________________ */


/*   PROGRAM  BEGINS  HERE                                            */

#include <errno.h>
#include <fcntl.h>
#include <stdio.h>
#include <termios.h>

main(int argc, char *argv[])
{
        int fd;
        if (argc < 2) {
                fprintf(stderr, "Usage: %s /dev/tty?\n", argv[0]);
                exit(1);
        }
        if ((fd = open(argv[1], O_RDWR | O_NONBLOCK)) < 0) {
                perror(argv[1]);
                exit(errno);
        }
        if (tcflow(fd, TCOON) < 0) {
                perror("tcflow");
                exit(errno);
        }
        close(fd);
        exit(0);
}
/*   PROGRAM  ENDS  HERE                                              */
```

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:

END OF DOCUMENT (crash.ttylp.zap, 4FAX# 2263)

DSMBEG323I STARTING PASS 2 OF 2.
+++EDF258W Cross references were not resolved. Check cross-reference listing to find problems. (Pag
e 10 File: CRTTY SCRIPT)
DSMMOM397I '.EDF#END' WAS IMBEDDED AT LINE 187 OF 'BOOKPROF'