report

MIT Kavli Institute
1 Hampshire Street
Cambridge MA 02139–4307
Tel: 617-253-7277
Fax: 617-253-8084

To: ACIS Science Operations Team
From: Peter Ford, NE80-6071 <pgf@space.mit.edu>
Date: June 29th 2011
Subject: Testing SIMODEs without a Hardware Simulator (v 2.0)

Introduction

On February 10th 2010, the ACIS Engineering Unit (EU) malfunctioned. Although the Instrument Team worked hard to diagnose the problem and replace the failed component, the EU was not available again until March 2nd. Another hardware failure occurred on April 20th 2011 which was fixed on May 20th. The EU was then menaced by a water leak on May 21st stemming from air conditioning problems on the 7th floor OCC. Our procedures require that all new ACIS science mode (SIMODE) commands must be tested on the EU before they are made available to the offline system for inclusion in daily loads. This report addresses the risks of partial or complete EU failure, and their various mitigations.

Engineering Unit Status

All replacement BEP, FEP, DEA, and interface boards have been tested and found to be in working condition. We have also tested the spares for the other components that are required for the EU to be functional, with the results shown in the following table, showing the minimum number required to execute all functions. The rightmost column identifies 2 items for which we do not currently possess working spares, and whose failure would cause the EU to be unavailable for an extended period.

Unit	Abbreviation	Flight	EU[a]	Minimum	Risk?
Analog Board	DEA	10	9+1[b]	2	
Interface Board	Board 11, 12	2	2+2	1	
DEA Backplane		1	1+0+1	1	Yes
Front End Processor	FEP	6	6+3+3	2	
Back End Processor	BEP	2	2+1	1[c]	
DPA Backplane		1	1+0+1	1	Yes
Focal Plane Assembly	FPA	1	1+0	0[d]	
Power Supply	PSMC	1	1+0	1[e]	
Image Loader		N/A	2+1	1	
DPA Input Switch		N/A	1+0	1[f]	
Telemetry Control Unit	L-RCTU	N/A	3+0	1	
Workstations & interfaces		N/A	4+3	1	

[a] The first number indicates the quantity currently installed, the second the number of tested spares, the third (if specified) the number of spares that could be made to function after assembly or repair.

[b] 6 “flight” boards are equipped with full housekeeping sensors; the remainder are used with a software patch.

[c] 2 BEPs are needed for testing DPA A/B redundancy; otherwise, one is sufficient.

[d] The focal plane is available but isn’t required for testing and hasn’t been attached since launch.

[e] The PSMC output to the DPA can be replaced by a pair of laboratory power supplies, which are available and fully tested. PSMC output to the DEA is redundant. If both sides fail, they can be replaced by a set of power supplies.

[f] If the switch fails, cable connectors can be swapped manually.

There are three main reasons for maintaining the EU: (1) to test parameter blocks before they are included in daily loads; (2) to develop and test flight software patches; and (3) to help diagnose—and hopefully repair or mitigate—anomalies that are discovered in flight hardware or software. A working EU with minimum components as listed in the preceding table, will always be essential for (2) and (3), but carefully written software should be capable of validating SIMODE commands at least as well as is currently done with the EU.

ACIS Software Simulator

During ACIS development, it was recognized that most of the flight software could be run on the ground in a suitable simulator. The BEPs and FEPs use Motorola$^{\text{TM}}$ R3000 processors whose MIPS architecture is a subset of the RISC CPUs used in the DECstation 3000 series of workstations from the Digital Equipment Corporation[1]. The ACIS instrument team retains 4 "mothballed" DECstations with the express purpose of running the software simulator, but it is little used because it is slow (< 1% of real-time), can only support 1 or 2 FEPs without bogging down completely, and cannot easily support patch development; nor can it be integrated into regression testing.

Testing SIMODES with the EU

Currently, all new ACIS command loads are tested with a PERL script named *run-dat3.pl*. This program was designed to be run from any workstation with network access to *cypress*, the computer that maintains the hardware interfaces to the EU via the Image Loader and the L-RCTU. Networking incompatibilities have largely restricted access to workstations running the ancient SunOS 4.1 operating system, and most often to *cypress* itself.

run-dat3.pl is supplied with the pathnames of the ACIS tables to be sent to the CXC Off-Line System, along with a list of the new SIMODEs to be tested.

```
run-dat3.pl [ options ] simode1 [simode2 ...]

  -D         Run in debug mode: list packets, test images, but don't sent to EU.
  -Q         Quicken the pace: shorten inter-command delay by 250 msec.
  -L dir     Directory containing the ACIS uplink command tables.
  -c file    Uplink table file containing SIMODE configuration arrays.
  -d file    Uplink table file containing ACIS command packets.
  -h host    Network host with access to the Image Loader and L-RCTU.
```

The procedure forks a shell process that monitors the output of the EU and, when it receives an exposure header packet, sends a *stopScience* command back to the EU. It exits when it receives a *scienceReport* packet, indicating that the run has ended, or a *fatalMessage*, or *bepStartupMessage* indicating that the BEP crashed. Meanwhile, the parent script examines the command packets associated with the given SIMODE, passes them to the EU, and waits for the delay time specified in the SIMODE configuration array. If the command is a *loadCcBlock* or *loadTeBlock*, *run-dat3.pl* invokes the *genObjectImage* to create a suitable image that will serve as input to the FEPs for bias creation and then for event detection. It is sent to the Image Loader via the *putImages* command. The SIMODE is considered to have "passed" if the following criteria are met:

- The uplink command files were read without error.
- The SIMODE was located in the configuration array table file.
- Its commands were located in the command packet table file.
- Each command packet was of the correct length.
- Images constructed from the parameter block fields were acceptable to *genObjectImage*.
- The EU accepted the commands and ran to the point of generating exposure header packets.

[1] Sold to Compaq in 1998 and merged with Hewlett-Packard in 2002

Example

The following table was generated by *run-dat4.pl,* a slightly updated version of *run-dat3.pl* that can reboot the BEP after detecting a serious error, and therefore test large numbers of SIMODEs automatically. In this example, it used the EU to validate all ‘CC’ and ‘TE’ SIMODEs in the uplink tables that had a ‘B’ suffix, *i.e.*, those that force bias maps to be created.

```
acis# run-dat4.pl -L /acis/h1/www/bin -A -R -S ; make report
Patch level E-opt-E:
standard dearepl smtimedlookup cc3x3 eventhist compressall untricklebias

       Date           Run   siMode    pBlock   runBlock Flg Comments
Tue Dec  8 17:51:09    4 CC_00008B 00008014 00008014 R   DEA_IO_ERROR
Tue Dec  8 17:53:15    5 CC_0000AB 0000a014 0000a014 R   (cc_raw_faint)
Wed Dec  9 09:50:35  133 TE_00058B 00058014 00058014 R   (te_raw_faint)
Wed Dec  9 09:54:57  134 TE_0005AB 00058014 00058014 R   (te_raw_faint)
Wed Dec  9 10:00:00  135 TE_0005CB 0005c014 0005c014 R   (te_hist_faint)
Wed Dec  9 10:44:58  138 TE_00062B 00062014 00062014 R   DEA_IO_ERROR
Wed Dec  9 10:47:05  139 TE_00064B 00064014 00064014 R   (te_raw_faint)
Thu Dec 10 15:37:39  225 TE_0013CB 0013d014 -------- P   recomputeBias=0
Thu Dec 10 15:37:40  226 TE_0013EB 0013f014 -------- P   recomputeBias=0
Fri Dec 11 22:33:47  327 TE_00212B 00212014 00212014 TR  (te_ev5x5_faint)
Sat Dec 12 04:15:10  343 TE_00232B 00233014 -------- P   recomputeBias=0
Sat Dec 12 06:12:55  349 TE_0023EB 0023e014 0023e014 RD  DEA_IO_ERROR
Sun Dec 13 09:09:04  449 TE_00306B 00307014 -------- P   recomputeBias=0
Sun Dec 13 19:25:34  489 TE_00366B 00366014 00366014 RD  (te_raw_faint)
Sun Dec 13 19:40:07  490 TE_00368B 00368014 00368014 RD  (te_raw_faint)
Mon Dec 14 01:30:03  511 TE_0039CB 0039c014 0039c014 TRD (te_ev3x3_faint)
Mon Dec 14 03:55:00  520 TE_003AEB 003af014 -------- P   recomputeBias=0
Mon Dec 14 03:55:01  521 TE_003B0B 003b1014 -------- P   recomputeBias=0
Mon Dec 14 03:55:02  522 TE_003B2B 003b3014 -------- P   recomputeBias=0
Mon Dec 14 03:55:03  523 TE_003B4B 003b5014 -------- P   recomputeBias=0
Mon Dec 14 16:45:19  575 TE_0041CB 0041d014 -------- P   recomputeBias=0
Mon Dec 14 16:45:20  576 TE_0041EB 0041f014 -------- P   recomputeBias=0
Mon Dec 14 16:45:21  577 TE_00420B 00421014 -------- P   recomputeBias=0
Mon Dec 14 18:01:59  583 TE_0042CB 0042d014 -------- P   recomputeBias=0
Mon Dec 14 19:14:15  588 TE_0043EB 0043f014 -------- P   recomputeBias=0
Mon Dec 14 19:14:17  589 TE_00440B 00441014 -------- P   recomputeBias=0
Mon Dec 14 20:29:41  595 TE_0044CB 0044d014 -------- P   recomputeBias=0
Mon Dec 14 20:51:47  597 TE_00450B 00451014 -------- P   recomputeBias=0
Mon Dec 14 20:51:48  598 TE_00454B 00455014 -------- P   recomputeBias=0
Tue Dec 15 01:29:12  613 TE_00472B 00473014 -------- P   recomputeBias=0
Tue Dec 15 01:29:13  614 TE_00474B 00475014 -------- P   recomputeBias=0
Tue Dec 15 01:42:42  616 TE_00478B 00479014 -------- P   recomputeBias=0
Tue Dec 15 01:42:43  617 TE_0047AB 0047b014 -------- P   recomputeBias=0
Tue Dec 15 01:42:44  618 TE_0047CB 0047d014 -------- P   recomputeBias=0
Tue Dec 15 01:42:45  619 TE_0047EB 0047f014 -------- P   recomputeBias=0
Tue Dec 15 11:00:19  651 TE_004BEB 004be014 004be014 R   (te_ev3x3_faint)
Wed Dec 16 04:18:25  700 TE_00524B 00525014 -------- P   recomputeBias=0
Wed Dec 16 04:18:26  701 TE_00526B 00527014 -------- P   recomputeBias=0
Wed Dec 16 04:18:27  702 TE_00528B 00529014 -------- P   recomputeBias=0
Wed Dec 16 06:11:54  709 TE_0053AB 0053b014 -------- P   recomputeBias=0
Sun Dec 20 06:05:54 1046 TE_00858B 00858014 00858014 TR  (te_ev5x5_faint)
```

The meaning of the ‘`Flg`’ codes in column 6 are as follows: **`D`**: the parameter block contained a non-zero *deaLoadOverride* field; **`P`**: an unsuitable parameter block field was found; **`R`**: the BEP crashed; **`T`**: a BEP command timed out. After an error of type ‘`R`’ or ‘`T`’, the BEP was warm-booted and the script continued.

Testing SIMODEs with Software

The philosophy behind hardware testing is to send the commands to the EU and see what happens. Although this approach only uncovers errors that prevent runs from completing, it surely prevents ACIS from sitting idle while X-rays are hitting the CCDs! The software test script, *run-dat5.pl,* has no hardware object to verify the correctness of the loads, so it must contain the full knowledge of the syntax and semantics of ACIS commands and how they relate to one another.

```
run-dat5.pl [ options ] { -A | -B | simode1 [simode2 ...] }
```

`-A`	Test all CC and TE SIMODEs in the ACIS uplink tables.
`-B`	List the contents of all SIMODEs in the ACIS uplink table files.
`-b`	Check consistency of *recomputeBias* and *trickleBias* fields ***vs***. the SIMODE 'B' suffix.
`-c` *file*	Uplink table file containing SIMODE configuration arrays.
`-D`	List the contents of the command packets.
`-d` *file*	Uplink table file containing ACIS command packets.
`-L` *dir*	Directory containing the ACIS uplink command tables.
`-m` *pgm*	Use *pgm* to generate PRAM from the contents of the parameter block.
`-p` *pgm*	Use *pgm* to check the PRAM generated by the parameter block.
`-S` *file*	Use *file* to supply the default SRAM when validating parameter blocks.
`-s` *n*	When the –A and/or –B flags are used, skip the first *n* SIMODEs.
`-v`	Be verbose.

Since *run-dat5.pl* can test an SIMODE in a tiny fraction of the time taken by *run-dat3.pl,* the –A and –B flags have been added to let the script check *every* entry in the uplink tables. Note that, without an EU, the script must validate the *loadCcBlock* and *loadTeBlock* parameter blocks itself, which it does by running the *buildPram* program to generate PRAM and SRAM microcode, and then *pchk* to execute it and count the frequency of CCD row-shift transitions. (Alternative commands can be specified by the –m and –p options). Both routines are just wrappers around flight software routines that perform the same functions in the BEP: to identify and prevent science runs that would potentially damage the DEA hardware. In all, the following tests are performed:

- The uplink command files were read without error.
- The SIMODEs are defined in the ACIS configuration table.
- All SIMODE packets are defined in the ACIS command packet table.
- The content of each packet can be parsed by *lcmd*. *
- The packet types are valid within an SIMODE. *
- The packet delays are sufficient. *
- No more than 6 CCDs have been powered up by *changeSystemConfig* commands. *
- The required CCDs have been powered up by *changeSystemConfig* commands. *
- The required FEPs have been powered up by *changeSystemConfig* commands. *
- If needed, a window block has been loaded into the correct slot. *
- The *startScience* command refers to the same slot that the parameter block was stored in. *
- The parameter block generates acceptable PRAM.
- Overridden DEA microcode (+ default SRAM) is acceptable.

If the –b flag is specified, warning messages are generated in the following situations:

- The parameter block commands that FEP source code is to be overwritten. *
- An SIMODE ('CC' or 'TE') is suffixed with 'B', but its parameter block doesn't force bias maps to be recomputed (*recomputeBias*=`0`) or to be copied to telemetry (*trickleBias*=`0`). This warning is only issued if the science run actually uses bias maps. *

- Conversely, a warning will be issued for any SIMODE without the 'B' suffix if its parameter block forces bias maps to be recomputed (*recomputeBias*=1). *

The asterisks denote the tests that are not currently being performed by *run-dat3.pl*.

Comparison between EU and Software Tests

Although the software tests appear to be more thorough than those on the EU, the semantics are subtly different, as illustrated in the following table:

run-dat3.pl using the EU	run_dat5.pl using software
Testing runs slowly (~30 minutes per SIMODE) and requires the exclusive use of the EU.	Testing is very fast (< 1 second per CPU) and can run on any computer with PERL installed and which can compile the ACIS flight software routines responsible for PRAM generation and testing.
An SIMODE may contain any combination of ACIS software serial commands. The EU figures out whether they conflict in a manner than prevents science event processing.	An SIMODE may contain any mixture of ACIS software serial commands, and their contents will be validated by *lcmd*, but only a subset[a] of commands will be examined in detail.
All parameter block and window block slots have valid contents, although not necessarily those that the observer intends. When testing a sequence of SIMODEs, the contents of parameter and window blocks, and of temporary DEA microcode, will remain from run to run.	Only parameter block and window block slots that are updated by commands in the current SIMODE will be assumed to contain valid fields.
WriteBep commands can perform many possible functions, *e.g.*, update PRAM or SRAM microcode, override FEP software, patch commands and data in BEP or FEP memory, etc., etc.	*WriteBep* commands are assumed to be restricted to updates of PRAM and/or SRAM microcode. Multiple *writeBeps* must write to contiguous memory, originating at the location specified in the *dealLoadOverride* fields of the accompanying parameter block. Any other use will be flagged as an error.
ExceuteBep commands will be tested.	*ExecuteBep* commands will be tested for correct format, but no further.
Errors related to flight software bugs triggered by parameter block or *writeBep* contents might be caught.	Errors related to flight software bugs triggered by parameter block or *writeBep* contents will not be caught.
In *run-dat3.pl*, SIMODEs that don't generate exposure packets will not be terminated automatically by the test script and must be stopped manually. In *run-dat4.pl*, such SIMODEs will time out and the script will move on to test the next SIMODE.	No test is performed to determine whether an SIMODE is capable of generating exposure packets; *e.g.*, if the DEA microcode is overridden, the pixel data stream read by the FEPs may be incompatible with the parameter values extracted by the BEP from the SIMODE parameter block.

[a] Only the following commands are examined: *changeConfigSetting, dumpHuffman, dumpSysConfig, executeBep, load1dBlock, loadCcBlock, loadTeBlock, load2dBlock, startScience, stopScience*, and *writeBep*.

Example 1

A single SIMODE, `TE_004DE`, is tested with verbose output, showing the names of the individual command packets, their requested post-execution delay time in seconds, and the minimum delay time estimated by the *run-dat5.pl* script.

```
acis# run-dat5.pl -L /acis/h1/www/bin -v TE_004DE
/acis/h1/www/bin/current.cfg: 2669 cfgs
/acis/h1/www/bin/current.dat: 3986 pkts
------------------------------------------------------------
SImode: TE_004DE   Date: Wed Mar 31 12:35:05 US/Eastern 2010
WSVIDALLDN  Delay: 18/4
WSPOW0CF3F  Delay: 63/54
WT004DF024  Delay: 4/4
RS_0000001  Delay: 4/4
RH_0000001  Delay: 23/23
XTZ0000005  Delay: 4/4
Ending processing of TE_004DE
------------------------------------------------------------
Ending:    Date: Wed Mar 31 12:35:06 US/Eastern 2010
------------------------------------------------------------
acis#
```

Example 2

All valid SIMODEs of the ACIS uplink tables—*i.e.*, those beginning 'CB', 'CC', 'CN', 'TB', 'TE', or 'TN'—were tested in a single execution of *run-dat5.pl*. None of these bad SIMODEs were ever included in a daily load and most of the errors are caused by omitting the necessary *changeSystemConfig* and *writeBep* commands from the SIMODE. Note that most of the bad SIMODEs detected by the EU and listed in the table on page 3 were removed from the uplink tables before the current test was run.

```
acis# run-dat5.pl -L /acis/h1/www/bin -A
CN_0005A  WC0005B014: CCD 0 not powered up
CN_0005AB WC0005B014: CCD 0 not powered up
CN_0005C  WC0005D014: CCD 0 not powered up
CN_0005CB WC0005D014: CCD 0 not powered up
CN_0005E  WC0005F014: CCD 0 not powered up
CN_0005EB WC0005F014: CCD 0 not powered up
TB_00112  WSPOW3F03F: insufficient delay 18 secs (< 54 secs)
TB_00112  WT00112014: CCDs 4,5,6,8,9 not powered up
TE_0003C  WSPOW0CF3F: insufficient delay 9 secs (< 54 secs)
TE_0003CB WSPOW0CF3F: insufficient delay 9 secs (< 54 secs)
TE_00042  WSPOW3F03F: insufficient delay 9 secs (< 54 secs)
TE_00042B WSPOW3F03F: insufficient delay 9 secs (< 54 secs)
TE_00044  WSPOW0CF3F: insufficient delay 9 secs (< 54 secs)
TE_00044B WSPOW0CF3F: insufficient delay 9 secs (< 54 secs)
TE_00046  WSPOW00F0F: insufficient delay 9 secs (< 36 secs)
TE_00046B WSPOW00F0F: insufficient delay 9 secs (< 36 secs)
TE_00048  WSPOW1F83F: insufficient delay 9 secs (< 54 secs)
TE_00048B WSPOW1F83F: insufficient delay 9 secs (< 54 secs)
TE_00366  WT00367014: CCD 4 not powered up
TE_00446  WSPOW3F03F: insufficient delay 9 secs (< 54 secs)
TE_00446B WSPOW3F03F: insufficient delay 9 secs (< 54 secs)
TE_0053A  WT0053B014: BEP load address 0x8003ec98 empty
TE_0053AB WT0053B014: BEP load address 0x8003ec98 empty
TN_00142  WT00143014: no window block loaded into slot 4
TN_00142B WT00143014: no window block loaded into slot 4
992.80u 789.52s 49:07.19 60.4%
acis#
```

SIMODE Validation

The desired SIMODE tests (see page 4) were reviewed for completeness by the ACIS operations team. The *run-dat5.pl* script and associated binaries, *buildPram* and *pchk*, were subjected to numerous checks, including a coverage test using the `Devel::Cover` module distributed by CPAN. Inputs included the contents of the current OFLS ACIS tables, augmented by several fake tables constructed with known errors. The coverage statistics verified that all desired tests were being performed, and a visual inspection of the *run-dat5.pl* output verified that the errors were being reported correctly.

Conclusions

There is general agreement that the current procedure—whereby all ACIS commands to be included in the daily loads are first tested on the Engineering Unit—has been necessary and sufficient to guarantee that all science runs are executed in the manner that the investigators proposed. The only reason for considering an alternative is that the EU is growing old and, from our recent experience, takes time to repair.—a few days for simple problems, but potentially much longer, witness the month taken to recover from the April 2011 anomaly. This is exacerbated by the upcoming reduction to part-time employment of the technician who possesses the most experience working with ACIS hardware.

Through the above examples that examined very large numbers of SIMODE commands, software testing through *run-dat5.pl* and hardware simulation with the EU have located the same errors, and a number of additional errors have surfaced through software testing alone. However, the software tests are less flexible since they assume a particular pattern of commands for each SIMODE, i.e.,

- One or more *changeSystemConfig* commands to turn DEA and FEP boards on or off.
- A single *loadCcBlock* or *loadTeBlock* command to load a parameter block.
- A single *load1dBlock* or *load2dBlock* command to load a window block.
- Optionally, one or more *writeBep* commands to define a block of replacement DEA microcode.
- A single *startScience* command.
- Any number of *stopScience, dumpHuffman, dumpSysConfig,* and *executeBep* commands.
- Any other command will be reported as an error.

Since all current SIMODEs confirm to this command structure, even those such as the "squeegee-mode" runs of 2004-5, that employ non-standard DEA microcode, software testing is currently a viable alternative. Should SIMODE commanding become more complex in the future, *e.g.*, as a result of hardware degradation or flight software enhancements, EU validation would probably remain unchanged, whereas software test scripts would probably need to be rewritten.

A PSMC failure would not be entirely unexpected. The PSMC comprises redundant outputs for DPA and DEA. Both "sides" are required to power the DPA, but since both supply +5V, they are easily replaced by existing laboratory supplies. The DEA requires power at ±6V, ±15V, +24V, and +28V, but either side can power all DEA boards, so the most likely PSMC failure would still be able to power the DEA through the remaining outputs. Even a total failure could be circumvented by acquiring a series of power supplies at the various voltages. Since we possess a full set of PSMC design documents and schematics, and a number of replacement boards and components, we should be able, given sufficient time, to repair the PSMC, but laboratory power supplies would keep the EU running while this was going on.

Recommendations

1. Continue to verify all SIMODE command loads using the EU, using either *run-dat3.pl* or *run-dat4.pl.*
2. Also run all SIMODEs through *run-dat5.pl* and compare the results to those obtained from the EU, to gain confidence that we can rely on *run-dat5.pl* alone were the EU to be unavailable.
3. Estimate the cost and document the steps needed to assemble a replacement DEA and/or DPA backplane from the current components, should either of the EU's existing backplanes fail irreparably.

NAME

run-dat3.pl – execute one or more ACIS configurations on the EU.

SYNOPSIS

run-dat3.pl [**–D**] [**–Q**] [**–X**] [**–L** *dir*] [**–c** *cfg*] [**–d** *dat*] [**–h** *host*] *item1* [*item2* . . .]

DESCRIPTION

This command extracts one or more ACIS configuration *items* and their associated packets from ACIS tables and sends them to a *cserver*(1) process running on a remote *host*. If a configuration includes a *loadCcBlock* or *loadTeBlock* packet, *run-dat3.pl* also constructs a suitable test image and loads it into the Image Loader on the same remote host.

While ACIS is being configured for a science run, *run-dat3.pl* lists the names of the packets that are being sent and accumulates the telemetry that is generated. Then, after sending the *startScience* command, it prints all of the *commandEcho* packets and then waits until the first exposure packet. It then sends a *stopScience* command and waits for the *scienceReport* command before proceding to the next configuration *item*.

If the **–D** flag is specified, *run-dat3.pl* executes in "*diagnostic*" mode—no commands or images are sent to the remote host. Instead, the table contents are verified and *lcmd*(1) is used to display their contents on the standard output stream.

OPTIONS

–D diagnostic mode—don't execute the ACIS commands. Instead, check them for syntax and use *lcmd*(1) to write their formatted contents to *stdout*.

–Q delay for 0.25 seconds *less* than the time specified in the CFG file, to simulate the worst-case round-off error within the OBC.

–X this flag is used internally; it should *never* be specified on the *run-dat3.pl* command line.

–L *dir* look for *cfg* and *dat* tables in the *dir* directory. If not specified, *run-dat3.pl* looks for them in the current directory.

–c *cfg* the name of the ACIS configuration table file.

–d *dat* the name of the ACIS packet table file.

–h *host* the remote host to which to send the ACIS command. It is assumed that *cserver*, *shim*, and *filterServer* processes have already been started on that machine.

AUTHOR

Peter G. Ford, MIT CSR

SEE ALSO

cserver(1), filterServer(1), nap2dat.pl(1), shim(1)

DIAGNOSTICS

warning: tag ´*tag*' not found

the named *tag* is missing from the ACIS configuration table.

NAME

run-dat4.pl – execute simodes on the ACIS engineering unit

SYNOPSIS

run-dat4.pl [**−ABDEQRSv**] [**−L** *dir*] [**−c** *cfg*] [**−d** *dat*] [**−h** *host*] [**−p** *port*] [**−s** *skip*] [**−w** *secs*] [*siMode* [*siMode...*]]

DESCRIPTION

In its default operating mode, this command extracts one or more *siModes* from ACIS tables and sends their command packets to a *cserver*(1) process for execution on the ACIS Engineering Unit (EU). If an *siMode* includes a *loadCcBlock* or *loadTeBlock* command, a suitable test image is constructed and piped through *genObjectImage* and *putImage* into the Image Loader on the *cserver* host.

run-dat4.pl lists on *stdout* the names of the command packets that are being sent. It also accumulates the resulting telemetry, using *ltlm* to print to *stdout* only the *commandEcho* packets. When the first *exposure*∗ packet is received, it sends a *stopScience* command to the EU and waits for a *scienceReport* packet before proceding to the next *siMode*. If the run times out (the **−w** option) or ends prematurely without producing an exposure packet, *run-dat4.pl* kills the *filterClient* and *ltlm* processes and goes on to the next *siMode*.

If either the **−B** or **−D** flag is specified, *run-dat4.pl* executes in *“diagnostic´´* mode—no commands or images are sent to the remote host. Instead, the command packets are passed to *lcmd*(1) to display their contents on the standard output stream.

OPTIONS

−A Execute *all* of the CC and TE *siModes* in those ACIS table files specified by the **−L**, **−c**, and **−d** options. Unless **−S** is specified, the *siModes* will be executed in random order. However, when used in combination with **−B**, no commands or images are sent to the EU. Instead, **−A** modifies the usual **−B** listing to include only the contents of the WC and WT parameter blocks contained within CC and TE configurations.

−B Don’t send commands or images to the EU. List the contents of all *siMModes* in the ACIS table files specified by the **−L**, **−c**, and **−d** options. Unless **−S** is specified, the *siModes* will be listed in random order. When used in combination with the **−A** flag, only the contents of WC and WT parameter blocks contained within CC and TE configurations are listed.

−D Diagnostic mode—don’t send commands or images to the EU. Instead, check them for syntax and use *lcmd*(1) to write their formatted contents to *stdout*. **−D** differs from **−B** in two respects: it displays individual *siModes* that are specified on the command line, and it is slower since it forks a process to run *ltlm* for each command packet.

−E When creating an image for the Image Loader, inspect other entries in the parameter block and ignore the *siMode* if any of the following *“undesirable´´* characteristics is detected:

recomputeBias == 0

Don’t run the *siMode* unless it forces a bias recomputation. For this reason, the **−A** flag will always skip such runs, whether or not **−E** was specified in the command line.

trickleBias == 0

Don’t run the *siMode* unless it commands a bias readout.

fepLoadOverride != 0

Skip *siModes* that call for non-standard FEP flight software.

deaLoadOverride != 0

Skip *siModes* that call for non-standard DEA microcode.

histogramCount ∗ *primaryExposure* ∗ *nfeps* > 10000

Skip a raw- or event-histogram run if the time from the start of histogram creation to the first *exposure*∗ packet is greater than ˜1000 secs plus the bias time. (*nfeps* is the number of FEPs to be used.) Note that the **−R** flag can also be used to speed up raw- and histogram-mode runs.

–L *dir* Look for *cfg* and *dat* tables in the *dir* directory. The default is the current directory.

–Q Delay waiting after sending each packet to *cserver* by 0.25 seconds *less* than the time specified in the CFG file, to simulate the worst-case round-off error within the OBC.

–R When executing raw- or histogram-mode runs, don't issue a *stopScience* command when the first *exposure∗* packet is received. Instead, reboot the BEP after the first *data∗* packet, and then process the next *siMode*. This eliminates much waiting time for these modes, which might otherwise end in a timeout and then a reboot.

–S Used in conjunction with the **–A** and/or **–B** flags to process the *siModes* in ascending alphabetical order.

–c *cfg* The name of the ACIS configuration table file. The default is *“acis.cfg´´*.

–d *dat* The name of the ACIS packet table file. The default is *“acis.dat´´*.

–h *host* The name of the host running *cserver*, to which to send the ACIS commands and images. The default is *“cypress´´*.

–p *port* The port number on the host running *cserver*, to which to send the ACIS command. The default is 7000.

–s *n* When the **–A** and/or **–B** flags are used, skip the first *“n´´ siModes*. The default is not to skip any. This option is usually used in combination with **–S**.

–w *secs* The maximum time (in seconds) to wait between sending the first command and reading the first *exposure∗* packet. The default is 1800 seconds. If this limit is exceeded, *run-dat4.pl* will reboot the BEP and go on to the next *siMode*.

–v Be verbose: report the number of records read from the ACIS tables and include exposure packets among those listed coming from the EU. Each output line from *ltlm* will be preceded by two numbers, the number of exposure packets and the number of run-terminating packets received. If the **–v** flag is repeated, also list the scripts that are sent to the Image Loader through *genObjectImage* and *putImage*.

AUTHOR

Peter G. Ford, MIT CSR

SEE ALSO

cserver(1), filterServer(1), genObjectImage(), ltlm(1), putImage(1), run-dat3.pl(1) shim(1)

DIAGNOSTICS

Rebooting the BEP

After a timeout, a *fatalMessage*, or *bepStartupMessage* packet, or a *scienceReport* packet with a bad *terminationCode*, the BEP will be warm-rebooted, to reload patches. *run-dat4.pl* will continue executing the next *siMode*.

timeout after '*wait*´ secs

No *exposure*, *scienceReport*, *fatalMessage*, or *bepStartupMessage* packet has been received from ACIS after *wait* seconds. The current science run is terminated.

-A flag overrides command arguments

Any *siModes* specified on the command line will be replaced by a list composed of all the *siModes* in the ACIS tables that begin with CC or TE.

siMode '*tag*´ not found

The requested *siMode* is missing from the ACIS configuration table.

NOTES

This script is an improvement over *run-dat3.pl* in the following respects:

1. The **–A** flag runs *all* CC and TE configurations in the specified ACIS tables.
2. The **–B** flag lists the contents of all ACIS *siModes* and packets, and does it very efficiently.

3. Timeouts from the EU, when no exposure packets are received from *filterClient* within a given time, are trapped and reported; *run-dat4.pl* causes a warm BEP reboot and then continues with the next *siMode*.
4. Abnormal run terminations, (*i.e.*, a *fatalMessage* or *bepStartupMessage* packet) or a bad *terminationCode* value (other than 1 or 2) in a *scienceReport* packet, will cause a warm BEP reboot and continues with the next *siMode*.
5. A bug in *run-dat3..pl* was fixed that prevented the initial *commandEcho* packet of each *siMode* from being listed.
6. The **−X** flag is replaced with an internal fork so that *run-dat4.pl* can be updated while a previous version is running.

NAME

run-dat5.pl – test ACIS SImodes in software

SYNOPSIS

run-dat5.pl [**−bDv**] [**−mp** *pgm*] [**−cdS** *file*] [**−L** *dir*] [**−s** *n*] {**−A** | **−B** | *SImode* [*SImode*...]}

DESCRIPTION

This command extracts SImodes from ACIS tables and verifies the following from the contents of their command packets:

1. The SImode is defined in the ACIS configuration table
2. The SImode's packets are defined in the ACIS command packet table
3. The content of each packet can be parsed by *lcmd*.
4. The packet types are valid within an SImode.
5. The packet delays are sufficient.
6. No more than 6 CCDs have been powered up by *changeSystemConfig* commands.
7. The required CCDs have been powered up by *changeSystemConfig* commands.
8. The required FEPs have been powered up by *changeSystemConfig* commands.
9. The requested parameter block has been loaded into the correct slot.
10. If needed, a window block has been loaded into the correct slot.
11. The parameter block generates acceptable PRAM.
12. Overridden DEA microcode (+ default SRAM) is acceptable.

The *run-dat5.pl* script assumes that all CCDs and FEPs are powered down at the start of each SImode and the parameter slots are empty. Any microcode that is to be loaded by a non-zero *deaLoadOverride* field must have been initialized by *writeBep* packets within the same SImode.

To test PRAM and SRAM, the parameter block values are fed into *buildPram*, which converts them into PRAM microcode, default SRAM is appended, and the result is sent to *pchk*, which uses routines borrowed from ACIS flight software to verify that the CCDs wouldn't overheat with excessive clock transitions. If *deaLoadOverride* is non-zero, the contents of prior *writeBep* packets are assembled and converted to PRAM and SRAM microcode to be used in place of the *buildPram* output.

In verbose mode (**−v**), the SImodes and their packet names are listed on *stdout*, along with their delay times and the estimated minimum delay necessary to avoid packet loss. Without **−v,** only error messages are written. In addition, if the **−b** flag is used, warning messages are generated for the following conditions:

13. The parameter block commands that FEP source code is to be overwritten.
14. An SImode ('CC' or 'TE') is suffixed with 'B', but its parameter block doesn't force bias maps to be recomputed (*recomputeBias*=0) or to be copied to telemetry (*trickleBias*=0). This warning is only issued if the science run actually uses bias maps.
15. Conversely, a warning will be issued for any SImode without the 'B' suffix if its parameter block forces bias maps to be recomputed (*recomputeBias*=1).

If **−B** is specified, no SImodes are tested; instead, their contents are filtered through *lcmd* and written to *stdout* in ASCII format.

OPTIONS

−A Test *all* of the CC and TE SImodes in the ACIS tables specified by the **−L**, **−c**, and **−d** options. The SImodes will be examined in ascending alphabetical order.

When used in combination with **−B**, no packets are tested. Instead, **−A** restricts the output to the contents of the WC and WT parameter blocks contained within CC and TE configurations.

−B Instead of testing packets, list the contents of all SImodes in the ACIS table files. When used in combination with **−A**, only WC and WT packets are listed.

−b Check the consistency of the *recomputeBias* and *trickleBias* fields in parameter blocks that use bias maps, reporting if they are inconsistent with the presence or absense of a 'B' sufffix to the SImode name.

−c *cfg* The name of the ACIS configuration table file. The default is *“current.cfg”*.

−D List the contents of the command packets while testing them. Use together with **−A**, to list and test all SImodes.

−d *dat* The name of the ACIS packet table file. The default is *“current.dat”*.

−L *dir* Look for *cfg* and *dat* files in the *dir* directory. The default is *“/nfs/acis/h1/www/bin”*.

−m *pgm* Use *pgm* to generate PRAM from the contents of the parameter block. The default is *“buildPram”*.

−p *pgm* Use *pgm* to check the PRAM generated by the contents of the parameter block. The default is *“pchk”*.

−S *sram* Use the contents of *sram* to supply the default SRAM when validating the parameter block. The default is *“$ACISTOOLSDIR/lib/defaultSram.dea”*.

−s *n* When the **−A** and/or **−B** flags are used, skip the first *‘n’* SImodes. The default is not to skip any.

−v Be verbose: report the number of records read from the ACIS tables, and list the packet IDs and delays as they are being tested.

FILES

/nfs/acis/h1/www/bin/current.cfg ACIS configuration table.

/nfs/acis/h1/www/bin/current.dat ACIS command packet table.

$ACISTOOLSDIR/*lib/defaultSram.dea* Default SRAM contents.

ENVIRONMENT

ACISTOOLSDIR Location of ACIS EGSE tools

AUTHOR

Peter G. Ford, MIT Kavli Institute <pgf@space.mit.edu>

SEE ALSO

buildPram(1), lcmd(1), pchk(1), run-dat3.pl(1), run-dat4.pl(1),

DIAGNOSTICS

−A flag overrides command arguments

Any SImodes specified on the command line will be replaced by a list composed of all the SImodes in the ACIS tables that begin with CC or TE.

SImode *‘name’* **not found**

The named SImode is missing from the ACIS configuration table.

When not in verbose mode, *i.e.*, when **−v** is not specified, the SImode will be prefixed to the packet name in the following error messages.

BEP load address *addr* **empty**

No *writeBep* command has preceded execution of a parameter block whose *deaLoadOverride* value is *addr*. Unless the **−v** flag is used, this message will be suppressed for ‘TN’ SImodes.

BEP load address *addr1* **not** *addr2*

The first *writeBep* command of this SImode contained a *writeAddress* of *addr2*, but the value of the *deaLoadOverride* field in the parameter block was *addr1*.

CCD *n* **not powered up**

According to the *changeConfigSetting* packet(s), CCD_*n* is not powered-up for this SImode.

command: message

An error *message* resulted from executing *command* on the PRAM generated by the parameter block, or if *deaLoadOverride* is non-zero, on the PRAM and non-default SRAM supplied by *writeBep* packets.

FEP *n* **not powered up**

According to the *changeConfigSetting* packet(s), FEP_*n* was not powered-up for this SImode.

insufficient delay *n* **secs (<** *n* **secs)**

The specified delay to follow execution of this command packet appears to be insufficient. Unless the **–v** flag is used, no such error message will be generated for the last packet in each SImode. *run-dat5.pl* estimates minimum delays based on command type, *i.e.*, 4 seconds for all types except 23 seconds for *dumpHuffman* and a content-dependent delay for *changeConfigSetting*. The latter is assumed to need 1 second to power up each CCD and 8 seconds to power up each FEP, but the total delay must be no less that 4 seconds.

more than 6 CCDs powered up

Running more than 6 CCDs simultaneously can damage the ACIS power supply.

no bias recomputed for B-type SImode

This message is produced when the **–b** flag is specified, the SImode ends in the letter B, and the run uses FEP bias maps, but the *recomputeBias* field in the parameter blocks has zero value, indicating that a bias is not to be recomputed if an uncorrupted bias map still exists in its FEP.

no bias trickled for B-type SImode

This message is produced when the **–b** flag is specified, the SImode ends in the letter B, and the run uses FEP bias maps, but the *trickleBias* field in the parameter blocks has zero value, indicating that if a bias is recomputed, it will not be reported in telemetry.

bias recomputed for non-B-type SImode

This message is produced when the **–b** flag is specified, the SImode does not end in the letter B, and the run uses FEP bias maps, but the *recomputeBias* field in the parameter blocks has value 1, indicating that bias maps will always be recomputed.

no pblock loaded into slot *n*

A *startScience* command is executed for slot *n*, which was not filled with a prior *loadCcBlock* or *loadTeBlock* command during this SImode.

no window block loaded into slot *n*

A *startScience* command is executed with a parameter block that calls for a window block in slot *n*, but no prior *load1dBlock* or *load2dBlock* command had loaded the slot during this SImode.

non-contiguous BEP memory: *addr1* **–** *addr2*

There is a gap between the last address to be loaded by a *writeBep* command and the *writeAddress* of the next *writeBep*.

packet missing

The packet was not found in the command packet table.

unsupported packet type: *type*

The packet contains a *type* command that is not usually found in SImode configuration table entries, *i.e., changeConfigSetting*, *dumpHuffman*, *dumpSysConfig*, *executeBep*, *load1dBlock*, *load2dBlock*, *loadCcBlock*, *loadTeBlock*, *startScience*, *stopScience*, and *writeBep*.

warning: FEP overridden

The value of the *deaLoadOverride* field in the parameter block of this SImode is non-zero.

NOTES

1. This script is adapted from *"run-dat4.pl"*. All output is directed to *stdout*, not to *stderr*.
2. The **–B** flag lists the contents of all ACIS SImodes and packets, and does it more efficiently than the combination of **–A** and **–D** flags, although the latter tests the SImodes at the same time.
3. *executeBep* packets are not tested. Non-zero *fepLoadOverride* fields are reported but not tested.