NAME

acisCtl – user interface to the ACIS instrument

SYNOPSIS

acisCtl –e [**–P** *phost*] [**–d** *dir*] [**–h** *host*] [**–p** *file*] [**–v**]

acisCtl –f [**–F** *port*] [**–d** *dir*] [**–h** *host*] [**–p** *file*] [**–v**]

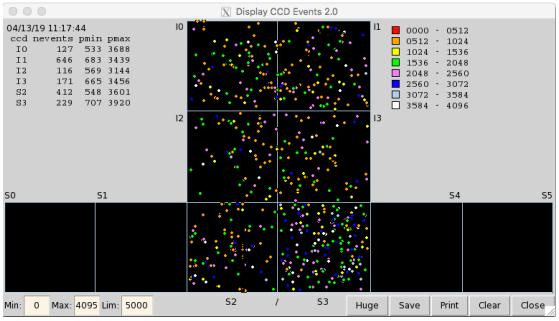

Figure 1. Display of event locations & statistics

DESCRIPTION

The *acisCtl* command controls access to ACIS telemetry. When invoked with the **–e** flag, it can also send commands to the instrument. If a second interface is available through the **–P** option, *acisCtl* can also send command to, and receive status from, the PSMC power unit. The main menu is shown in Fig. 2a (**–e** mode), Fig. 2b (**–e** mode with **–P**) and Fig. 3 (**–f** mode) on Page 2.

OPTIONS

–e *acisCtl* sends commands to the ACIS engineering unit and receive ACIS telemetry packets in return. The interface can be remote: see the description of **–h**, below.

–f *acisCtl* waits for TCP connections, *e.g.*, from a COG server, through which it receives 1029-byte Chandra telemetry records (SFDUs) containing science and engineering data.

–F *port* specifies the TCP port to be used by *filterServer* to accept data connections.

–P *phost* indicates that the PSMC power unit can be accessed via a telemetry server on *phost*. This option is ignored in **–f** mode.

–d *dir* specifies the location of ACIS binaries, *i.e.*, replaces the `$ACISTOOLSDIR` environment variable within *acisCtl* and its associated procedures.

–h *host* if **–e** is specified, **–h** indicates that the engineering unit interface (usually *shim*) is running on *host*; if **–f** is specified, it is ignored since the COG interface must run on the local host.

–p *file* specifies a startup file to be sourced to supply *acisCtl* with environment variables; if this option is omitted, *acisCtl* first sources "`$ACISTOOLSDIR`*/lib/acisegse.parms*", and then "*~/.acisctlrc"* if it exists. These files must contain valid *sh* commands. Note that the "Save" button of the "*acisCtl* Options" dialog (see Figure 17 and the "Environment" section, below) saves current parameters in *file*, if specified, or in "*~/.acisctlrc*" if not.

–v reports the name of the startup file and the directory containing *acisCtl* scripts and maps.

–w *file* the name of the TCL interpreter; if omitted, *acisCtl* uses the *wish* executable from the same directory as *acisCtl* itself. If this is missing, it finds "*wish*" in the user's `$PATH`.

PRIMARY MENU

The content of this menu depends on whether *acisCtl* was started with **–e** or **–f**. In **–e** mode, menu items that are used to command the ACIS engineering unit are displayed with a grey background. Those that are used to monitor telemetry have a white background. The **–e** mode menu is shown in Fig. 2a, the start of the **–e** mode with **–P** in Fig. 2b, and the **–f** mode menu in Fig. 3.

Most of these commands will not work unless the ACIS interface has been started (via the "Start I/O Server..." item, described in detail in the following section.) "Critical" commands to the ACIS PSMC will be sent by `$DANGER_CMD`; all other by `$EXECUTE_CMD`. Forked processes started from the above

Figure 2a. The main *acisCtl* menu in **–e** mode

menu expect to read ACIS telemetry packets from port 7002 of *localhost*, except for *psmc.tcl* when **–P** is specified, when it reads from port 7001 of `$PSMC_SERVER`. The "Print" button beneath the main menu permits you to print any of the *acisCtl* windows, the "Close" button closes that particular window, but *acisCtl* continues. "Quit" terminates *acisCtl*, cleaning up all background processes and killing all windows that it has created.

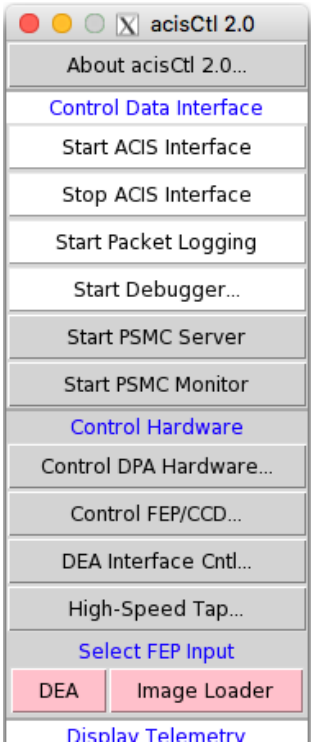

Figure 2b. The top sections of the main *acisCtl* menu in **–e** mode with **–P** specified

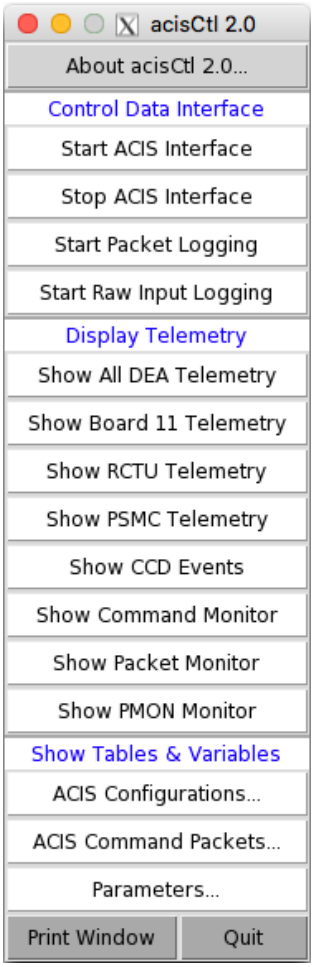

Figure 3. The main *acisCtl* menu in **–f** mode

Menu Item	Description
About acisCtl 2.0	Displays a window containing the *acisCtl* manual in color with buttons and scroll bars letting the user move from page to page.
	Control Data Interface
Start ACIS Interface	Start `$RCTU_CMD` on the host named `$DATAHOST`, reading its *stdout* in a dedicated window. In **–f** mode, the default value of `$RCTU_CMD` is *acisTstShim;* in **–e** mode, it is *acisEUshim*. The former starts the *tlmGet* process on the local host to listen for incoming TCP connections, *e.g.*, from a COG server; the latter starts *cserver* and *filterServer* on the local machine, piping *cserver* output to *shim* on `$DATAHOST` and *shim* output to *filterServer*. The *stderr* from all processes is displayed in a scrolling window, with error messages colored in red.
Stop ACIS Interface	Stop the currently running ACIS interface, which has the effect of closing all processes currently reading telemetry from *filterServer*.
Start Packet Logging	Command *pktCpy* to begin logging ACIS packet input. Log files are written to `$TLM_LOG_FILE` in the `$TLM_LOG_DIR` directory. If `$TLM_LOG_FILE` contains "%" characters, it will be expanded by *strftime*(3) prior to opening, so that date and time substitution can be performed. If `$TLM_LOG_FILE` ends in "*.gz*", the log file will be piped through *gzip*. Once logging starts, this button will be colored pink and read "Stop Packet Logging". When clicked, it closes the log file and reverts to its original appearance. When the uncompressed size of a log file exceeds the value of `$LOGMAXBYTES`, *pktCpy* will close the file and begin a new one, substituting "%" fields in the file name with the current date and time. Similarly, while logging is in progress, *pktCpy* will start a new log file if more than `$LOGWAITSECS` elapses without any new data.
Start Raw Input Logging	(**–f** mode only) Command the I/O server *tlmGet* to begin logging telemetry input. Log files are written to `$RCTU_DUMP_FILE` in the `$RCTU_DUMP_DIR` directory. If `$RCTU_DUMP_FILE` contains "%" characters, it will be expanded by *strftime*(3) prior to opening, so that date/time substitution can be performed. If `$RCTU_LOG_FILE` ends in "*.gz*", the log file will be piped through *gzip*. Once logging starts, this button will be colored pink and read "Stop Minor Frame Logging". When clicked, it closes the log file and reverts to its original appearance. When the uncompressed size of a log file exceeds the value of `$LOGMAXBYTES`, *tlmGet* will close the file and begin a new one, substituting "%" fields in the file name with the current date and time. Similarly, while logging is in progress, *tlmGet* will start a new log file if more than `$LOGWAITSECS` elapses without any new data.
Start Debugger...	(**–e** only) Display a dialog (see Figures 13a and 13b) to interact with the ACIS engineering unit, sending *readBep* and *readFep* commands and displaying the results in a variety of formats, including assembler listings identifying global locations via BEP and FEP load maps.

Menu Item	Description
Start PSMC Server	(**–e** mode with **–P** only) Start the interface to the PSMC by starting *psmcDown* in `$DISP_TERM` window (usually *xterm* or *rxvt*). This interface has not been used since pre-launch thermal vacuum tests. It requires a second L-RCTU interface specified by the **–P** option of *acisCtl*.
Start PSMC Monitor	(**–e** mode with **–P** only) Display PSMC telemetry from the *psmcDown* process, piping it through *DoPSMC* in a `$DISP_TERM` window. It requires a second L-RCTU interface specified by the **–P** option of *acisCtl*.
Control Hardware (–e mode only)	
Control DPA Hardware...	(**–e** only) Display a dialog (see Figure 4) to send commands to the engineering unit DPA via its serial hardware interface.
Control FEP_CCD...	(**–e** only) Display a dialog (see Figure 5) that indicates the power status of DEA and FEP boards, and powers them on and off.
DEA Interface Cntl...	(**–e** only) Display a dialog (see Figure 6) that controls various functions of the DEA, *i.e.*, relay settings, bake-out heaters, etc.
High-Speed Tap...	(**–e** mode with **–P**) Display a dialog (see Figure 8) to control the high-speed tap interface to ACIS.
Select FEP Input (–e mode only)	
DEA	(**–e** mode only) Command the Pixel Switch to receive FEP input from the DEA. This button will be colored red before selection; green if selected, white otherwise.
Image Loader	(**–e** mode only) Command the Pixel Switch to receive FEP input from the Image Loader. This button will be colored red before selection; green if selected, white otherwise.
Display Telemetry	
Show All DEA Telemetry	Display the DEA telemetry currently being received by the I/O server. (See Figure 15). If only interface board channels are needed, use the "Show Board 11 Telemetry" item instead.
Show Board 11 Telemetry	Display the DEA interface board (board 11 or 12) telemetry currently being received by the I/O server. This display is identical in layout and contents to the right-hand section of Figure 15.
Show RCTU Telemetry	Display RCTU channel telemetry currently being received by the I/O server. See Figure 16.
Show PSMC Telemetry	In **–f** mode, display PSMC status (see Figure 16) from telemetry currently being received by the I/O server; in **–e** mode with **–P,** as for **–f** mode, but also send commands to the PSMC.
Show CCD Events	Display the location and statistics of events currently being received by the I/O server. See Figure 1.
Show Command Monitor	(**–e** only) Display a scrolling window showing commands being sent to the I/O server and a brief description of the responses. Rejected commands and responses that report errors will be colored red.

Menu Item	Description
Show Packet Monitor	Display a scrolling window showing science packets currently being received by the I/O server, formatted by "*psci –m*". Critical packets, *i.e.*, *fatalMessage, bepStartupMessage*, will be colored red.
Show PMON Monitor	Display information about the science and housekeeping data currently being received by the I/O server, as formatted by *pmon*.
Send Commands	
Command Loader...	(**–e** only) Display a dialog (see Figure 11) of ACIS command and packet files, *i.e.*, **.bcmd* and **.pkts*, from the *acisCtl* library, with the options of listing or editing their contents, and sending them to the software serial port of the ACIS engineering unit.
BEP Dump/Reset...	(**–e** only) Display a dialog (see Figure 7) that commands ACIS to dump the contents of various BEP memory structures in science telemetry, or to clear the data contents of these structures.
Timed Exposures...	(**–e** only) Display a dialog (see Figure 9) to list, edit or load timed-exposure parameter blocks stored in the *acisCtl* library; also start or stop a science run, or compute bias maps.
2-D Windows...	(**–e** only) Display a dialog to list, edit or load 2-dimensional CCD window parameter blocks stored in the *acisCtl* library. The dialog is similar to Figure 9.
Continuous Clocking...	(**–e** only) Display a dialog to list, edit or load continuous-clocking parameter blocks stored in the *acisCtl* library; also start or stop a science run, or compute bias maps. The dialog is similar to Figure 9.
1-D Windows...	(**–e** only) Display a dialog to list, edit or load 1-dimensional CCD window parameter blocks stored in the *acisCtl* library. The dialog is similar to Figure 9.
DEA Housekeeping...	(**–e** only) Display a dialog to list, edit or load DEA housekeeping parameter blocks stored in the *acisCtl* library. The dialog is similar to Figure 9.
Image Loader...	(**–e** only) Display a dialog (see Figure 10) to list or edit image definition files from the *acisCtl* library; optionally, process them though *genObjectImage* and copy them to the ACIS image loader.
Show Tables & Variables	
ACIS Configurations...	Display a dialog (see Figure 12) of the contents of an ACIS configuration database, convert the items into command packets, and either list them or send them to ACIS with appropriate delays.
ACIS Command Packets...	Display a dialog of the contents of an ACIS packet database and either list them or send them to ACIS. The dialog is similar to Figure 12.
Parameters...	Display a dialog (see Figure 17) that lists *acisCtl* parameters and permits them to be saved and updated during program execution.
Print Window	Click in any X11 window to print the contents of that window to the device specified by the `$PRINTER` environment variable.
Quit	Exit from *acisCtl*, stopping all displays and telemetry logs, and closing all *acisCtl* windows.

CONTROL DPA HARDWARE DIALOG

This consists of a series of selectors and "radio" buttons (see Figure 4). Whenever one is selected, an appropriate command will be sent to the PSMC or DPA via `$EXECUTE_COMMAND`:

Command	Mnemonic	Description
DPA Side-A	`1DP1AEN`	Enable DPA A
	`1DPPSAON`	Power-on DPA A
DPA Side-B	`1DP1BEN`	Enable DPA B
	`1DPPSBON`	Power-on DPA B
DEA Side-A	`1DEPSBDS`	Disable DEA B
	`1DEPSAEN`	Enable DEA A
	`1DEPSAON`	Power-on DEA A
DEA Side-B	`1DEPSADS`	Disable DEA A
	`1DEPSBEN`	Enable DEA B
	`1DEPSBON`	Power-on DEA B
DEA Off	`1DEPSADS`	Disable DEA A
	`1DEPSBDS`	Disable DEA B
Cold Boot	`1BMODIBM 0`	DPA Boot Modifier Off
	`1WRMBTSB 0`	DPA Warm Boot Off
	`1RSETIRT 1`	DPA Halt
	`1RSETIRT 0`	DPA Run
Warm Boot	`1BMODIBM 0`	DPA Boot Modifier Off
	`1WRMBTSB 1`	DPA Warm Boot On
	`1RSETIRT 1`	DPA Halt
	`1RSETIRT 0`	DPA Run
Select BEP Side-A	`1BSELICL 0`	BEP Select Side A
Select BEP Side-B	`1BSELICL 1`	BEP Select Side B
Run BEP	`1RSETIRC 0`	DPA Run
Halt BEP	`1RSETIRC 1`	DPA Halt
Radiation Lo	`1RMONIRM 0`	DPA Radiation Monitor Low
Radiation Hi	`1RMONIRM 1`	DPA Radiation Monitor High
UplinkBoot Off	`1BMODIBM 0`	DPA Boot Modifier Off
UplinkBoot ON	`1BMODIBM 1`	DPA Boot Modifier On

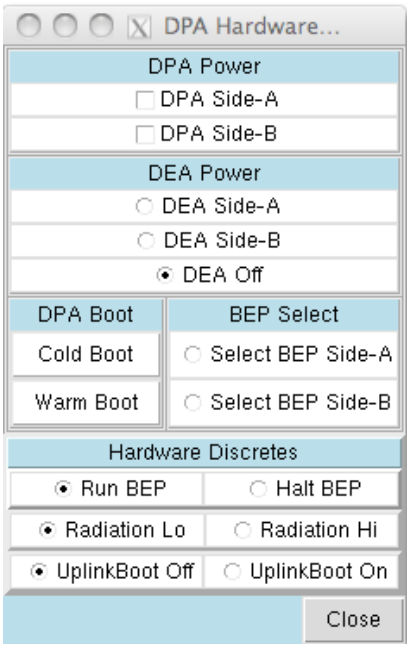

Figure 4. The DPA hardware dialog (**–e** mode only)

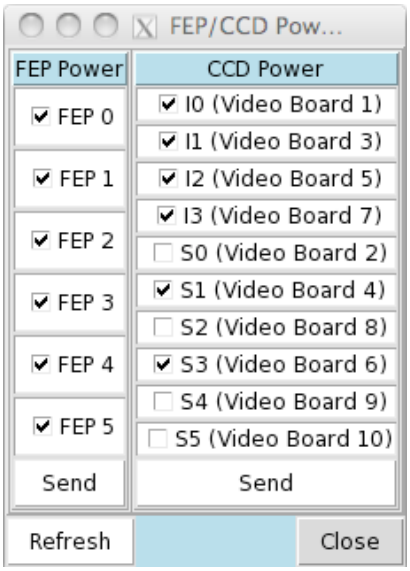

Figure 5. The FEP/CCD Power Dialog (**–e** mode only)

CONTROL FEP/CCD POWER DIALOG

This dialog (see Figure 5) lets the user select which combination of FEP and CCD boards should be powered up. No *changeConfigSetting* command will be sent until the appropriate "Send" button is clicked. If ACIS can receive commands via `$EXECUTE_CMD`, the "refresh" button will send the BEP a *dumpSysConfig* command, receive the reply, and update the selection boxes accordingly.

CONTROL DEA INTERFACE DIALOG

This dialog (see Figure 6) controls miscellaneous features of the DPA/DEA Interface. With the exception of the "Bakeout Enable" button, which is passed immediately to `$DANGER_CMD`, changes to the buttons and data entry fields will not take effect

until the “Send” button is clicked, at which time a single multi-entry *changeConfigSetting* command will be sent to ACIS.

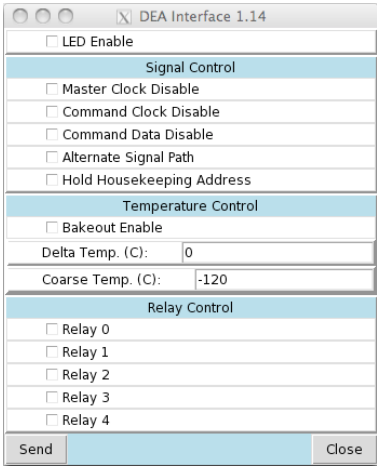

Figure 6. DEA Interface Dialog (**–e** mode only)

BEP DUMP/RESET DIALOG

This dialog (see Figure 7) sends commands to ACIS to dump or clear the contents of specific parameter blocks. When the “Display” box is selected at the bottom of the window, the contents of a dumped region will also be converted to ASCII and displayed in a scrolling window. The items to be displayed can be selected by block index (0–4), *ccdId* (0–9) or *fepID* (0-5): enter the number (or range of numbers separated by a dash) in the box to the right of “Item Range”. Note that items in the Patch List are selected by *patchId* value.

Menu	Range	Regions Affected
Dump Huffman Tables	0–4	Tables used to compress CCD bias maps and raw frames.
Dump System Configuration	0–9	The BEP system configuration table describing video boards and DEA and DPA power.
Dump Patch List	0–n	Flight software patches currently stored in BEP I-Cache.
Dump Bad Pixel Map	0–9	Bad CCD pixels, listed by ccdId, row and column.
Dump Timed Exposure Blocks	0–4	Parameter blocks used in timed-exposure mode.
Dump 2-D Window Blocks	0–4	2-dimensional window blocks used in timed exposure mode.
Dump Bad TE Column Map	0–9	Bad CCD columns in timed-exposure mode, listed by ccdId and column.
Dump Contin. Clocking Blocks	0–4	Parameter blocks used in continuous clocking mode.
Dump 1-D Window Blocks	0–4	1-dimensional window blocks used in continuous clocking mode.
Dump Bad CC Column Map	0–9	Bad CCD columns in continuous clocking mode, listed by ccdId and column.
Dump DEA Housekeeping Blocks	0–4	Parameter blocks used to report DEA housekeeping channels.
Dump fepCtl D-cache Structure	0–5	The *fepCtl* structure in the execution stack (in D-cache) of a specified FEP.
Dump FepBiasParity2 Structure	0–5	The status block maintained by the *fepbiasparity2* patch.

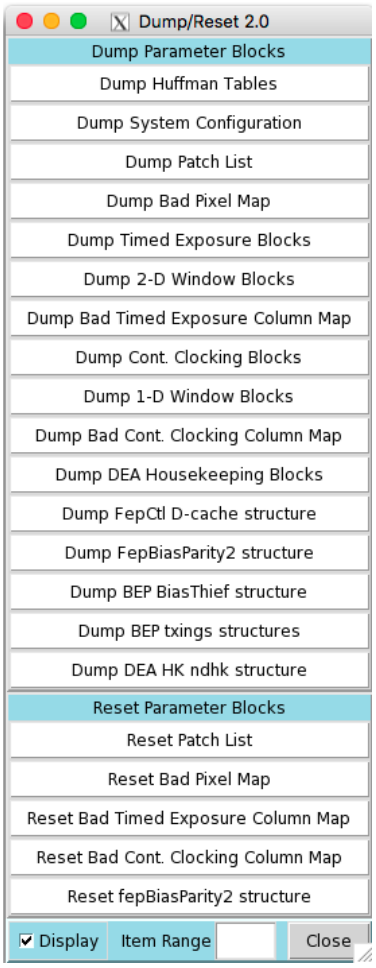

Figure 7. Dialog to command the BEP to dump or reset its control blocks (**–e** mode only)

Menu	Range	Regions Affected
Dump BEP Bias-Thief structure	N/A	The *biasthief* object
Dump BEP txings structures	N/A	The *TXings*, *TXinit* and *TXnext* structures
Dump DEA HK ndhk structure	N/A	The *ndhk* structure created by the *deahktrip* patch
Reset Patch List	all	Flight software patches currently stored in BEP I-Cache.
Reset Bad Pixel Map	all	Bad CCD pixels.
Reset Bad TE Column Map	all	Bad CCD columns in timed-exposure mode.
Reset Bad CC Column Map	all	Bad CCD columns in continuous clocking mode.
Reset FepBiasParity2 Structure	0–5	The status block maintained by the *fepbiasparity2* patch.

HIGH-SPEED TAP DIALOG

This dialog was intended for use when the ACIS instrument was under test. It activates the "high-speed tap" feature in a designated video board, whose pixel output will be collected by additional hardware. It is only displayed in **–e** mode when the **–P** option is specified. See Figure 8.

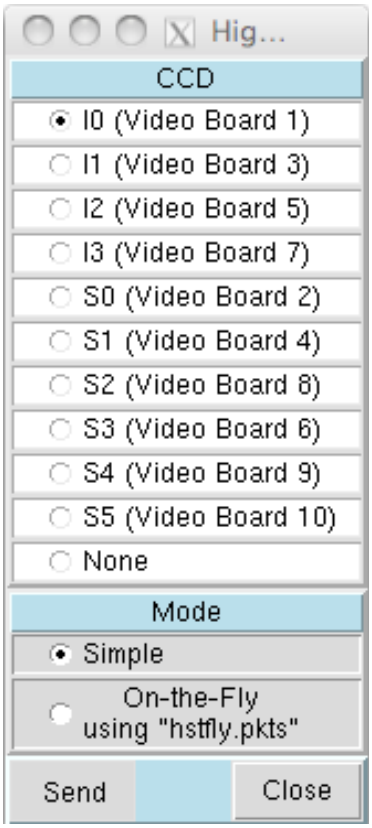

Figure 8. High-Speed Tap Dialog (**–e** mode with **–P**)

COMMAND SELECTION DIALOGS

These dialogs (see Figures 9–12) select files, or components of files, and send their contents to ACIS, either as *bcmd* input to be read by `$EXECUTE_CMD`, or as image pixel definitions to be sent to *genObjectImage*, or as binary command packets to send to `$LOAD_RAW_COMMAND`.

Menu Item	Extension	Default Directory	Description
Command Loads	**.bcmd* **.pkts*	`$RAW_CMD_LIB`	Command or raw packet files
BEP Dump/Reset			See above
Timed Exposures	**.te*	`$PARAM_BLOCK_LIB`	TE-Mode Parameter Blocks
2-D Windows	**.2d*	`$PARAM_BLOCK_LIB`	TE-Mode Window Blocks
Continuous Clocking	**.cc*	`$PARAM_BLOCK_LIB`	CC-Mode Parameter Blocks
1-D Windows	**.1d*	`$PARAM_BLOCK_LIB`	CC-Mode Window Blocks
DEA Housekeeping	**.dea*	`$PARAM_BLOCK_LIB`	DEA Housekeeping Blocks
Image Loader	**.img*	`$IMAGE_LIB`	Image Loader Input Files
ACIS Configurations	**.cfg*	`$ACIS_CFGS`	ACIS Configurations
ACIS Command Packets	**.dat*	`$ACIS_PBLKS`	ACIS Uplink Commands

In the table above, the items to be displayed are either files with a particular extension in the directory specified by an environment variable, or for the last two items, they are individual elements in the files specified by `$ACIS_CFGS` or `$ACIS_PBLKS`. Buttons at the bottom of each dialog allow the user to apply various functions to each selected file or element, *e.g.*, to list their contents in a separate window, to edit their contents (using `$EDITOR`), to load the file or element into ACIS via `$EXECUTE_CMD` (for ASCII files) or `$LOAD_RAW_CMD` (for binary files or elements), or *genPixelImage* for image files to send to the image loader.

Some dialogs have additional buttons and entry fields for special purposes: parameter blocks may be loaded into, and executed from, a choice of slots from 0 through 4; science runs may be started, stopped, or run only to generate bias maps. The "Software Patches" dialog includes a "Stop" button that kills the dialog window but lets the loading continue, and a "Cancel" button that commands the loading task to stop and waits until it has done so.

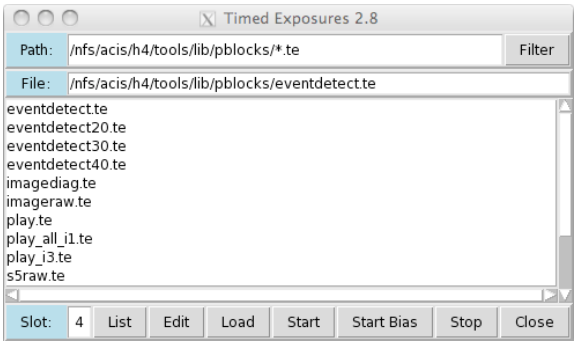

Figure 9. Dialog to list, edit or upload a timed-exposure parameter block into a specified memory slot; also to create bias maps or execute a science run or. The other parameter block dialogs (CC, window, and DEA housekeeping) are similar. (**–e** mode only)

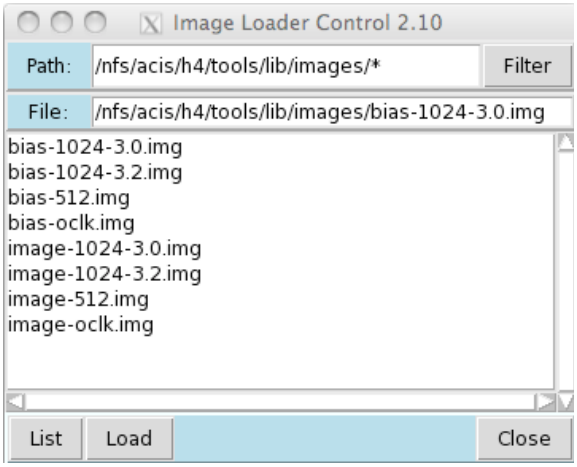

Figure 10. Dialog to list or load a pixel image in *genObjectImage* format. (**–e** mode only)

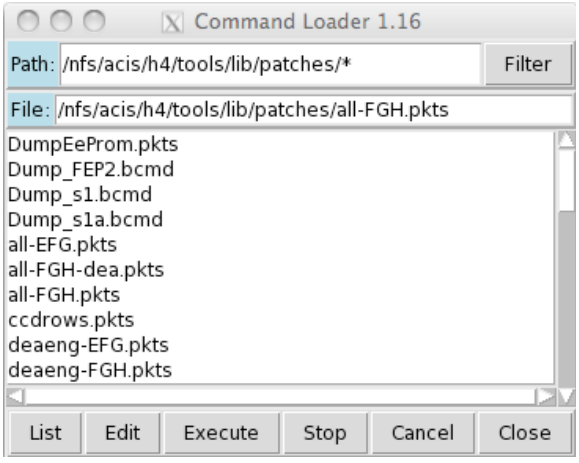

Figure 11. Dialog to list, edit or execute a command script in *bcmd* format; or to list or execute a binary command file. (**–e** mode only)

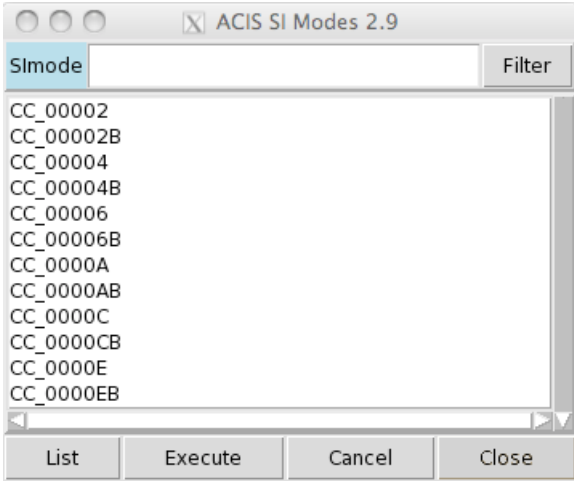

Figure 12. Dialog to list or execute an ACIS configuration (timed sequence of commands for starting a science run (**–e** mode only)

THE DEBUG DIALOG

This window can only be displayed and used in **–e** mode with the **–D** flag, *i.e.*, when *acisCtl* is able to directly command the ACIS engineering unit. It is invoked from the "Start Debugger..." button of the main menu. An example is shown in Figure 13a. Before displaying the window, *acisCtl* reads two storage maps: the BEP map from `$BEP_MAP` and the FEP map from `$FEP_MAP`. The default values are respectively "*acisBep.map*" and "*acisFepSci.map*" and are located in "`$ACISTOOLSDIR`/*lib/acisctl*". These maps describe the software installed in the flight instrument's EEPROMs. To describe the load maps after patches have been applied, these should be replaced by the appropriate maps, *e.g.,* "*$ACISFS/patchbld/release-*-opt-*/release/dist/*-release-*-*/*.map*". Since the maps are generated incrementally as each patch is linked, the most useful for debugging purposes is the last linked, which thus far ("*release-F-opt-G*") has been "*opt_dearepl.map*".

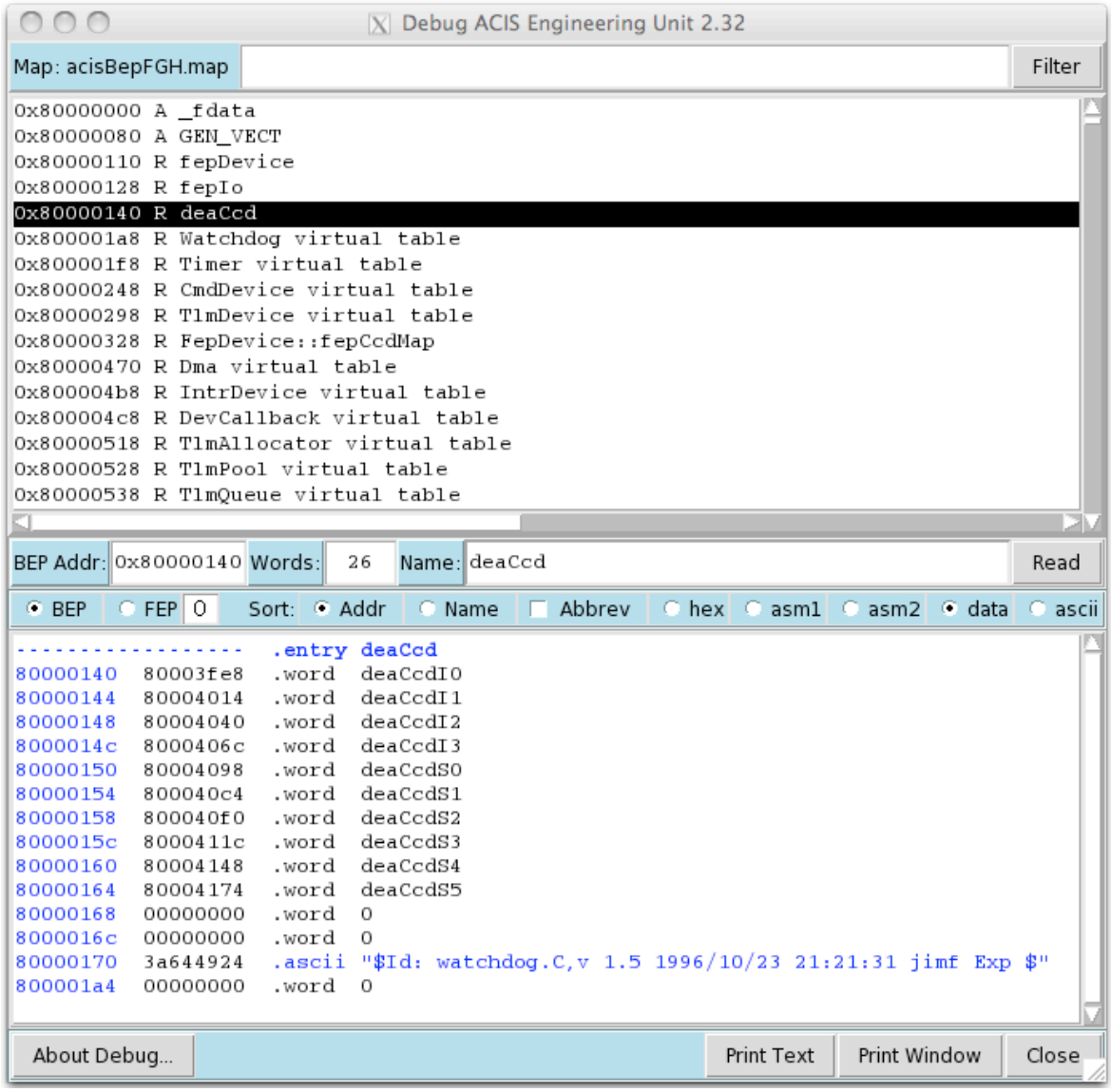

Figure 13a. The Debug Dialog; the output format is "data", so the contents of the "*deaCcd*" block are interpreted as either global addresses, ASCII character strings, or 32-bit binary integers. Refer to the text for a description of the other boxes and buttons.

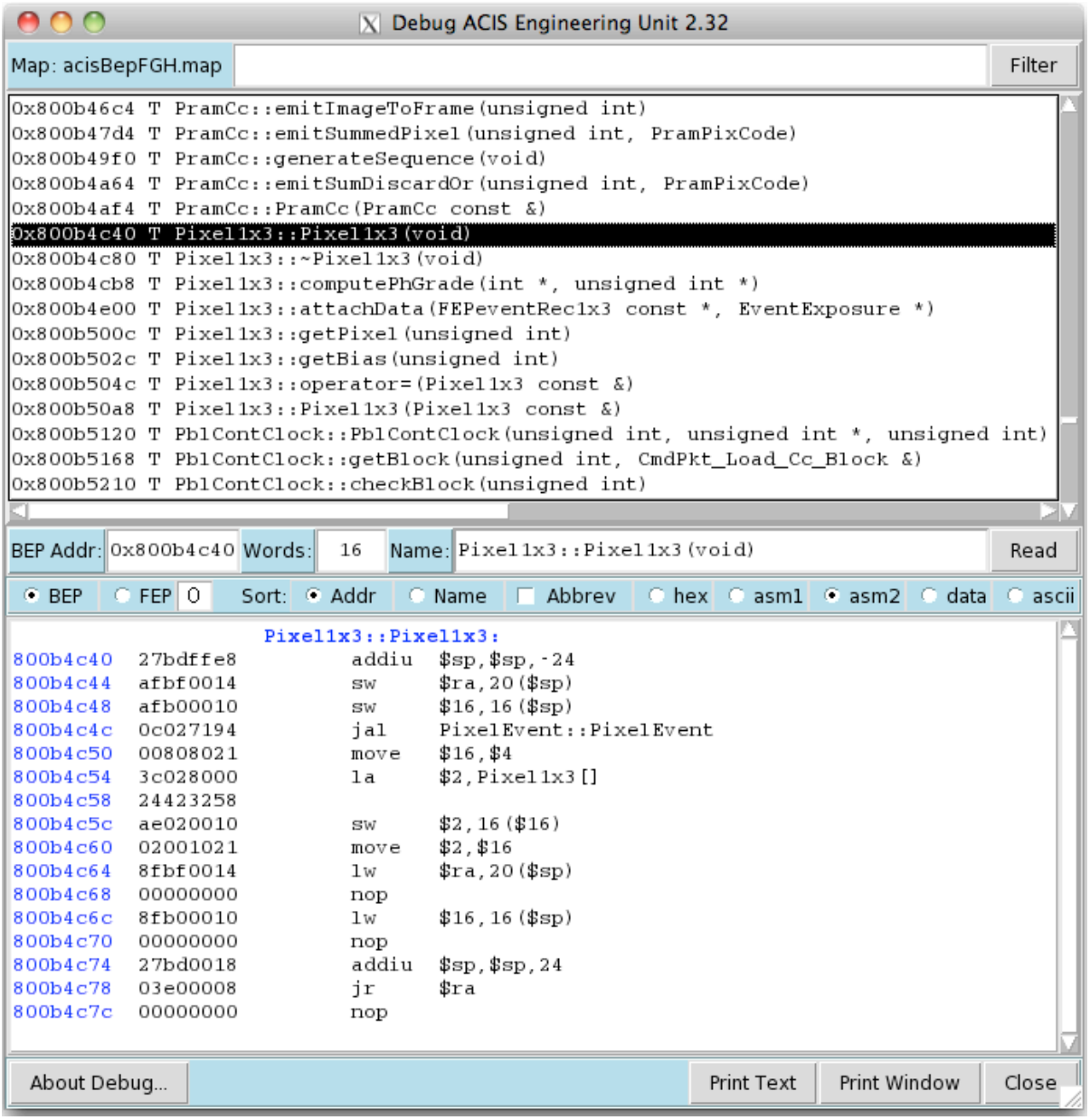

Figure 13b. The Debug Dialog; the output format is "asm2", so the two instruction to load the `Pixel1x3` address are interpreted as a single "la" macro. Refer to the text for a description of the other boxes and buttons.

The dialog starts by displaying the BEP map in a scrolling list, sorted by ascending address. Click the "FEP" button to switch to the FEP map, and type the number of the FEP that you want to debug into the box at its right. To sort by symbolic name, click the "Name" button. Select "Abbreviate" to list "virtual table" as "[]", and subroutine arguments as "()". Select a subset of names by typing an expression into the window to the left of the "Filter" button and hitting the RETURN key or clicking the "Filter" button. You can use the usual wildcards in your search, but parentheses are not allowed. Click on any item in the scrolling list and two things will happen: the address, length (in 32-bit words) and symbolic name will appear in the line below the scrolling list, and a "*readBep*" or "*readFep*" command will be sent to the engineering unit. *acisCtl* will wait until a reply is received, which will be displayed in the result area in the selected format:

hex display each 32-bit word as a hexadecimal string with `0x` prefix

asm1 treat each word as an R3000 instruction and list in assembler format, resolving global addresses through the BEP and FEP load maps

asm2 as *asm1*, but merge certain 2-word commands, *e.g.*, "la", into assembler "macros"

data where appropriate, display each 32-bit word as a global address or ASCII character string; otherwise, display in hexadecimal

ascii where appropriate, display the memory block as a series of ASCII character strings; otherwise, display blanks

Each line of the display is prefixed by the address of the first (or only) word described on that line. The result area can be extended by resizing the window, but there is a limit of 1000 words that can be requested. The contents of addresses that are not in the maps can be displayed by entering their hexadecimal address in the "Addr:" box and then hitting the `RETURN` key (with the inserting point still in the address box) or by clicking the "Read" button. Symbolic address will be displayed as a non-negative byte offset from nearest known address, *e.g.*, with "*FepDevice::fepCcdMap*" at `0x80000328` (see Figure 13), requesting `0x80000330` will display "*FepDevice:: fepCcdMap*+8" in the "Name:" box. The "Words:" box is similar to the "Addr:" box: to alter the length of the region to be examined, retype the value and either hit `RETURN` or click the "Read" button.

TELEMETRY DISPLAY WINDOWS

ACIS output is converted to a stream of varying-length packets and sent to the *filter-Server* command, which accepts TCP connections from *filterClient* commands. Only rarely does *acisCtl* start a *filterClient* directly because this might cause all its displays to hang. Instead, they are run as separate *wish* processes (the name of the *tcl* interpreter), inheriting *acisCtl* environment variables. The following figures illustrate the major engineering displays: the DEA housekeeping channels reported in BEP science packets and some of the more important engineering channels reported by RCTU #12.

A third window displays the location of events in the ACIS CCDs. An example is shown at the top of Page 1. The events are written in 8 colors according to the data range specified in the "Min" and "Max" boxes and listed in the top right of the window. Statistics are listed in the top left. The "Clear" button resets the event displays and restarts the statistics. The button marked "Huge" in the example will cycle between "Small", "Large" and "Huge" to specify the size of the points are they are drawn.

The PSMC status has its own display, shown in Figure 14. In **–f** mode, it displays engineering

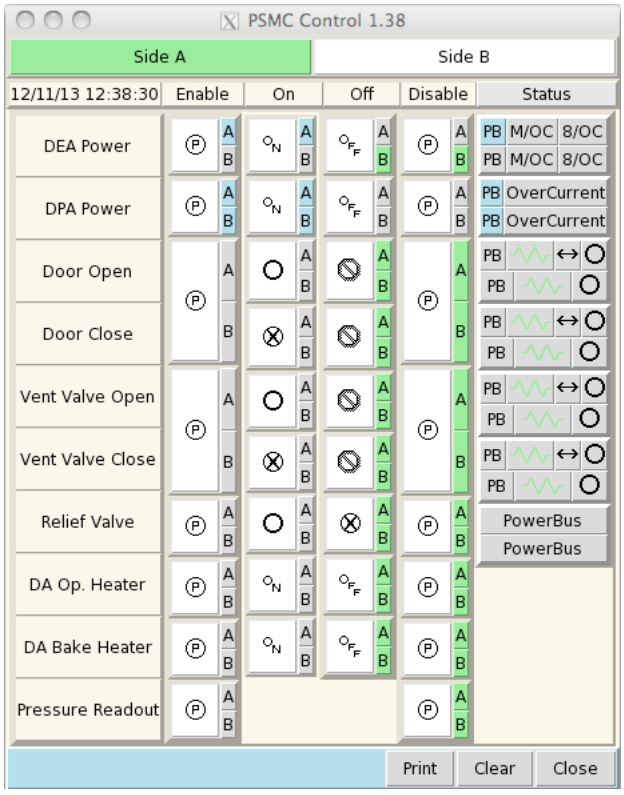

Figure 14. Display of selected ACIS PSMC channels. In **–f** mode, the A/B buttons will be colored to correspond to the status reported in downlink telemetry. In **–e** mode, only the top line (Side A/B) will be updated unless **–P** is specified, in which case clicking the A/B buttons will send commands to the PSMC and will reflect its current status.

telemetry relating to the power supply and its ancillary functions. It is not available in **–e** mode unless the **–P** option is also used to point to the location of a PSMC interface, at which time the display becomes an interactive dialog: clicking on the various buttons will cause the appropriate commands to be sent to the PSMC and their result displayed.

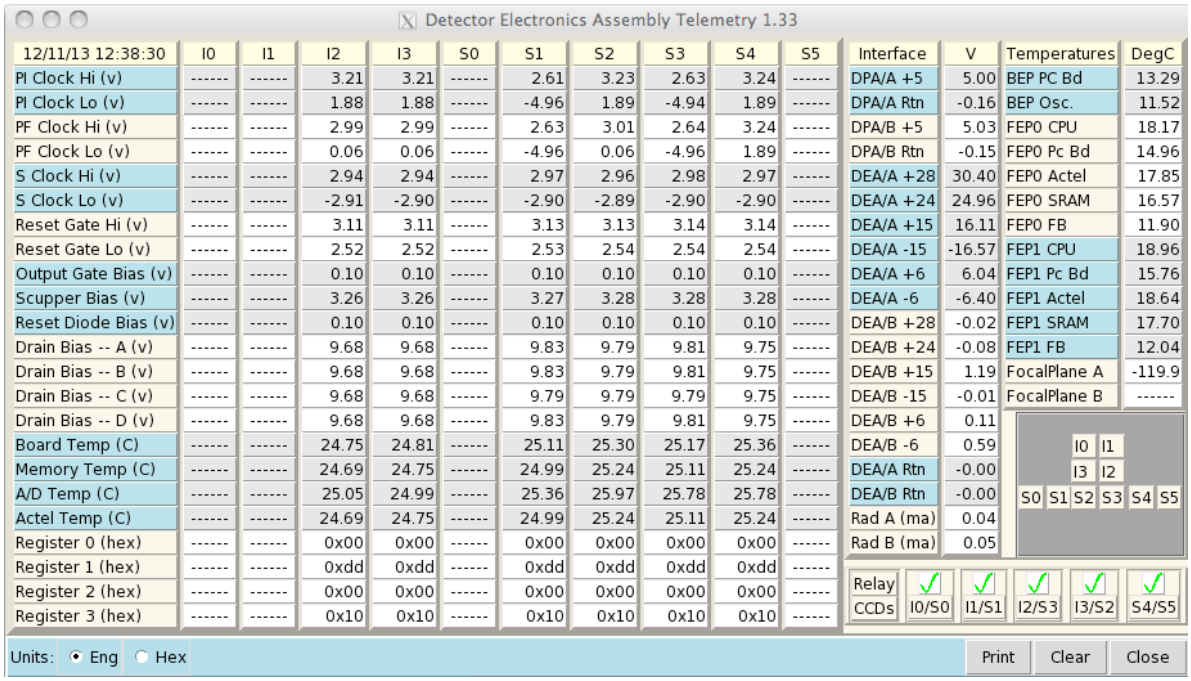

Figure 15. Display of DEA housekeeping channels. The Board 11 display is identical to the right-hand section of this window, with the same set of control buttons beneath.

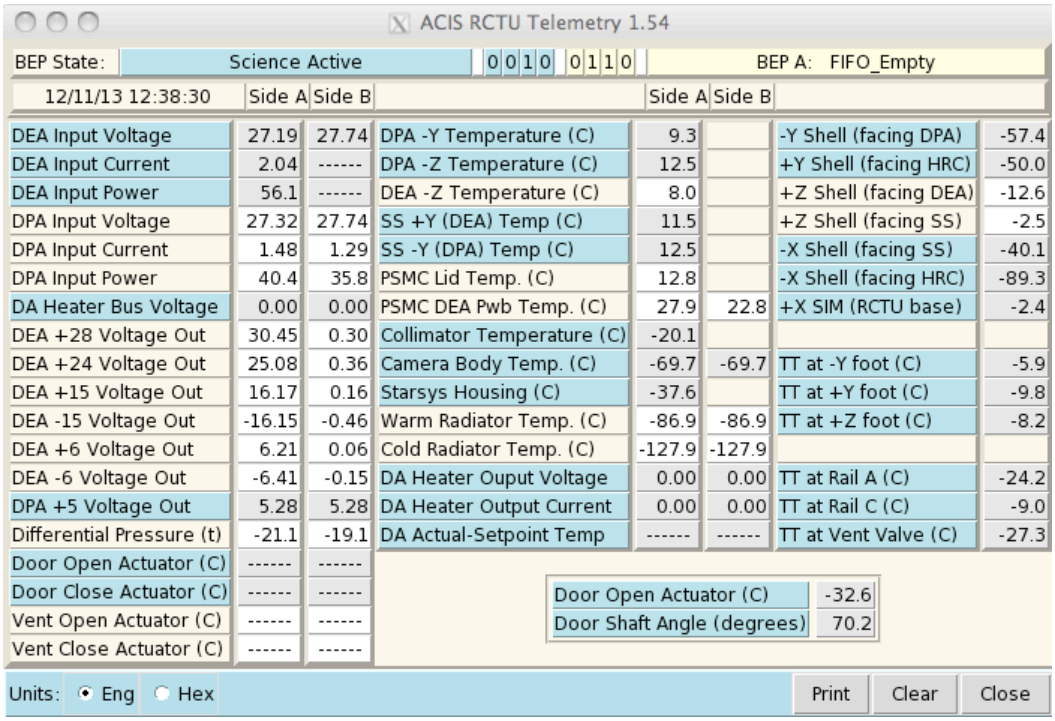

Figure 16. Display of ACIS engineering channels collected and reported by RCTU 12. In **–e** mode, only the state and bi-levels will be useful.

Figure 17. Display of *acisCtl* parameters. This example shows the items displayed in **–e** mode. Somewhat fewer are shown in **–f** mode. Any alterations to the text in the "Current Value" column will take immediate effect. To preserve for subsequent *acisCtl* runs, hit the "Save" button to save them in "*~/.acisctlrc*".

The remaining displays are text-oriented.

- Selecting "Start ACIS Interface" from the "I/O Server" menu displays the "ACIS Interface" window in which messages from the command and telemetry servers are displayed: chiefly commands to send to the engineering unit (**–e** mode), colored blue, and messages from *filterServer* in black. The display scrolls and its size can be altered by typing in the "Max Lines:" box, but the window cannot be resized.

- Selecting “Show Packet Monitor” displays a text window into which *psci* writes a one-line summary of every telemetry packet received. Packet names are boldfaced, numeric fields are written in blue and important (*i.e.*, disastrous) packet names are in red accompanies by an audible signal. The display scrolls and its size can be altered by typing in the “Max Lines:” box, but the window cannot be resized.
- “Show Command Monitor” is similar to “Show Packet Monitor”, but more selective. It lists lists commands sent to ACIS along with their return codes. The command mnemonic is colored green and error return codes are colored red. If the command is expected to return one or more data packets, these are summarized in blue. A series of *addPatch* commands is reported as a single entity.
- Selecting “Show PMON Monitor” displays a text window in which *pmon* displays statistics on command usage, FEP events, science parameters and reports, and software- and DEA-housekeeping messages.
- Several dialogs contain a “List” button which displays the contents of ASCII files or, in the case of the “Command Loads” dialog, first uses *lcmd* to convert binary packet files to ASCII before displaying them. For clarity, lines beginning “#” will be displayed in blue.
- Some dialogs contain an “Edit” button. The default editor is *gvim*. It is started in the directory of the item that is to be edited, so the editor’s “:shell” command can be used to execute UNIX commands in a sub-shell in that same current working directory. Use “exit” to return to the editor.

Whenever a window is closed, *acisCtl* keeps track of its location, updating “*~/.acisctlgeom*”. It also uses this file to tell the X11 window manager where to open a new window of that same type. As you can imagine, this situation is far from ideal should two copies of *acisCtl* be invoked simultaneously if they share the same home directory.

ENVIRONMENT

acisCtl uses a bewildering number of environment variables, almost all of which revert to default values unless specified. Some can be set by *acisCtl* command line options, while others are initialized in a parameter file whose default location is *“*`$ACISTOOLSDIR`*/lib/acisegse.params”*. The most useful ones can be displayed, modified, and saved from the “acisCtl Options” dialog invoked from the “Parameters...” item in the main *acisCtl* menu. See Figure 17 for an example.

`ACISPARAMSFILE` overrides the default location of the parameter file (see above). This is a csh shell script which is sourced immediately prior to the start of TCL/TK.

`ACISTOOLSDIR` The location of ACIS executables. The default is *“/nfs/acis/h4/tools”*.

`ACISTTMFILE` The telemetry mapping file to be used by *getPackets* or *getp* when locating ACIS engineering channels in telemetry frames. The default is *“acisEng+ObsidAll.ttm”*, and if the name doesn’t begin with “/”, *acisCtl* will assume that it is located in “`$ACISTOOLSDIR`/*lib*”.

`ACIS_CFGS` An ACIS command configuration file containing entries that specify sequences of ACIS commands (see `$ACIS_PBLKS`) to be executed for each SIMODE. The default is *“/nfs/acis/h1/www/bin/current.cfg”*.

`ACIS_IN_HRC` The location of ACIS data in format 1 telemetry, i.e., when the HRC instrument is in the focal plane. The default is “526,4,768,12”. See *getPackets*(1) for further details.

ACIS_PBLKS The file containing ACIS parameter blocks, also used by *pmon* to parse exposure and event packets. The default is *"/nfs/acis/h1/www/bin/current.dat"*.

BEP_MAP (**–e** mode only) A load map of BEP flight software in the format output by the *gcc* compiler for use in the "Debug" dialog. The default map location is "`$ACISTOOLSDIR`/*lib /acisctl/acisBep.Map*".

CCD_SCALE The size, in screen pixels, of each CCD displayed by the "CCD Events" display window. The default is 128.

CMDLOG (**-e** mode with **-P** only) Controls command logging in the (obsolete) *acisShell* interface.

CTU_SIDE (**-e** mode with **-P** only) Controls CTUE component selection in the (obsolete) *acisShell* interface.

DATAHOST The name of the host that is physically connected to ACIS hardware interfaces. In **–f** mode, this will always be the name of the host running *acisCtl*. The default is the output from *"/bin/hostname"*, with trailing *".mit.edu"* removed.

DATAPORT The TCP port number on `$DATAHOST` from which the telemetry data is to be read. All *acisCtl* functions use the TCL "`socket`" command to open the port, and read telemetry packets in non-blocking mode.

DEA11LIMITS The pathname of a file containing yellow and red alarm limits for each of the housekeeping channels reported by the DEA interface board (Board 11). Each line begins with the following seven fields, delimited by space and/or tab characters. Lines beginning "%" are ignored.

- **name** The channel mnemonic as defined in Section 5.14 of the IP&CL Release Notes, omitting the leading "`DEAHOUSE_CNTL_`".
- **board** The ID of the DEA board: always 10 and unused by *acisCtl*.
- **count** The conditioning count: unused by *acisCtl*.
- **red_lo** The lower red alert limit: valid channel values below this value will be displayed with a red background.
- **yel_lo** The lower yellow alert limit: valid channel values below this value will be displayed with a yellow background.
- **yel_hi** The upper yellow alert limit: valid channel values above this value will be displayed with a yellow background.
- **red_hi** The upper red alert limit: valid channel values above this value will be displayed with a red background.

Lower limit values of -1000 and upper limit values of 1000 will be disregarded, i.e., will not cause the backgrounds to be colored.

DISP_TERM The window application that is to display ASCII output from *acisPmon*, *psmcDown*, *DoPSMC*, or from `$EDITOR`. The default is "*rxvt*" on SunOS or Solaris hosts, and "*xterm*" on others.

EDITOR Command to edit ASCII parameter block files. The default is "*xgim*".

ENGLIMITS The pathname of a file containing yellow and red alarm limits for each of the engineering channels displayed in the "ACIS RCTU Telemetry" window. Each line begins with the following five fields, delimited by space and/or tab characters. Lines beginning "%" are ignored.

- **name** The channel mnemonic as defined by the Chandra Telemetry Database. Only ACIS and ISIM channels (beginning "1" and "3" respectively) will be selected.
- **red_lo** The lower red alert limit: valid channel values below this value will be displayed with a red background.
- **yel_lo** The lower yellow alert limit: valid channel values below this value will be displayed with a yellow background.
- **yel_hi** The upper yellow alert limit: valid channel values above this value will be displayed with a yellow background.
- **red_hi** The upper red alert limit: valid channel values above this value will be displayed with a red background.

Lower limit values of -1000 and upper limit values of 1000 will be disregarded, i.e., will not cause the backgrounds to be colored.

EXECUTE_CMD (**–e** mode only) The command (or */bin/sh* pipe) to use to read ASCII commands in *bcmd* format, convert them to binary, and send them to *cserver*. The default is "*bcmd* | *cclient* `$DATAHOST 8541`".

FEP_MAP (**–e** mode only) A load map of FEP flight software in the format output by the *gcc* compiler for use in the "Debug" dialog. The default map location is "`$ACISTOOLSDIR`*/lib /acisctl/acisFepSci.Map*".

FILTERSERVER_OPTS Options to pass to *filterServer* when invoked to pass ACIS telemetry packets to *filterClient* processes. The default is "`-v -n16`".

FIXED_FONT The name of the fixed-width font to use in listings. The default is "`-*-courier-medium-r-*-*-14-*`".

GETPACKETS_CMD The command to use to extract ACIS packets from telemetry frames. The default is "*getp*".

GRAB_WINDOW The command used to translate an *acisCtl* window into PostScript, to be printed by `$PRINT_CMD`. The name of the window will be appended. The default is "*import -gravity Center -resize "540x720>" ps:-*", which assumes that the ImageMagick import command is in the user's `$PATH`. Otherwise, you should use the command "*xwd*".

IMAGE_LIB (**–e** mode only) The name of the directory containing ASCII image loader pixel definition files. The default is "`$ACISTOOLSDIR`*/lib/images"*.

LOAD_IMAGE_CMD (**–e** mode only) The command to use to load a file generated by *genObjectImage* into the ACIS Image Loader. The command takes one argument, the name of the file, and will be executed on `$DATAHOST`. The default is "*putImages –q*".

LOAD_RAW_CMD (**–e** mode only) The command to use to load a file containing binary 16-bit ACIS commands, preceded by 4-byte headers, *i.e.*, as output from "*bcmd*". The default value is "*dapkts -foo* | *cclient* `$DATAHOST 8541`".

LOGMAXBYTES The maximum uncompressed size of a log file created by *pktCpy* (or *tlmGet* in **–f** mode) at which a new file will be started.

LOGWAITSECS The mininum waiting time for new input data in *pktCpy* (or *tlmGet* in **–f** mode) after which a new file will be started.

PARAM_BLOCK_LIB (**–e** mode only) The location of the directory containing ACIS parameter blocks in *bcmd* input format, used by the commands in the "Primary Menu" to list, load, and execute parameter blocks. The default value is "`$ACISTOOLSDIR`*/lib/pblocks*".

PIXEL_AB_DEV (**–e** mode only) The name of the device on `$DATAHOST` that controls the pixel switch, i.e., it switches FEP input between the Image Loader and the DEA. The default is "*/dev/ttya*".

PMON_CMD The name of the executable that will run *pmon* in a detached window. The default is "*acisPmon*".

PRINT_CMD The name of the command that sends PostScript to a printer. The default is "*lpr -P*`$PRINTER`".

PSMC_CMD (**–e** mode, and **–f** mode with **–P**) The command (or */bin/sh* pipe) that sends commands to the PSMC through `$PSMC_SERVER`. The default is "*bcmd* | *cclient* `$PSMC_SERVER 8541`".

PSMC_SERVER (**–e** mode with **–P** only) The name of the host commanding the ACIS PSMC through a second L-RCTU or CTUE. (**–f** mode) The name of the host running *acisCtl*. Its default is "*localhost*".

PSMC_PORT (**–e** mode with **–P** only) The port on `$PSMC_SERVER` sending PSMC through a second L-RCTU or CTUE. The default is "`7001`". (**–f** mode) The local port sending ACIS packets. The default is "`7002`".

PSMC_TLM_CMD (**–e** mode with **–P** only) The name of the command (or */bin/sh* pipe) that is to receive telemetry from the PSMC interface. The default is "*tclient -h* `$PSMC_SERVER 7001` | *ltpxd*", but the interface hasn't been used since launch.

RAWLOG (**–e** mode with **–P** only) This variable controls telemetry logging in the (obsolete) *acisShell* interface. Its functions has been replaced by `$TLM_LOG_DIR` and `$TLM_LOG_FILE`.

RAW_CMD_LIB (**–e** mode only) The directory containing raw packets to be loaded into ACIS via `$LOAD_RAW_COMMAND`. The default is "`$ACISTOOLSDIR`*/lib/patches*".

RCTU_CMD The command (or */bin/sh* pipe) that receives ACIS data and starts a telemetry server on port `7002`. In **–e** mode, it also accepts commands on port `8541`. Three servers are currently available: *acisTstShim* to run in the Chandra TST room, receiving minor-frame telemetry via TCP port `7543`; *acisEUshim* to interface with the L-RCTU, sending commands to the ACIS engineering unit and receiving telemetry in return; and the now obsolete *acisServer*, which is similar to *acisTstShim*, but can also interface to the engineering unit via a CTUE (which we no longer possess!)

RCTU_DUMP_DIR The directory to hold telemetry log files when "Start Minor Frame Logging" is selected from the "I/O Server" menu. The default is "*/tmp*".

RCTU_DUMP_FILE The name of the file that is to receive minor-frame data when "Start Minor Frame Logging" is selected in the "I/O Server Menu". The default value is "*tlm*-`$DATA-HOST`-*%Y-%m-%dT%T.mnf*", where the "%" fields will be replaced by the values current when the file is opened, as described in *strftime*(3), *e.g.*, "*tlm-emily-2013-01-31T10:40:23.mnf*". Note that, if the file is to be compressed (see `$LOGCOMPRESS`, above), a further extension, e.g., "*.gz*" or "*.Z*", will be appended.

TCL_LIBRARY The directory containing runtime TCL functions. For Linux and Solaris systems, the default is "`$ACISTOOLSDIR`*/lib/tcl8.5*", and for SunOS4 it is "`$ACISTOOLSDIR`*/lib/tcl7.6*".

TERMLINES The default number of lines to retain in scrolling *acisCtl* windows

TEXT_FONT The name of the font to use in text windows. The default is "*-*-fixed-medium-r-*-*-12-**". When bold-faced characters are desired, the same font will be used with "*medium*" replaced by "*bold*".

TK_LIBRARY The directory containing runtime TK functions. For Linux and Solaris systems, the default is "`$ACISTOOLSDIR`*/lib/tk8.5*", and for SunOS4 it is "`$ACISTOOLSDIR`*/lib/tk4.2*".

TLM_LOG_DIR The directory into which to write packet log files when "Start Packet Logging" is selected from the "I/O Server" menu. The default is "*/tmp*".

TLM_LOG_FILE The name of the file that is to receive packet data when "Start Packet Logging" is selected from the "I/O Server" Menu. The default value for this variable is "*/tmp/tlm*-`$DATAHOST`-*%Y-%m-%dT%T.pkt*", where the "%" fields will be replaced by the appropriate date and time values current when the file is opened, as described in *strftime*(3). Note that, if the file is to be compressed (see `$LOGCOMPRESS`, above), a further extension may be appended to the name, *e.g.*, "*/tmp/tlm-emily-2013-01-31T10:40:23.pkt.gz*".

ACISCTL TCL COMPONENTS

The following TCL files are located in the "`$ACISTOOLS`*/lib/acisctl*" directory. They are either loaded by *acisCtl.tcl*, or run in the background by separate instances of the *wish* interpreter.

acisCtl.tcl	Main component of *acisCtl*. Starts and stops interfaces.
acispower.tcl	Manages the "FEP/CCD Power" dialog.
acisTable.tcl	Displays and executes ACIS configurations and command packets.
beppower.tcl	Manages the "DPA Power" dialog.
deaif.tcl	Manages the "DEA Interface" dialog.
debug.tcl	Manages the "Debug" Interface dialog
dump.tcl	Manages the "Dump / Reset" dialog.
highspeedtap.tcl	Manages the "High-Speed Tap" dialog.
imageLoad.tcl	Manages the "Image Loader" dialog.
interface.tcl	Manages the I/O server interface window.
options.tcl	Manages the "Preferences..." dialog.
pblocks.tcl	Manages several parameter block dialogs.
psmc.tcl	Manages the "PSMC Control" dialog.
rawpackets.tcl	Manages the "Command Loader" dialog.
rctu.tcl	Displays RCTU engineering channels.

runacis.tcl	Manages the main "Run ACIS" menu.
showit.tcl	Displays CCD event locations.
showtlm.tcl	Displays telemetry packet summaries.
textDisp.tcl	Displays a text file in a scrolling window.
video11.tcl	Displays DEA interface board channels.
videotm.tcl	Displays all DEA housekeeping channels.

OTHER EXECUTABLES

Filename	Type	Description
acisCtl	*csh* script	Wrapper for "*acisCtl.tcl*".
acisEUshim	*sh* script	Format 2 interface to engineering unit in **–e** mode.
acisEUshim500	*sh* script	Format 1 interface to engineering unit in **–e** mode.
acisPmon	*sh* script	Wrapper for "*pmon*".
acisTables	*perl* script	Extracts entries from configuration and command files.
acisTstRemote	*sh* script	Interface to remote *filterServer* in **–f** mode.
acisTstShim	*sh* script	Interface to COG client in **–f** mode.

The following ACIS EGSE executables are also invoked by components of *acisCtl.*

bcmd	*perl* script	translate non-critical ASCII commands to binary.
bcmdCritical	*perl* script	translate critical ASCII commands to binary.
cclient	UNIX binary	send commands or data via TCP to *cserver.*
cserver	UNIX binary	receive commands or data from *cclient* connection.
dapkts	UNIX binary	copy binary ACIS commands, inserting delays.
filterClient	*perl* script	log ACIS packets from *filterServer.*
filterServer	*perl* script	send ACIS packets to *acisCtl* and *filterClient* sockets.
pktCpy	UNIX binary	copy and, optionally, log a nACIS packet stream
pmon	UNIX binary	reformat ACIS packets and display on smart terminals.
rxvt	UNIX binary	smart terminal emulator, alternative to *xterm.*
tclient	UNIX binary	receive binary CTUE data from *tserver.*
tlmGet	UNIX binary	TCP server to receive and, optionally, log raw data
tserver	UNIX binary	send CTUE data to *tclient* connections.

OTHER FILES AND DIRECTORIES

`$ACISTOOLS`*/bin/*`$ARCH`	Directory containing *acisCtl* executables.
`$ACISTOOLS`*/lib/acisctl*	Directory containing *tcl* scripts invoked by *acisCtl*.
`$ACISTOOLS`*/lib/acisctl/acisBep.map*	Default BEP load map for the *acisCtl* debugger.
`$ACISTOOLS`*/lib/acisctl/acisFepSci.map*	Default FEP load map for the *acisCtl* debugger.
`$ACISTOOLS`*/lib/acisctl/images*	Directory containing image files used by *acisCtl*.
`$ACISTOOLS`*/lib/*`$ARCH`	Directory containing architecture-dependent libraries.
`$ACISTOOLS`*/man*	Directory containing online manuals.
~/.acisctlrc	Default user *acisCtl* startup file.
~/.acisctlgeom	File containing location of *acisCtl* windows.
/nfs/acis/h1/www/bin/current.cfg	Default ACIS offline-system configuration tables.
/nfs/acis/h1/www/bin/current.dat	Default ACIS offline-system command tables.
/nfs/acis/h1/www/asc/pmon-limits.txt	Default yellow/red alarm limits for Board 11 channels
/nfs/acis/h1/www/asc/aciseng-limits.txt	Default yellow/red alarm limits for RCTU channels

EXAMPLE

Here are the steps needed to conduct a science run on the ACIS engineering unit using *acisCtl*. The test assumes that it is being run from the interface host, currently *cypress.mit.edu.*

- Type *acisCtl –e* on the command line.
- Select “Start ACIS Interface” from the main menu and wait until “load Complete” appears in the “ACIS Interface” window.
- It is good practice to monitor subsequent ACIS output by selecting “Show PMON Monitor” from the main menu. The *pmon* display will appear in a separate window on the right of the screen.
- Either cold-boot the ACIS BEP or reset its patch list. Cold-booting is performed by selecting “Control DPA Hardware...” from the main menu and then “Cold Boot” in the “DPA Hardware” dialog; resetting the patch list is done by selecting “BEP Dump/Reset...” from the main menu and then “ResetPatch List” from the “Dump/Reset” menu. Cold booting is safer.
- Now install patches by selecting “Command Loads...” from the main menu, selecting a patch file in the “Command Loader” dialog and clicking the “Execute” button in that window. That button will turn pink while the patches are being loaded. (By convention, full loads of a given patch release are named “*all-XYZ.pkts*”, where “*X*” represents the standard level, “*Y*” the optional level, and “*Z*” the certifications level. Such a patch load will configure the BEP to expect input from the Image Loader, *i.e*., to ignore the video boards. To use video boards, load the alternative “*all-XYZ-dea.pkts*” file which includes the “*deaeng*” patch that accesses the video boards, both flight and engineering versions.)
- Warm boot the BEP by selecting “Control DPA Hardware...” from the main menu and then “Warm Boot” in the “DPA Hardware” dialog;
- Start receiving housekeeping from the DEA interface board by selecting “DEA Housekeeping” from the main menu and then selecting a suitable housekeeping command file, *e.g*., “*fullhouse.dea*” from the “DEA Housekeeping” dialog, and then hitting the “Load” and “Start” buttons in that same window.
- Before starting a science run, power up the necessary CCDs and FEPs by selecting “Control FEP/CCD...” in the main menu. Clicking the “Refresh” button will (after a short delay while ACIS is being interrogated) display tick marks against the FEPs and CCDs that are currently powered up. Click in these boxes to change the configuration, and then select the appropriate “Send” button to apply your choices. Allow a second for each CCD to power-up and 10 seconds for each FEP.
- Don’t forget to specify FEP input before you start a run. Click either “DEA” or “Image Loader” in the main menu and wait for the button to turn green. The former will feed the DEA with digitized “noise” from the DEA, the latter with pixels from the image loader. Be sure that you know which patches are running in the engineering unit: to receive input from the DEA, the *deaeng* patch must be loaded; conversely, for image loader input, *dearepl* must be loaded. To send an image to the image loader, select a suitable definition file in the “Image Loader Control” window and click the “Load” button.
- You are now ready to start a science run. If it needs window blocks to be loaded, select either “2-D Windows” or “1-D Windows”, and choose and load them into the appropriate slots. Then select “Timed Exposures...” or “Continuous Clocking...”, choose a suitable parameter block, “Load” it into the BEP, and start the run by clicking the “Start” or “Start Bias” button.
- End a science run by selecting either “Timed Exposures..” or “Continuous Clocking...” and clicking the “Stop” button in the dialog.

- Alternatively, you can run an entire ACIS SIMODE from a set of ACIS offline system tables, *i.e.*, those defined by the filenames in the `$ACIS_CFGS` and `$ACIS_PBLKS` variables. Click "ACIS Configurations..." in the main menu, choose an SIMODE from the list, and click "Execute". This will send the commands to the engineering unit with the appropriate delays. All you have to do is to keep the image loader supplied (if you're not using the DEA), and stop the run at some point, via the "Stop" button in the "Timed Exposures..." or "Continuous Clocking..." window.

AUTHORS

Bob Goeke, Demitrios Athens, Peter Ford; MIT CSR, MKI

IMPLEMENTATION

acisCtl is a Bourne shell script that embeds the user's parameter file (by default, "*~/.acisctlrc*" but see `$ACISPARAMSFILE`) and then executes *"wish –f* `$ACISTOOLSDIR`*/lib/acisctl/acisCtl.tcl"* in the background. The TCL interpreter, *wish*, must therefore be in the user's `$PATH` before starting *acisCtl*. Once started, *"acisCtl.tcl"* loads other TCL modules from *"*`$ACISTOOLSDIR`*/lib/acisctl/",* supplies defaults for any undefined environment variables, and displays the "acisCtl" menu.

The most crucial menu item is the "Start ACIS Interface" item starts the `$RCTU_CMD` process in the background, copying its *stdout* to a separate *acisCtl* text window. Any *stderr* output from `$RCTU_CMD` will be treated by *acisCtl.tcl* as an error, so `$RCTU_CMD` shouldn't generate any unless it intends to fail, in which case the *stderr* text will be displayed in a dialog on the user's screen. Meanwhile, *stdout* text in the interface window will be highlighted according to some simple rules applied in *interface.tcl*.

In **–f** mode, the *stdin* to `$RCTU_CMD`, will be closed. In **–e** mode, its *stdin* will be sent commands in *bcmd* output format. If a secondary PSMC server is used (the **–P** option of **–e** mode), PSMC commands will be sent from a *cclient* process to port 8541 of `$PSMC_SERVER`, and a *tserver* process will listen for *tclient* connections to port 7001 of `$PSMC_SERVER`.

In general, all *acisCtl* commanding is performed by the *"acisCtl.tcl"* process, whereas telemetry is displayed by separate invocations of *wish*, the *tcl* interpreter, which inherit the environment variables established by *acisCtl*. Most of the telemetry display processes can also be started directly from the user's terminal, *e.g.*, the DEA telemetry display from "Show All DEA Telemetry" can also be displated by typing "*wish –f* `$ACISTOOLSDIR`*/lib/acisctl/videotm.tcl"* at the console, although it may first be necessary to load environment variables, *i.e.*, *"source ~/.acisctlrc*".

The most convenient way to test *acisCtl* components is to launch it with the **–f** flag, start the default `$RCTU_CMD` server, *acisTstShim*, and then feed it with minor-frame telemetry via port 7543, *e.g.*, *"ehs2mnf /nfs/maax/r2/eh/ssr/2012** | *pthrottle -m* | *cclient cypress* 7543*"*. Note the use of *pthrottle* to slow down the rate at which *cclient* sends data to *acisTstShim*. If removed, *acisCtl* would most likely gag as its sockets would fail to keep pace with the incoming data. If a modest increase in data rate is desired, use *pthrottle*'s **–t** option.

When any *acisCtl* window is closed, its location is saved in the user's "*~/.acisctlgeom*" file. This file is reread whenever a window is opened, so that it can be drawn in the same screen location as before. This can cause confusion when running *acisCtl* on a smaller screen than before, in which case the "*~/.acisctlgeom*" file should be edited or removed before starting. When the main *acisCtl* window is closed, it closes all *acisCtl* windows that remain open, saving their locations. Note that most *acisCtl* windows are not currently resizable because some X11 servers have been found to crash when their Tcl/Tk windows are resized while they are refreshing.

Windows that only contain ASCII text are printed by piping them through "*enscript –p– |* `$PRINT_CMD`". Those containing graphics are printed with "`$GRAB_WINDOW` | `$PRINT_CMD`". The "Print Window" button in the main menu waits for the user to click the mouse on the desired window, which can be any window displayed by X11, not just those created by *acisCtl*; the other "Print" buttons insert the name of their window into the user's `$GRAB_WINDOW` definition, either "*xwd –name* `win`" or "*import –window* `win`". For this reason, *acisCtl* requires that the strings "xwd" or "import" must occur at some location in the `$GRAB_WINDOW` string.

DIAGNOSTICS

Most errors that are caught by *acisCtl* components cause diagnostic messages to be displayed in dialog windows, after which processing continues. The following errors are caught at startup time, and are written to the *stderr* stream, after which *acisCtl* terminates.

- *acisCtl:* **too many environment variables defined**

  On older UNIX systems, there is a limit of 100 environment variables that can be supplied to any one process. If you see this message, you should check your *~/.cshrc* or `$HOME`*/.profile* and remove as many unused environment variables as possible.

- *acisCtl:* **unrecognized window processor in GRAB_WINDOW:** *string*

  The `$GRAB_WIN` variable must contain either "`xwd`" or "`import`".

Also, when *acisCtl* components that read telemetry packets in non-blocking mode experience errors that cause them to terminate, a brief diagnostic message is written to the *stderr* stream of the terminal from which *acisCtl* was invoked.

BUGS

- The user can change the heights of some *acisCtl* windows, e.g., those displaying scrolling text, but they cannot be widened.