

Sec	Page	Function
010	1	Introduction
020	1	Getting Started - Things you MUST do
030	2	Location of Important Files
040	3	PACMAN Configuration File
050	6	PACMAN Run-time Help File
060	7	Running PACMAN
070	9	Histogramming Tasks
080	9	Control of VME Front-End Processor
090	11	FILE Commands
100	12	Miscellaneous Commands
110	14	System Status Commands
120	16	Testing VME/Workstation Connection
130	18	What to Do if PACMAN Refuses to RUN
140	19	How to Exit PACMAN
150	19	Run-time Help Text
160	23	Built-in command list
170	23	How to Access On-Line Documentation

PACM.010 Introduction

PACMAN is an executive which controls all processes necessary to run the data acquisition system and write event data to a list-data file (LDF file). It initially loads all processes required and continually monitors those processes throughout data acquisition. If any essential process terminates, the user is notified of what corrective action is required. The commands to PACMAN fall into 4 categories:

- 1).. VME Front-end processor control
- 2).. File control
- 3).. System status and test
- 4).. USER commands defined in a configuration file

Throughout this document examples of screen output will be reproduced. In these examples, user input will be in Bold print.

PACM.020 Getting Started - Things YOU need to do.

- 1).. Setup your VME System
 - see Section 030 of VME Front-End System document
 - see document file - /usr/hhirf/vme/doc/VMEsys.doc
- 2).. Configure the VME/Workstation system
 - see Section 060 of VME Front-End System document
 - see document file - /usr/hhirf/vme/doc/VMEsys.doc

- 3).. Prepare your PACOR program
PACOR user documentation
see document file - /usr/hhirf/doc/pacor.doc
- 4).. [optional] Make a PACMAN configuration file
see Section 040
- 5).. [optional] Make a custom run-time help file
see Section 050
- 6).. Start PACMAN and start data acquisition
see Section 060
- 7).. Start your histogram tasks
see document file - /usr/hhirf/doc/scanor.doc

PACM.030

PACM.030 Location of Important Files

You may start PACMAN in any directory you wish. However, there are two files which PACMAN processes will access or create if necessary. These files will ALWAYS be in your home directory (i.e. login directory). The files are orphas.vme? and pacman.fig.

orphas.vme?

Before starting PACMAN, you MUST set the environment variable VME to the VME processor you will be using. For example, if the VME processor is vme2, you should

```
setenv VME vme2
```

The log file name will be orphas.vme2.

If the log file does not exist in your home directory, PACMAN will create it. Otherwise, PACMAN appends to the existing file. Each entry has the name of the process which originated the message, a time stamp and a message. Most of the commands to PACMAN are logged. The log file is a fairly complete record of what has been done.

Remember, the log file is ALWAYS in your home directory.

pacman.fig

This your custom PACMAN configuration file. PACMAN expects this file to be in your home directory unless you specify a path and file name on the command line invoking PACMAN. If there is no pacman.fig in your home directory and no file is specified on the pacman command line, the default

configuration file is /usr/acq/wks/pacman.fig.

PACM.040 PACMAN Configuration File

Several PACMAN parameters may be customized by making your own configuration file. This section describes those parameters you may specify.

The syntax rules are very simple. Each statement type is identified by a keyword. First level keywords must be the first word on a line and must be upper case. If a statement requires more than one line, there must be a left curly bracket on the first line and a right curly bracket at the end of the last line.

The statement \$USER_CMD allows a second level of keywords which must be lower case and keywords are separated by semicolons.

\$BANNER

This is the PACMAN startup banner. If not present in your configuration file, the default is identical to that shown below in the listing of /usr/acq/wks/pacman.fig.

\$BUFFER_SIZE

This parameter defines the size of event data buffers in bytes. The buffer size should be a power of 2 and in the range of 2048 thru 32768.

If this parameter is not present in the configuration file, the default is 32768.

\$LOG_INTERVAL

The acquisition system writes rate information to the log file periodically. This parameter is the time interval in seconds. For example,

\$LOG_INTERVAL 600

means to log the data rates every 10 minutes.

If the parameter is not present in your configuration file, the default is 300.

\$USER_PFTOIPC

The acquisition system process which receives data from the VME system is pftoipc (Packet Filter to Interprocess Communication). This process receives ethernet packets from the VME and builds event data buffers in shared memory. All other processes get event data from shared memory.

The default process is /usr/acq/wks/pftoipc. The user may provide a custom process to replace pftoipc. If you have a custom process, you need to define it in your configuration file. The statement

\$USER_PFTOIPC /usr/users/myacq/mypftoipc

means that the process /usr/users/myacq/mypftoipc is to be used instead of the default.

\$HELP_FILE

If you have a custom help file, you need to define it in your configuration file. The statement,

```
$HELP_FILE /usr/users/mcsq/my.help
```

means that the file /usr/users/mcsq/my.help will be read as the help file instead of the default /usr/acq/wks/pacman.fig.

\$WINDOW

While it is quite easy to create new xterm windows, there are some drawbacks. Usually all windows have the same name and the position of the Icon is selected by the XWindow manager. Sometimes it is convenient for a window to have a unique name associated with the task usually performed in that window. The \$WINDOW statement allows you to specify a window to be created when PACMAN starts. For example, the statement

```
$WINDOW scan = {/usr/bin/X11/xterm -geometry 80x32-0+10 #-41-38
                -sb -ls -iconic }
```

creates a window named SCAN. The position of the window and the Icon are specified. Also, the window starts as an Icon. See the manual pages X and xterm for additional information.

Windows may not have the same name as an existing command. A command by the same name as the window will be added to PACMAN. This command is available to recreate the window should you accidentally delete it.

\$USER_CMD

Typically users have hardware specific codes to set high voltages, discriminator thresholds, etc. You may elect to run these codes in temporary windows from PACMAN. The statement \$USER_CMD may be used to define the code to be executed and the default directory while the code executes. For example, the statement

```
$USER_CMD { dmon = /usr/bin/X11/xterm -geometry 80x50+10+10 -sb -e
             /usr/acq/wks/dmon }
```

creates a PACMAN command dmon. When you type dmon in the PACMAN window, a temporary window is created and the code /usr/acq/wks/dmon begins execution in that window. When dmon terminates, the temporary window is deleted and you return to PACMAN.

NOTE: PACMAN will not accept additional commands until a user defined command terminates.

There are two additional command statements valid only within a \$USER_CMD statement. These are second level commands which begin with a \$ and MUST be lower case. Commands within the \$USER_CMD statement MUST be separated

by semicolons.

```
$stopvme; .....
```

Often, hardware setup codes require that acquisition in the VME front-end system be stopped. This statement directs PACMAN to do a STOPVME command prior to running your code.

```
$cd directory; .....
```

This statement specifies the default directory while you code is executing.

The following example shows the usage of these second level commands. Note the semicolons at the end of the second and third lines. They are essential!

```
$USER_CMD { mycode = /usr/bin/X11/xterm -geometry 80x50+10+10 -sb -e
             ~myname/setup/mycode;
             $stopvme;
             $cd ~myname/setup }
```

```
$ALIAS .....
```

If you don't like the command mnemonics we have chosen, you may assign a new name to any existing command using the \$ALIAS statement. However, the new name MUST be unique. Furthermore, the log file entries retain the original command name. For example, the statement

```
$ALIAS go trun
```

defines go to mean the same thing of the built-in command trun.

The following is a listing of the default configuration file, /usr/acq/wks/pacman.fig.

```
-----
$BANNER {
*****
*                               ORPHAS                               *
*                               *                                   *
*                               pacman II                           *
*                               Physics Acquisition Manager          *
*                               *                                   *
*****}
$BUFFER_SIZE 32768
$LOG_INTERVAL 300
$HELP_FILE /usr/acq/wks/pacman.hep

$USER_CMD {emacs = /usr/local/bin/emacs}

$USER_CMD {
dmon = {
    /usr/bin/X11/xterm
    -geometry
    80x50+10+10
```

```
-sb
-e
/usr/acq/wks/dmon };
}

#if unknown displays

$WINDOW scan = {/usr/bin/X11/xterm -geometry 80x32+345+10 #+955+722 -sb -ls
-font 8x13bold -iconic }

$WINDOW {
damm = { /usr/bin/X11/xterm -geometry 80x32+10+320 #+950+744 -sb -ls
-font 8x13bold -iconic } }

#endif
```

PACM.050 PACMAN Run-time Help File

If you have a custom configuration file, you may wish to also make a custom help file.

A line for which the first 4 characters are \$\$\$\$, is the beginning of a Help topic. All characters on the line following the \$\$\$\$ become an entry in the help directory which is displayed when you type help. The 4 characters immediately following the \$\$\$\$ must be unique. These 4 characters are the help topic. Help topics may have trailing spaces but they may NOT have leading spaces.

Lines following the help topic line are saved as is until another help topic line or the end-of-file is encountered. What you see in the file is what you get on the screen.

Section 150 has a complete listing of the default help file.

PACM.060 Running PACMAN

Login at the console of the workstation. You probably have only one window. In that window, type

```
pacman [file.fig]
```

NOTICE: It is important, for reasons I will not attempt to explain, that you do this in your login window. So, why don't you just humor the old fat man and do it my way?

The file.fig argument on the command line is optional. If the argument is present, it should be path and file name of a configuration file. Note that all examples in this document assume use of the default configuration file - /usr/acq/wks/pacman.fig.

After about 5 seconds, PACMAN should be started and initialized. At the bottom left of the screen is a long 10 line window. Every message appearing

in this window is written to the log file - orphas.vme? in your home directory. At the bottom right of the screen there are two icons, DAMM, and SCAN. Nothing is running in either of these windows. These were created and named for functions you are likely to use. The size and position of these windows may be changed as a you would any other window. There are no restrictions on what you may run in these windows.

In your PACMAN window, you will see the startup banner and a directory of the help available.

```

-----
Using configuration file: /usr/acq/wks/pacman.fig
*****
*                               ORPHAS                               *
*                               *                                   *
*                               pacman II                           *
*                               Physics Acquisition Manager           *
*                               *                                   *
*****
VME vme3, Beam OFF, Events          0, Records 0

```

```

FILEOU->PID = 28428, Old Priority = 0, New Priority = -15
PID = 28429, Old Priority = 0, New Priority = -10

```

The latest PACMAN documentation is dated: Tue Sep 7 17:10:04 2004
If your copy is older, discard it and print a new copy using the command - prtdoc.

- Type: H SCAT - For Scaler-to-File related commands
 - Type: H VME - For commands to control and get status of Front-End
 - Type: H ACQ - Commands related to data acquisition
 - Type: H FILE - For File control commands
 - Type: H DISP - For commands which display Data-records
 - Type: H STAT - Commands to display system status and test VME system
 - Type: H MISC - Miscellaneous Commands
 - Type: H EXIT - How to STOP the acquisition system
 - Type: H CMDS - Command list
 - Type: H NEW - How to update Help messages
- pacman:

These help topics contain information about commands that may be useful in your work. The help includes most of the commands available. In latter sections, we will explore the commands in greater detail.

The remainder of this section is an example of what is needed to start acquisition to disk file after all hardware setup is complete. The steps required are:

- 1).. Load VME acquisition software
- 2).. Compile and load the proper PAC file into the VME processor
- 3).. Setup the FILEOU process to record the data stream on disk

4).. Start acquisition

pacman: loadacq
Delete the acquisition tasks in the VME processor

Now load the acquisition tasks. Any error messages here are BAD news.

Acquisition system now loaded

pacman: pacor linux 1
#BYTES OBJECT CODE GENERATED = 584
CONSIDER YOURSELF LOADED WITH 584 BYTES
NO ERRORS

pacman: hnum 1

pacman: FILEOU->htit testing 1,2,3

pacman: FILEOU->ouf /tera/mcsq/test.ldf

pacman: Open LDF file: /tera/mcsq/test.ldf

FILEOU->trun boff

FILEOU->pacman: PAC file: /tera/mcsq/Dlinux/Dpac/linux.pac - 999001
testing 1,2,3 - 1

pacman:

PACM.070 Histogramming Tasks

See the SCANOR documentation for details on making a histogramming task -
/usr/hhirf/doc/scanor.doc.

PACM.080 Control of VME Front-End Processor

From the PACMAN window you can do everything you need to do with the VME front-end processor. You can boot the VME processor, load the VME acquisition code, compile and load your PAC program, initialize the VME acquisition system, start VME acquisition, stop VME acquisition and get the acquisition status of the VME system.

BOOTVME

This command loads the code for a minimal operating system into the VME processor. After the basic operating system is loaded, the workstation can communicate directly with hardware such as FASTBUS and CAMAC. This operating system must be loaded and functional before using your setup codes which directly access CAMAC or FASTBUS.

You may ask when do I need to boot the VME system? The most obvious time is after any time when power has been off in the VME crate. There are also various other hints. If your setup programs give connection failure errors, it is a good hint that the VME operating system is not working. This can be checked with the LT command (see Section 110).

Just remember that after BOOTVME, there are several other things you must do in preparing to take data.

LOADACQ

Once the basic operating system is loaded(BOOTVME above), you can load the acquisition code into the VME processor. This is the code which decodes your readout instructions, responds to Event inputs, reads the specified hardware and sends event data to the workstation. This is a necessary but not sufficient step in starting data acquisition.

LOG your comments.....

The log file maintains a record of what has been done during data acquisition. The LOG command provides a way for you to insert comments into the log file. This line, including the keyword LOG, is written to the log file along with the date and time of day.

CD directory

Change PACMAN's working directory to the directory you specify. This applies only to the EDIT and PAC commands below. This is a convenient way to avoid typing complete paths for these two commands.

PWD

Show PACMAN's current working directory.

DIR [options].....

List the files in the current working directory. Use this command just as you would use ls in a normal window. In fact, you may type ls instead of dir.

EDIT filename

This command creates a new window on top of your PACMAN window and starts a vi session on the specified file. When you exit your vi session, the window is deleted and you return to PACMAN. Note that PACMAN will NOT execute another command until you exit the vi session! Also, when you exit the vi session, an entry will be made in the log file show that the EDIT command has been used. NO, it does not indicate if changes were make in the PAC file! Sorry about that.

PAC filename [L]

When satisfied with your PAC program, you need to compile and load the PAC into the VME processor. This command is identical to what you need to execute PACOR in any other window. See the PACOR documentation for details.

If you specified the L option and PAC completed without errors, the following commands are also executed:

- 1)... STOPVME - Stops the VME Acquisition
- 2)... INITVME - Initialize the VME acquisition system

If the INITVME command completes without error, the next TRUN command will cause the PAC file to be written to disk file.

INITVME

The VME acquisition system must be initialized after a new PAC is loaded. Since the PACOR command now does this, you no longer need to do the INITVME command. However, it does no harm. Note that you can NOT initialize the VME acquisition system if VME acquisition is running.

The workstation which sends the INITVME command is the one which receives event data and messages from the VME system.

Many errors are possible during the initialization of the VME(nonexistent CAMAC crates or FASTBUS modules etc. etc.). If there are errors, you will get an error message in the PACMAN window. However, additional information about the error(s) may be present in the logger window.

STARTVME

If the INITVME executed without error, you may now start the VME acquisition system. You must STARTVME if you want data sent to the workstation. The TRUN command also starts VME acquisition.

STOPVME

Stop the VME acquisition. Readout by the VME processor stops and no more data are sent to the workstation. TSTOP also stops VME acquisition.

STATVME

You may ask for the state of the VME acquisition system with this command. Possible responses are 1) not initialized, 2) running and 3) stopped. This status is reported only in the PACMAN window.

VMEHOST

Show the name of the workstation which receives event data and messages form the VME system. This the workstation from which the last INITVME command was issued.

PACM.090 FILE Commands

The process which records event data to an LDF file is called FILEOU. This is stripped down version of LEMOR. It has a limited command set with some syntax differences. There are many commands for positioning the output file, examining input data. If you are a LEMOR user, the main difference is that FILEOU has only one output stream and commands don't now have an output stream specification. If you are not a LEMOR user, you should first read the LEMOR documentation(See document file /usr/hhirf/doc/lemor.doc). In any case, the help topics ACQ, FILE and DISP have a brief description of all commands. FILEOU automatically writes an End-of-File and after every 10000 records.

There are a few commands which are significantly different from LEMOR. So, lets look at them.

OUF /tera/username/directory/file.ldf

You MUST specify the full path name of the list-data disk file. You may specify the file in two parts using the FDIR command. For example, the following sequence is equivalent to the above OUF command:

```
fdir /tera/username/directory
ouf file.ldf
```

Note that the FDIR command without an argument displays the current directory.

FDIR

As noted above, the output filename MUST be a FULL PATH name (i.e. the name MUST begin with a /). The command FDIR allows you to specify the full path of the directory in which the output list-data file is to be created. The OUF filename is then just a file within the directory specified in a previous FDIR command.

An FDIR command without an argument displays the current directory.

The directory path as specified in a FDIR command, is used for all OUF commands until a new directory is specified or the current directory string is erased. To erase the directory string, do

```
FDIR xx
```

since 'xx' is NOT a directory path, the directory string will be erased.

TRUN

This command starts recording event data on list-data disk file. TRUN also starts the VME acquisition.

If you have loaded a new PAC, the PAC file is written to the ourput file. The header number for PAC files begins with 999001 and is incremented after the file is written. The header title is the PAC file name. Example PAC file header:

```
PAC file: /usr/orph22/users/mcsq/test.pac          999001
```

You can recover a PAC file from the list-data disk file by searching for the header number and then using the STEX command in LEMOR.

TCONT

This command should be used when you are adding records to the last file.

TSTOP

In LEMOR, CTRL records. PACMAN does not support this use of CTRL the command tstop any time you would have used CTRL

SCAT

The SCAT commands control writing scaler data to the list-data disk file. For details, see the document /usr/hhirf/doc/scat.doc. For a list of commands, type scat help.

AFON

Enables automatic file marks on the output file. A file mark is written on the output file after every 10000 records. This is the DEFAULT.

AFOF

Disables automatic file marks on the output file.

PACM.100 Miscellaneous Commands

CMD and CMDF file

Normally commands to PACMAN are input via the keyboard. CMD or CMDF direct PACMAN to get commands from the specified file. By default, the PACMAN appends .cmd to the file argument. For example, the command

cmd disk

reads commands from the file disk.cmd in the current working directory. If there is a period in the argument file, PACMAN does not append the .cmd and file is taken as the complete file name. For example, the command

cmd disk.xxx

executes commands from the file disk.xxx.

DATE and TIME

Display the date and time. DATE and TIME are identical.

DIR and LS

List the files in the current working directory. Use this command just as you would use ls in a normal window. In fact, you may type ls instead of dir.

HUP

If your count rate is very low, it may take a long time to get a buffer of event data to be histogrammed. The HUP command forces any event data in the VME processor to be sent to the workstation. The workstation then pads the remainder of the current buffer with End-of-Events and passes the buffer to all processes getting event data from shared memory.

IPCS [options]

IPCS displays the status of the interprocess communication facilities. See the manual page ipcs for a description of the options field.

KILL ALL

This is the command to terminate PACMAN and exit. See Section 140 for details.

LOG

The log file maintains a record of what has been done during data acquisition. The LOG command provides a way for you to insert comments into the log file. This line, including the keyword LOG, is written to the log file along with the date and time of day.

PRTDOC

Use PRTDOC to print a copy of the PACMAN documentation.

PS [options]

PS displays the status of processes running in the workstation. See the manual page for ps for a description of the options field.

PWD

Show PACMAN's current working directory.

USERCONFIG

The PACMAN parameters which may set from a configuration file may be displayed using the USERCONFIG command. An example output follows:

```

-----
pacman: userconfig
***** Windows *****
SCAN /usr/bin/X11/xterm -n SCAN -geometry 80x32-0+10 #-41-38 -sb -ls -iconi
DAMM /usr/bin/X11/xterm -n DAMM -geometry 80x32+10+400 #-45-20 -sb -ls -ico

***** Command Aliases *****

***** User Commands *****
Command: dmon
       : /usr/bin/X11/xterm -geometry 80x50+10+10 -sb -e /usr/acq/wks/dmon

Help file: /usr/acq/wks/pacman.hep
Shared Memory buffer size (bytes): 8192
Log Interval (seconds): 300
pacman:
-----

```

PACM.110 System Status Commands

There may be rare occasions when you wonder if everything is working as you planned. Well, here are a few sanity checks. I will show a few

commands and what you should expect to see in the PACMAN window. Getting the expected response to these commands is a necessary but not sufficient condition to ensure that everything is working!

LT

This is a check of the VME system. The response shown below assumes that you have previously executed the BOOTVME and LOADACQ commands.

pacman: lt

VME Processor: vme2

ID	Memory/Task	Size	Address	pc	evt1/evt2	pri/time
0	T00 Task0	8	7000 9000	FF002CC4	97 0	64 1 W
1	T01 Lan_Driver	16	9000 D000	ABF8	56 64	70 1 W
2	T02 Mem_mgr	76	D000 20000	FDA8	0 0	69 1 R
3	M Acq_Params	128	20000 40000			
4	T03 cnafx	42	40000 4A800	424B0	-1 1	69 1 W
5	T04 fastxx	40	4A800 54800	4C9A8	-1 1	63 1 W
6	T05 data_proc	42	54800 5F000	56C38	72 0	69 1 W
7	T06 VMEacq	1174	5F000 184800	6873C	-1 1	69 1 W
8	T07 vmemon	42	184800 18F000	186C02	-128 0	69 1 W
9	T08 vmexx	40	18F000 199000	19123C	-1 1	69 1 W

pacman:

This is a list of the processes running in the VME processor. The fact that you can get this list is a very good sign that the VME system is working. Note that this list may at times have a different order and that additional processes may be loaded. What is shown above is simply the minimum required.

If the VME processor has gone belly up, the response, after about 15 seconds, would be:

pacman: lt

Connection failure, no acknowledgment - /usr/acq/vme/lt

So what do I do now? Well, try BOOTVME followed by LOADACQ. If these commands seem to work, try LT again. If LT now works, you will need to load the PAC and STARTVME to resume data acquisition.

The next thing to try is manually resetting the FORCE CPU-40. Go to the VME crate and push the Reset switch. The Reset switch is the upper switch on the module front panel. Now at the Workstation do BOOTVME, LOADACQ and LT again. If LT works, you still need to load the PAC and STARTVME to resume data acquisition.

If none of that works, you are in deep dodo. Quit while you're ahead, call

for HELP and find something else to amuse yourself with until HELP arrives.

PS

This shows the processes in the workstation. The response shown below only includes the essential processes. Your list will most likely include many others. Just check to see that the ones listed below are in your list.

```
pacman: ps
  PID TTY          TIME CMD
 9457 pts/2        00:00:00 csh
 9533 pts/2        00:00:00 pacman
 9536 pts/2        00:00:00 pftoipc
 9537 pts/2        00:00:00 tape
 9538 pts/2        00:00:00 xterm
 9539 pts/2        00:00:00 femsg
 9545 pts/2        00:00:00 xterm
 9546 pts/2        00:00:00 xterm
 9599 pts/2        00:00:00 ps
pacman:
```

If any of the processes above is not present, you have a problem. The proper thing to do is KILL ALL in the PACMAN window and run PACMAN again.

STATVME

You may ask for the state of the VME acquisition system with this command. Possible responses are 1) not initialized, 2) running and 3) stopped. This status is reported only in the PACMAN window.

VMEHARDWARE

VMEHARDWARE displays a list of the hardware modules the VME processor found when it was booted. Compare the list of interface modules with what is physically installed in the VME crate. Example output follows:

```
pacman: vmehardware
VME System Hardware Configuration

  VME Processor Logical Name: vme2
VME Processor Ethernet Address: 00-80-42-00-29-79
  Default Host Ethernet Address: 08-00-2b-24-e0-5b
    Boot Multicast Address: 03-6d-63-73-71-00
Available Interface Modules are:

KSC 2917 - CAMAC Interface Module
LRS 1131 - FASTBUS Interface Module
CES 8170 - FERA Readout Module
TRIGGER - ORNL Trigger Module
```

pacman:

PWD

Show PACMAN's working directory. This command can be used to check what directory you are in prior to using an EDIT or PAC command.

DIR

List the files in the current working directory.

PACM.120 Testing VME/Workstation Connection

You may load the VME processor with a code which generates an event stream. Each event has 28 parameters. The IDs are 1 thru 28. Within an event the data for every parameter has the same value. That data value is generated by a random number routine and has a range of 0 thru 1023. The command is

TESTVME

An example using TESTVME follows:

pacman: testvme

***** WARNING *****
* Normal Acquisition Codes in the VME processor are *
* being replaced with test programs. These test *
* programs generate an event stream for testing the *
* VME/Workstation connection. *
* *
* NOTE: When finished testing, your should execute the *
* the command - loadacq *
***** WARNING *****
Delete the acquisition tasks in the VME processor

Now load the VME/Workstation test programs.

VME/Workstation test codes are now loaded.

pacman: initvme

pacman: startvme

FILEOU->rdi

RECORD, # BYTES READ = 1 8192 STAT =GOOD

FILEOU->dev 1,256

pacman: EVENT: Start word = 1 Number of parameters = 28

1 3CD; 2 3CD; 3 3CD; 4 3CD; 5 3CD; 6 3CD;
7 3CD; 8 3CD; 9 3CD; 10 3CD; 11 3CD; 12 3CD;
13 3CD; 14 3CD; 15 3CD; 16 3CD; 17 3CD; 18 3CD;
19 3CD; 20 3CD; 21 3CD; 22 3CD; 23 3CD; 24 3CD;
25 3CD; 26 3CD; 27 3CD; 28 3CD;

EVENT: Start word = 59 Number of parameters = 28

```

  1  29;    2  29;    3  29;    4  29;    5  29;    6  29;
  7  29;    8  29;    9  29;   10  29;   11  29;   12  29;
 13  29;   14  29;   15  29;   16  29;   17  29;   18  29;
 19  29;   20  29;   21  29;   22  29;   23  29;   24  29;
 25  29;   26  29;   27  29;   28  29;
EVENT: Start word = 117 Number of parameters = 28
  1  BC;    2  BC;    3  BC;    4  BC;    5  BC;    6  BC;
  7  BC;    8  BC;    9  BC;   10  BC;   11  BC;   12  BC;
 13  BC;   14  BC;   15  BC;   16  BC;   17  BC;   18  BC;
 19  BC;   20  BC;   21  BC;   22  BC;   23  BC;   24  BC;
 25  BC;   26  BC;   27  BC;   28  BC;
EVENT: Start word = 175 Number of parameters = 28
  1  28C;   2  28C;   3  28C;   4  28C;   5  28C;   6  28C;
  7  28C;   8  28C;   9  28C;  10  28C;  11  28C;  12  28C;
 13  28C;  14  28C;  15  28C;  16  28C;  17  28C;  18  28C;
 19  28C;  20  28C;  21  28C;  22  28C;  23  28C;  24  28C;
 25  28C;  26  28C;  27  28C;  28  28C;
FILEOU->

```

PACM.130 What to Do If PACMAN Refuses to RUN

If you attempt to run PACMAN in a second window or some other user is running PACMAN using the same VME processor, it will refuse your request. You will get a list of the processes and be informed that PACMAN is already running.

```

-----
orph38> pacman
9429 p1 I    0:26 /usr/acq/wks/pacman
9432 p1 S <  1:04 /usr/acq/wks/pftoipc -d vme2
9433 p1 I    0:00 /usr/acq/wks/femsg -d vme2
9434 p1 I <  0:00 /usr/acq/wks/tape vme2
9442 p2 S    0:00 /usr/acq/wks/logger vme2
pacman is already running!!
orph38>

```

It is also possible that all of PACMAN's essential processes do not terminate when you exit PACMAN. In this case, you will get a list of one or more processes which must be terminated before PACMAN can run again.

NOTE: The DAMM and SCAN windows are NOT essential processes. PACMAN never terminates these but does recognize their presence on start up.

The following is an example:

```

-----
orph38> pacman
9434 p1 I    0:00 /usr/acq/wks/tape vme2

```

One or more of pacman's processes are already running
orph38>

In this example, you need to kill process with PID 9434. The first attempt should be:

kill 9434

Now either try to run PACMAN again or use the ps command to see if the process is gone. If the process is still present, the following should terminate it:

kill -9 9434

PACM.140 How to Exit PACMAN

After several hectic days and sleepless nights, the experiment is over and it is now time to shutdown everything, pack and go catch our plane. How do you stop PACMAN? The command is

KILL ALL(this one MUST be upper case).....

If you were writing event data to disk, two End-of-Files(EOF) will be written on the disk file. Then the disk file is positioned between the two EOFs. All of PACMANs resources will be released and all of it's processes are terminated. PACMAN will exit and you will get a system prompt.

There are circumstances when PACMAN will prompt you to use the KILL ALL command and then run PACMAN again. What can I say, it is good advice.

PACM.150 Run-time Help Text - /usr/acq/wks/pacman.hep

\$\$\$\$SCAT - For Scaler-to-File related commands

Commands used to control scaler dumps to the data file -----

SCAT filename ;Initializes SCAT from filename

SCAT ON ;Dumps scalers before EOF only
SCAT ON CLR ;Dumps scalers before EOF & clears post dump
SCAT ON TSEC ;Dumps scalers every TSEC & before EOF
SCAT ON TSEC CLR ;Dumps scalers every TSEC & before EOF & clears
SCAT OFF ;Disables scaler dumps

SCAT NORT ;Normalize count rates to internal clock
SCAT NORS ;Normalize count rates as specified by snit-file
SCAT UNOR ;Count rates un-normalized

SCAT STAT ;Displays/logs setup status
SCAT HELP ;Display on-line help
SCAT H ;Display on-line help

SCAT LERR ON ;Turns error display/logging ON
SCAT LERR OFF ;Turns error display/logging OFF

SCAT ZERR ;Zeros error counters
SCAT ZDUM ;Zeros scaler dump counter
SCAT ZERO ;Zeros all scalers
\$\$\$\$VME - For commands to control and get status of Front-End
COMMANDS TO THE FRONT-END VME PROCESSOR
BOOTVME - Reboot the VME processor
LOADACQ - Download code for the data acquisition system to VME processor
LOADDSSD - Acquisition for the DSSD detector system

SETUP file - Run MODU_SETUP to initialize CAMAC modules.

INITVME [file] - Initialize the acquisition software. MUST be done after PAC.
If a file is specified, MODU_SETUP is run prior to
initializing the VME.

STARTVME - Start acquisition in the VME processor
STOPVME - Stop acquisition in the VME processor
STATVME - Show status of acquisition in VME processor

ZEROCLK - Zero the 100Hz VME clock

VMEHOST - Show workstation which receives event data
PACFILE - SHow name of PAC file

PACOR file 1 - Compile and load your front-end acquisition code.
EDIT file - Edit user front-end acquisition code (vi editor).

CD dir - Change directory to dir. Useful if set to directory of your
xxx.pac files. If so set, you need not specify the directory
path for PAC and EDIT commands.

PWD - Show current working directory.
DIR - List the files in the current working directory.
\$\$\$\$ACQ - Commands related to data acquisition
ACQUISITION COMMANDS
FDIR - Display directory for LDF files
FDIR dir - Set directory for LDF files
OUF file - Write list data to LDF file.

HTIT - Set title for output header generated by FILEOU
HNUM - Set next header number for output header
HOUT - Write locally generated header

EOF - Write EOF on OUTPUT

LOG mess - Insert comments into log file

TRUN beam - To START writing the Acquisition Data Stream to file
beam="boff" (beam off) or "bon" (beam on)
TCONT - To Continue writing the Acquisition Data Stream to file
TSTOP - To STOP writing the Acquisition Data Stream to file

TON beam - Starts file but does not start VME acquisition
beam="boff" (beam off) or "bon" (beam on)

HUP - Histogram update. Useful for very low rate acquisition.

ZBUF - Clear all shared memory buffers.

AFON - Enable automatic file mark after 10000 records (DEFAULT)

AFOF - Disable automatic file mark

\$\$\$\$FILE - For File control commands

FILE CONTROL COMMANDS

FDIR - Display directory for LDF files

FDIR dir - Set directory for LDF files

OUF file - Write list data to LDF file.

RDI N - Read N records from Data Stream

RDO N - Read N records from OUTPUT

FRO N - Forward N records on OUTPUT

BRO N - Backup N records on OUTPUT

FFO N - Forward N files on OUTPUT

BFO N - Backup N files on OUTPUT

RWO - Rewind OUTPUT

BTO - Go to bottom of OUTPUT (to DBL-EOF, backup 1 EOF)

CLO - Close OUTPUT

ULO - Unload and close OUTPUT

AFON - Enable automatic file mark after 10000 records (DEFAULT)

AFOF - Disable automatic file mark

\$\$\$\$DISP - For commands which display Data-records

COMMANDS FOR READING & DISPLAY OF DATA-RECORDS

RDI N - Read N records from Data Stream

RDO N - Read N records from OUTPUT

PEV IA,IB - Print 16-bit word IA thru IB in EVENT format

DEV IA,IB - Display 16-bit word IA thru IB in EVENT format

PZ IA,IB - Print 16-bit word IA thru IB in HEX format

DZ IA,IB - Display 16-bit word IA thru IB in HEX format

PA IA,IB - Print 16-bit word IA thru IB in ASCII format

DA IA,IB - Display 16-bit word IA thru IB in ASCII format

PI IA,IB - Print 16-bit word IA thru IB in INT*2 format

DI IA,IB - Display 16-bit word IA thru IB in INT*2 format

PIF IA,IB - Print 32-bit word IA thru IB in INT*4 format

DIF IA,IB - Display 32-bit word IA thru IB in INT*4 format

\$\$\$\$STAT - Commands to display system status and test VME system

COMMANDS TO SHOW STATUS AND TEST SYSTEM.....

USERCONFIG- Display commands and parameters from the configuration file.

TESTVME - Download code to the VME processor which generates a data stream. This is a fixed pattern of 28 parameters/event. IDs are 0 thru 27. Parameter values are all the same but the value increments after each event (range = 0 thru 4095).

LT - List processes in the VME processor. If this gives no error, the VME processor is alive and kicking.

STATVME - Show status of acquisition in the VME processor.

PS - Lists your processes in the workstation.

PWD - Show current working directory.

DIR - List the files in the current working directory.

VMEHARDWARE - Show hardware installed in the VME crate

\$\$\$MISC - Miscellaneous Commands

MISCELLANEOUS COMMANDS

CMD file - Get PACMAN commands from file.cmd

LOG mess - Write message to acquisition log file

DATE - Show date and time of day

TIME - Same as DATE

IPCS opts - Show status of system IPC resources

LS - Show directory

DIR - Same as LS

PRTDOC - Print PACMAN user document

PWD - Show current working directory

\$\$\$EXIT - How to STOP the acquisition system

HOW TO EXIT PACMAN

KILL ALL - STOP all acquisition. This terminates all acquisition processes in the workstation. To run acquisition again you must run pacman again. If file output is running, two EOFs are written and then the file is positioned between the two EOFs.

WARNING: This command MUST be upper case!

\$\$\$CMDS - Command list

VME commands:

bootvme	cd	edit	initvme
kt	lt	loadacq	loaddssd
pacor	pacfile	setup	startvme
statvme	stopvme	testvme	vi
vmehost	vmehardware	zeroclk	

Help commands:

911	end	exit	newf
h	help	quit	

Miscellaneous commands:

cmd	cmdf	date	dir
hup	ipcs	kill	log
ls	prtdoc	ps	pwd
time	userconfig	zbuf	

File commands:

afof	afon	bfo	bro
bto	clo	da	dev
di	dif	dz	eof
fdir	ffo	fro	hnum
hout	htit	ouf	ulo
pa	pev	pi	pif
psav	pz	rdi	rdo
rwo	scat	tcont	ton
trun	tstop		

\$\$\$\$NEW - How to update Help messages

HOW TO UPDATE YOUR HELP MESSAGES.....

NEWF - Help file may be revised when acquisition is running. This command will replace the current help with the revised version.

PACM.160 Built-in Command List

This section lists all of PACMAN's built-in commands. User defined commands and windows may NOT be identical to any of these commands.

VME commands:

bootvme	cd	edit	initvme
kt	lt	loadacq	loadssd
pacor	pacfile	setup	startvme
statvme	stopvme	testvme	vi
vmehost	vmehardware	zeroclk	

Help commands:

911	end	exit	newf
h	help	quit	

Miscellaneous commands:

cmd	cmdf	date	dir
hup	ipcs	kill	log
ls	prtdoc	ps	pwd
time	userconfig	zbuf	

File commands:

afof	afon	bfo	bro
bto	clo	da	dev
di	dif	dz	eof
fdir	ffo	fro	hnum
hout	htit	ouf	ulo
pa	pev	pi	pif
psav	pz	rdi	rdo
rwo	scat	tcont	ton
trun	tstop		

PACM.170 How to Access On-Line Documentation

Almost all documents covering data acquisition and analysis are available on-line. These are in the following three directories:

- 1).. /usr/hhirf/doc - Data analysis
- 2).. /usr/hhirf/wks/doc - Data acquisition
- 3).. /usr/hhirf/vme/doc - VME acquisition system

The .doc files in these directories are document files and are inputs to txx text formatting code. There are three output formats available.

- 1).. Plain ASCII file - Suitable for viewing with a text editor
- 2).. LN03 file - May be printed on a DEC LN03 printer or compatible
- 3).. Postscript Printer output

Plain ASCII files can be viewed with your favorite editor and printed on almost any printer. To get a plain ASCII file for this document, type

```
/usr/hhirf/txx /usr/hhirf/wks/doc/pacmanII
```

The output file is in your current working directory and is pacmanII.tex. Many documents are already available in this format in the document directories.

To produce a file for a LN03 or compatible printer, type

```
/usr/hhirf/txx /usr/hhirf/wks/doc/pacmanII x
```

The output is file pacmanII.txx which must be queued to the printer to get a hard copy. The output file is in your current working directory.

Finally, if you have a Postscript printer, type

```
/usr/hhirf/dodoc pacmanII
```

The output is queued to the default Postscript printer. However, two temporary files, pacmanII.txx and pacmanII.ps, are created in your current working directory. These files are deleted after the print request is queued.