THE FARM SHIFTER'S GUIDE

by Cathy Cretsinger
University of Rochester

V1.0 – 27 June 1994

OVERVIEW

The purpose of this document is to provide guidance for the (non-expert) production-farm shifter on what to do during shift. The basic jobs of the farm shifter are: (1) looking for bad tapes, hardware failures, and other problems; (2) fixing and/or reporting the problems; (3) submitting special jobs requests when so instructed; and (4) providing weekly reports of farm activity. This document is divided into these sections, dealing with an assortment of events one can expect to see on shift.

In order to keep the focus on "how-to" issues, I have not included a general description of the TPM farm management system. Documentation on that, in varying degrees of detail, exists elsewhere. (See the `USR$ROOT:[DZERO.DOCS]` area on D0FS.) Shifters are advised to learn about the system.

I must emphasize that this is not a UNIX manual, although it does include a few UNIX commands that are essential to monitoring the farm status. Shifters who are not familiar with UNIX should acquire and read manuals on the subject.

Credit goes to Soon Yung Jun, Kirill Denisenko, and Mei Gui for their notes, which I have used extensively in preparing this document. However, responsibility for the content of this document rests with me.

CONTENTS

NOTATION AND FONT CONVENTIONS

I use a variety of fonts to convey different intentions. Filenames, directories, processes, etc. on the farm are set in `Courier`; commands you should type are set in **`Courier (bold)`**; machine names, account names, etc. are set in Helvetica, just to make life more interesting. (These fonts are not visible in the text version.)

Throughout, the notation VM is used to denote a "virtual machine" (a.k.a. "logical machine"—a parallel server process running on an I/O node on the farm); the generic name `fnsfX_#` is used to denote a specific virtual machine. Capital **`X`** is always to be replaced by a (usually lowercase) letter when typing commands; small **`x, N,`** and **`#`** are to be replaced by numerals.

1) LOOKING FOR PROBLEMS

1.1 Routine monitoring

Log into fnsfe, fnsfd, and fnsff (the farm I/O nodes assigned to D0) as dzero and run the diagnostics as indicated. Also, log into the DZERO account on D0FS and read the mail, which will contain special requests, problem reports, inquiries, etc. You should archive the mail in this account, as detailed in section 1.2.

1.1.1. Display farm status

This is a nice way to start because it gives you an overview of what each virtual machine has been doing. You should do it at least twice per day to make sure that tapes are moving through the system. The command is:

`%`**`tpm_disp`** {If you have a 132-character terminal, you can use **`tpm_wide`**, which shows finish as well as start times}

```
ID    NODE    PROJECT        INVOL  OUTVOL DATE            STATUS
________________________________________________________________
12492 fnsfe_0 RECO_FULL_V12 WM5976 PWNODE Mar 16 16:31:10 ENDED
12493 fnsfe_2 RECO_FULL_V12 WM5948 PWNODE Mar 16 17:37:14 ENDED
12494 fnsfe_1 RECO_FULL_V12 WM5977 PWNODE Mar 16 21:34:46 ENDED
12495 fnsfe_0 RECO_FULL_V12 WM5978 PWNODE Mar 16 23:28:53 ENDED
12496 fnsfe_2 RECO_FULL_V12 WM5979 PWNODE Mar 17 02:13:19 ENDED
12497 fnsfe_1 RECO_FULL_V12 WM5980 PWNODE Mar 17 05:23:30 ENDED
12498 fnsfe_0 RECO_FULL_V12 WM5982 PWNODE Mar 17 06:12:49 ENDED
12499 fnsfe_2 RECO_FULL_V12 WM5985 PWNODE Mar 17 08:54:01 ENDED
12500 fnsfe_1 RECO_FULL_V12 WM5986 PWNODE Mar 17 08:57:49 ENDED
12501 fnsfe_0 RECO_FULL_V12 WM5987 PWNODE Mar 17 12:12:40 ENDED
12502 fnsfe_1 RECO_FULL_V12 WM5988 PWNODE Mar 17 13:55:59 ENDED
12503 fnsfe_2 RECO_FULL_V12 WM5989 PWNODE Mar 17 15:24:46 ENDED
12504 fnsfe_0 RECO_FULL_V12 WM5990 PWNODE Mar 17 16:51:20 ENDED
12505 fnsfe_1 RECO_FULL_V12 WM5991 PWNODE Mar 17 18:37:03 ENDED
12506 fnsfe_2 RECO_FULL_V12 WM5992 PWNODE Mar 17 20:41:01 ENDED
12507 fnsfe_2 RECO_FULL_V12 WM5999 PWNODE Mar 17 22:11:16 ENDED
12508 fnsfe_1 RECO_FULL_V12 WM5942 PWNODE Mar 18 16:49:51 ENDED
12509 fnsfe_2 RECO_FULL_V12 WM5982 PWNODE Mar 19 16:39:15 ENDED
12510 fnsfe_1 RECO_FULL_V12 WM5986 PWNODE Mar 19 16:58:26 FAILED_ON_INSP
12511 fnsfe_1 RECO_FULL_V12 WM5942 PWNODE Mar 20 16:40:54 INQUEUED
12512 fnsfe_2 RECO_FULL_V12 WM5986 PWNODE Mar 20 16:57:16 INQUEUED
%
```

This command shows you the status of the farm. (Note that you can use **`tpm_disp #`** to display more lines—`#` specifies how many—if you want to look at less recent jobs.) The `ID` is a sequential number assigned to each job. `NODE` shows which virtual machine (VM) the job ran/is running on. `PROJECT` tells what processing was/is done on the file. `INVOL` is the tape number, `OUTVOL` is always `PWNODE`. `DATE` is obvious. Possible `STATUS` entries for parallel servers (the usual D0 configuration) are:

INQUEUED - Job has been submitted to VM with multiple worker nodes.

ENDED - Inspooling finished, status of processing and output not checked.

FAILED_ON_INSP - Input spooler failed. Check its log file (see section 1.3.1) for more information. For standard RECO (WMxxxx tapes with ALL stream data), the tape will be resubmitted automatically after 48 hours. It is usually a good idea to wait for this, since the procedure that resubmits the tape also updates the corresponding `tapes_moved` file to remove any files that were processed successfully before the inspooler failed; this cuts down on reprocessing of data. For special jobs, you must update the `injobs` file manually (edit the file, changing status from `SUBMITTED` to `WAITING` for that tape).

INTAPE_IN_USE - Some other job is holding the input tape, which causes the inspooler to fail. You should investigate and, if appropriate, release the tape (see section 2.3).

There are other status flags that will appear in sequential or stand-alone modes, which are normally not used. For more information on those, refer to Kirill Denisenko's "TPM Production Manager Operator's Manual and User's Guide."

One thing to look for here is abandoned jobs in the `INQUEUED` state. You can tell they have been abandoned because the VM has gone on to the next tape (there is another tape `INQUEUED` to that VM at a later time) but the status for the older tape never changed to any of the end conditions. If there are abandoned jobs, this may be a sign of a problem with the RECO code, or simply a side effect of errors, crashes, etc. See Section 2.8 for more on this issue.

1.1.2. Check farm configuration

The relevant files are kept in the `~/proman/resources` directory on fnsfe. (To get there, you can use the command **`res`** on any of the I/O nodes.) These files show which VMs are available, which worker nodes are assigned to each VM, and how the inspoolers are distributed. Look at them when you want to check on how the farm is presently configured. Here are a few useful files:

- *The main resource file*:

% **re** (or **tpm_rsrc**) {types the resource file}

```
fnsfe_0 FARMLET RECO_FULL_V12 4
fnsfe_1 FARMLET RECO_FULL_V12 5
fnsfe_2 FARMLET RECO_FULL_V12 6
fnsfd_0 UNAVAILABLE RECO_FULL_V12 1
fnsfd_1 UNAVAILABLE RECO_FULL_V12 3
fnsfd_2 UNAVAILABLE RECO_FULL_V12 2
%
```

This file (`~/proman/resources/resource`) shows what VMs are defined, the status (`FARMLET` if the VM is able to process data, `UNAVAILABLE` otherwise), what project it is presently assigned to run, and which inspooler process the VM is using (designated by the number at the end of each line).

You will need to edit the resource file in order to:

-change the status of VMs;

-change the project assigned to one or more VMs;

-combine inspoolers

More on these operations (e.g., when to do them) appears in various sections below.

Note: When you edit the control files (especially the injobs file), it is best to copy the file to a dummy file, edit the dummy version, then copy or rename the dummy to the correct file name when you are satisfied with the changes. This is recommended because UNIX does not save old versions, which makes it very easy to overwrite a good file with junk.

- *Inspooler definition files*:

The number representing the inspooler in the resource file refers to a file in `~/pro-man/resources` called `inspool_list_N.fnsfX`, which contains the name(s) of a spooling disk(s) to be used by the VM(s) to which the number N is assigned in the resource file. These spooling disks are not assigned to any particular tape drives: when an inspooler is ready for a new tape, it grabs any available drive. Note: if tape drives are broken, you may have more than one VM assigned to the same inspooler. This is acceptable, provided that both VMs are running the same job. (See section 2.2 for more on assigning inspoolers.)

- *Worker node assignment files*:

Detailed information about which worker nodes are in use is contained in a number of files called `resource.fnsfX_#`. (These are also in the resource directory.) Each such file contains a list of worker nodes assigned to that VM and their status. If you should ever have to disable sending events to a broken node, this is the file you would edit to do that, changing the node's status from `READY` to `UNAVAILABLE`.

- *Tape drive files*:

These files identify the drives assigned to each physical machine (fnsfe, fnsfd, or fnsff). They are described in section 1.1.6.

- *Injobs file:*

This file contains the list of input tapes to be processed. It is discussed in more detail in section 1.1.4.

- *Blank tape file*:

The file `blanks` lists the existing output tapes and indicates whether they are `USED` or `READY` to be written on. Tapes are added to this file by the operators in Feynman Computing Center as they are initialized; shifters rarely need to edit this file.

1.1.3. Check output files

You should check the output on each (working) virtual machine every two to three hours; this is the best way to make sure that data is flowing through the farm. Log in to each I/O node and use the following commands:

`%` **`sp0`** (or **`sp1`** or **`sp2`**) {goes to the VM's spool directory—for use on fnsfe}
(or **`spd0, spd1, spd2`** — for use on fnsfd

or **spf0, spf1, spf2** - for use on fnsff.)

% **lps** {directory of output files, shorthand for **ls -l proman/sta/**}

```
total 1326911
-rw-r--r-- 1 dzero e740       120 Mar 22 13:01 12521_010_076089_10.ous
-rw-r--r-- 1 dzero e740       144 Mar 22 13:09 12524_003_076090_21.ous
-rw-r--r-- 1 dzero e740 275970240 Mar 22 13:09 12524_003_076090_21.sta
-rw-r--r-- 1 dzero e740 111252960 Mar 22 13:20 12524_004_076090_22.sta
-rw-r--r-- 1 dzero e740 292153680 Mar 22 13:01 ALL_076089_10.X_STA01REU
1210_ALL00_NONEX00_4032213
%
```

This shows you the status of the output STA files. If you type **lps** again after a few minutes, you should see the file size increasing on all `*.sta` files that are not paired with `*.ous` files. This tells you that the jobs are running and producing output. The `*.ios` and `*.ous` files you will sometimes see paired with the `*.sta` files are normal; their functions are described in Kirill Denisenko's "TPM Production Manager Operator's Manual and User's Guide."

There are a number of things that can cause the file sizes to freeze for a while, most of which are perfectly normal and require no action on your part except continued vigilance. What follows are some things you should notice, with some hints on how to tell if there is a problem. This should not be regarded as an exhaustive list, merely some starting points.

1.) If the response to **lps** does not change with time, look for `*.done` files paired with some of the files. When this appears, the parallel server process (`d0reco_prl_fnsfX_#`) probably needs to be restarted. The system is designed to do this automatically and send email confirming that it happened, so the first thing to do is to leave things alone and check later. If things hang up for a long time (hours), you might consider killing the parallel server yourself, especially if you have not seen any mail messages indicating that the process was restarted. Instructions for this are in section 2.7.

2.) If you see "total 0" in response to **lps**, check that the `inspool_fnsfX_N` process is running. (Remember that N here corresponds to the number assigned to each VM in the `resource` file, not to the VM number; see section 1.1.2.) *If the inspooler is **not** running*, check for mail messages complaining that no tape drives are available: you may need to combine inspoolers due to broken drives. (Tape drive diagnostics are covered in section 1.1.6; what to do with a broken drive is covered in sections 2.1 and 2.2.) If enough drives are available, make sure that the project assigned to the VM in the resource file has tapes waiting in `injobs`. If all these things are correct and the inspooler does not start for over half an hour, call an expert. *If the inspooler is running*, check the subdirectories `inspool/` and `raw/` to see whether new raw data files are being read in. If they are not, check that the proper input tape has been mounted (see section 1.1.6). You are likely to see "total 0" if the time stamp on the last `INQUEUED` job on the VM in question is quite recent; this is because a full file has to be read in before any processing can start. If the "total 0" has remained for a long time and all processes are running for that VM (see section 1.1.5), you should suspect RECO problems. Check on the worker nodes (**rlogin** to one or two of them and check that the appropriate processes are running as described in section 1.1.5). If they appear to be okay, check the log files in the `wrkdb` area (see section 1.3.3) for RECO error messages.

3.) Occasionally, you may see `.problem` files in the area, as in the following example:

```
sfe/spool00/dzero/fnsfe_0% lps
total 2517150
-rw-r--r-- 1 dzero e740 257755680 Mar 28 03:49 ALL_076151_28.
X_STA01REU1210_ALL00_NONEX00_4032804.problem
-rw-r--r-- 1 dzero e740 257755680 Mar 28 05:35 ALL_076151_28.
X_STA01REU1210_ALL00_NONEX00_4032805.problem
-rw-r--r-- 1 dzero e740 257722920 Mar 28 06:49 ALL_076151_28.
X_STA01REU1210_ALL00_NONEX00_4032806.problem
-rw-r--r-- 1 dzero e740 257755680 Mar 28 09:20 ALL_076151_28.
X_STA01REU1210_ALL00_NONEX00_4032809.problem
-rw-r--r-- 1 dzero e740 257788440 Mar 28 12:19 ALL_076151_28.
X_STA01REU1210_ALL00_NONEX00_4032812.problem
```

These are leftovers from an old attempt to diagnoze ZEBRA problems in the files being processed. They do not affect the quality of the output, nor do they mean that a partition (file) has been lost, but they can cause the VM to hang up. If this happens, you should delete all files with names ending in `.problem`; you should see processing resume fairly quickly after that.

1.1.4. Look at injobs file

You should do this every day and make a note of the last tape waiting; this is how you can make sure that tapes are being added. The injobs file is `~/proman/resources/injobs.` A shorthand command has been defined:

```
% in {shows tapes with status WAITING, equivalent to
        more injobs | grep WAITING}
WM5342 RECO_L0V_V12 WAITING
VGL063 D0GEANT_SGI_V11SHL WAITING 4
VG0661 D0GEANT_SGI_V11 WAITING 8
WM3233 RECO_FULL_V11 WAITING 1
WM3238 RECO_FULL_V11 WAITING 1
WM3420 RECO_FULL_V11 WAITING 1
%
```

You can also type the file, by first going to the resource directory (**res**), then typing:

```
% more injobs   {types the complete file}
```

```
[many lines deleted]
WM6010 EXPRESS_SGI PENDING
WM6011 RECO_FULL_V12 SUBMITTED
WM6013 RECO_FULL_V12 SUBMITTED
WM6014 RECO_FULL_V12 SUBMITTED
WM6015 RECO_FULL_V12 SUBMITTED
WM6016 RECO_FULL_V12 SUBMITTED
WM6017 RECO_FULL_V12 WAITING
WM6019 RECO_FULL_V12 WAITING
WM6020 RECO_FULL_V12 WAITING
```

```
WM6024 RECO_FULL_V12 WAITING
WM6025 RECO_FULL_V12 WAITING
WM6026 EXPRESS_SGI PENDING
WM6027 RECO_FULL_V12 WAITING
WM6028 RECO_FULL_V12 WAITING
WM6037 RECO_FULL_V12 WAITING
[more lines deleted]
%
```

You can also use **more injobs | grep** <tapename> to check the status of a particular tape.

As the examples show, injobs has the format <tape> <project> <status> <number of failures>. This last shows up only if it is larger than zero; it indicates how many times an inspooler has tried and failed to read the tape.

The status choices are SUBMITTED, WAITING, PENDING and BADTAPE. The first two are common. The job stays in WAITING until it is first in line and a VM is set up to run the job <project>; then it is SUBMITTED. BADTAPE is sometimes set automatically, but you should do it manually (see section 2.4) if you notice a large number of retries ($\geq$ 3) for some tape. PENDING is set manually (by you) when a tape is good, but for some reason (e.g., problems with the RECO code) the job needs to be held up. When you need to change the status of a tape, do so by editing injobs.

There is an automated process that adds new data tapes to be processed with the current version of RECO to the bottom of the injobs file. Another process removes old tapes from the file, presumably some weeks after they have been successfully processed and their output files properly disposed of and catalogued. The result is that this file is usually quite large. (If you see no tapes WAITING during the run, you should be highly suspicious. See section 2.8.)

Tapes processed for special requests (e.g. Monte Carlo, older or non-standard versions of RECO) are added to the injobs file manually (by you — see section 3 for instructions on special jobs); WM tapes for regular reconstruction are added automatically.

The injobs file is a queue: tapes are put in at the bottom and read out off the top. Therefore, you should add tapes to the end of the file, not to the beginning. You should only move tapes to the beginning if OCPB has granted certain runs high-priority status; Lee Lueking will notify you when this happens.

Note: When you edit this file, it is wisest to copy it to a dummy file first and edit the dummy file; copy the dummy to injobs only when you are satisfied that your changes are correct and that you have not deleted any tapes that should remain. (If injobs is overwritten with junk, the amount of work required to reconstruct it is large, so this is important.)

1.1.5. Checking whether processes are running

When data is not going through a VM, it is useful to check whether the relevant processes are running. The list below shows you one way to check for these processes. Note that you must be logged into the I/O node you want to investigate or you will not see the processes. (For explanations of what they do, see Kirill Denisenko's manual.)

% **pf tpm** {on fnsfe}

```
 dzero 5550 1 0 Jun 16 ? 3:46 tpm_submit_job tpm_submit_job
dzero 22309 21019 0 17:21:13 ?  0:00 /usr/local/home/dzero
/proman/tpm/check_rsh.csh -f /usr/local/home/dzero/proman/
dzero 7548 1 0 11:38:42 ?       2:20 /usr/local/home/dzero
/proman/tpm/d0reco_prl_fnsfe_0 /usr/local/home/dzero/proma
dzero 18861 1 0 17:04:44 ?      0:11 /usr/local/home/dzero
/proman/tpm/d0reco_prl_fnsfe_1 /usr/local/home/dzero/proma
dzero 12003 1 0 12:07:41 ?      2:29 /usr/local/home/dzero
/proman/tpm/d0reco_prl_fnsfe_2 /usr/local/home/dzero/proma
```

This shows you that `tpm_submit_job`, the master control process, is running. You also see the three parallel server processes for the VMs on fnsfe and any active child processes of `tpm_submit_job` (in this case, `check_rsh.csh`). If `tpm_submit_job` is not running, you can expect to receive mail (see section 1.2.7).

% **pf prl**

```
dzero 11422 1 0 07:50:32 ? 0:59     /usr/local/home/dzero/proman
/tpm/d0reco_prl_fnsfe_1        /usr/local/home/dzero/proma
dzero 24160 1 0 17:07:38 ? 7:51     /usr/local/home/dzero/proman
/tpm/d0reco_prl_fnsfe_2        /usr/local/home/dzero/proma
dzero 5198 1 1 03:03:49 ? 2:25      /usr/local/home/dzero/proman
/tpm/d0reco_prl_fnsfe_0        /usr/local/home/dzero/proma
%
```

This shows you that `d0reco_prl_fnsfX_#` (the parallel server process) is runninng for each VM currently in use. If you should need to kill any of these processes, you can get the process ID from this display (e.g., for `d0reco_prl_fnsfe_1`, the ID is `11422`).

% **pf inspool**

```
 dzero 24494 1 80 09:42:16 ?    11:16      /usr/local/home/dzero
/proman/exe/inspool_fnsfe_4 -inl WM6127 -dev sgi84 -queid
 dzero 6057 1 73 06:56:36 ?     17:19      /usr/local/home/dzero
/proman/exe/inspool_fnsfe_6 -inl WM6240 -dev sgi85 -queid
 dzero 7787 1 57 11:27:23 ?     3:52       /usr/local/home/dzero
/proman/exe/inspool_fnsfe_5 -inl WM6128 -dev sgi80 -queid
%
```

This shows the inspool processes, which copy files from raw data tapes to the assigned spooling areas. Note that the number following `fnsfX` (4, 6, or 5 in the above example) does not refer to the VM, but to the inspool area assigned to that VM in the resource file. (See section 1.1.2.)

```
% pf insrv
 dzero 14076 5198 70 03:59:03 ?         55:43 insrv_d0_fnsfe_0
 dzero 12406 11422 39 08:00:54 ?        27:01 insrv_d0_fnsfe_1
 dzero 12752 24160 55 01:17:52 ?        64:41 insrv_d0_fnsfe_2
%
```

This shows the insrv process running on each VM; the last part of the name tells you the VM. These processes serve raw events to the worker nodes for reconstruction. You may see multiple `insrv` processes for one VM; this is not a problem.

```
% pf outsrv
 dzero 12407 11422 80 08:00:54 ?        24:48 outsrv_d0_fnsfe_1
 dzero 14077  5198 71 03:59:03 ?        49:03 outsrv_d0_fnsfe_0
 dzero 12753 24160 76 01:17:52 ?        65:50 outsrv_d0_fnsfe_2
%
```

This shows the outsrv process running on each VM. These process pick up reconstructed events from the worker nodes and add them to the output files on the spooling disk (in the `proman/dst` and `proman/sta` subdirectories). You may see multiple `outsrv` processes for one VM; this is not a problem.

```
% pf outspool
 dzero 20778 1 0 09:12:07 ? 5:43    /usr/local/home/dzero/proman
/exe/outspool_fnsfe_2 -inl WN7814 -dev sgi83 -path
 dzero 2922 1 0 22:23:14 ? 22:21    /usr/local/home/dzero/proman
/exe/outspool_fnsfe_1 -inl WN7813 -dev sgi81 -path
 dzero 14235 1 78 19:33:14 ? 25:01  /usr/local/home/dzero/proman
/exe/outspool_fnsfe_0 -inl WN7811 -dev sgi82 -path
%
```

This shows you the outspooler processes, including which tape drive is used by which VM, and what tape is being written.

If any of the above processes (except `tpm_submit_job)` are missing, you can restart them by executing the script `d0reco_prl.csh.` It is a good idea to wait and watch for a while before doing that, since they usually get restarted automatically (usually within about 15 minutes, when the production manager process notices that they are dead).

On a ***worker node***, the key processes to look for (using **`rlogin`** to get to the node and **`pf`** to look for the processes) are: `wrkshell, d0reco.x` (there might be a different name for special jobs), `inreader.x`, and `outwriter.x`. There may be multiple `inreader` and/or `outwriter` processes; this is no problem, but multiple `wrkshell` or `d0reco` indicate a problem.

1.1.6. Check tape drive status

You should do this at least once per day, and note the results in the production logbook, so that we can see if errors are mounting quickly on some drive. This will help to flag failing drives.

Look at file `drives.fnsfX` in the directory `~/proman/resources`.

```
% more drives.fnsfe
sgi80 READY 0 27
sgi81 READY 0 40
sgi82 READY 0 45
sgi83 READY 0 6
sgi84 READY 0 13
sgi85 READY 0 6
sgi86 READY 0 1
%
```

The third column gives the current error count on this drive; the fourth is a cumulative total. If a problem is indicated (by the error count reaching 5 or by frequent ABEND errors on that drive in email messages), follow the tape-drive repair procedures below. Drives that reach 5 errors will be set to `UNAVAILABLE` in this file automatically. (Incidentally, drives on fnsfd have names beginning with `sts` rather than `sgi`, but all the procedures are the same.)

Once per day, copy each `drives.fnsfX` file to the corresponding `sk_drives.fnsfX` file. This backup file can be used to replace the drive file if it should be overwritten by blanks.

Current tape drive use can be determined using **`cps_tape -lt`**. (You must do this from the I/O node whose drives you are interested in.) The response to this command looks like:

```
TAPEDRIVE DEVTYPE    STATUS ALLOC       TAPE   TAPE_STATUS
sgi80 exabyte_8500 working allocated   WM6128 mount pending
sgi81 exabyte_8500 working allocated   WN7813 mounted
sgi82 exabyte_8500 working unallocated WN7811 mounted
sgi83 exabyte_8200 working allocated   WN7814 mounted
sgi84 exabyte_8500 working allocated   WM6127 mounted
sgi85 exabyte_8500 working allocated   WM6240 mounted
sgi86 exabyte_8500 working unallocated WM3420 mounted
```

If a drive is not working, it will have status `broken` in this list.

One thing to watch for here is a drive that stays in `mount pending` for over half an hour. If you see this, you should call the operators and ask them to check on it.

1.1.7. Check disk space

The farm will send mail messages when the disk in a spooling area is over 89% full; this is an important warning since processing will stop if the spooling disk is full. In addition, everything will stop if the user area is over 97% full. So, checking the disk space is useful, especially if something is slow or stopped. You can look at disk usage with the UNIX command **`df`**.

`%` **`df`** {shows disk use}

```
Filesystem          Type  blocks     use   avail %use Mounted on
/dev/root            efs   31430   21627    9803 69%  /
```

```
/dev/dsk/lv2         efs 7603592 5043005 2560587 66% /spool02
/dev/dsk/lv1         efs 7603592 5435444 2168148 71% /spool01
/dev/dsk/lv0         efs 7603592 5373084 2230508 71% /spool00
/dev/usr             efs  537327 501613    35714 93% /usr
/dev/dsk/dks0d1s7    efs 1642725 299369  1343356 18% /usr/people
fnsfg:/usr/people/dzero nfs 3800576 3053056 747520 80% /dzero
fnsfg:/dbl3          nfs 7603712 3033600 4570112 40% /dbl3
d0tsar:d0tsar$data1  nfs 15617024 1342464 14274560 9%
                                    /d0fs/disks/d0tsar_data1
fnsfg:/proman        nfs 3801600 2466304 1335296 65%  /proman
```

In this example, you see that the spooling areas (`/spoolXX` in the right-hand column) are well below 89% in use. Note that the only spooling areas shown are those on the I/O node you are logged into.

You should also pay attention to the disk mounts, particularly the nfs-mounted disks `/dbl3`, `/proman`, and `/usr/people/dzero`. If these are not mounted on an I/O node, its VMs will be unable to process data. (Likewise, a worker node where these disks are not mounted will be unable to process data.)

If you see that some area is getting quite full, you may want to track down how the space is being used. A useful tool for this task is the UNIX command **du**.

`%` **`du`** {shows size of all subdirectories of current area}

```
2        ./fnsf112/proman/inspool
1        ./fnsf112/proman/raw
4        ./fnsf112/proman
408      ./fnsf112/run
413      ./fnsf112
2        ./fnsf113/proman/inspool
1        ./fnsf113/proman/raw
4        ./fnsf113/proman
332      ./fnsf113/run
337      ./fnsf113
```

[lines deleted]

```
68145    ./summaries
1738303 ./histograms
78579    ./inspool
724      ./tmp
1497886 ./raw
707976   ./proman/dst
649707   ./proman/sta
3965     ./proman/linktemps
1361649 ./proman
```

[lines deleted]

```
4757416 .
```

This command shows the size of each subdirectory of the current directory. It is useful for tracking down files that have run amok and are eating disk space. The last line is a total size for the current directory and all subdirectories.

1.1.8. Check D0FS buffer area

Once per day, from the DZERO account on D0FS, set default to `D0$DATA$BUFFER` and do a directory. Look for DST files not paired with RCPs, or RCPs not paired with DSTs. If you find any, do the following:

1) If you have DSTs without RCPs that are over a day or two old, find out what output tapes have the corresponding STAs. Use the timestamp in the DST file name to select the correct subdirectory of `~/proman/pdbkd` (see section 1.3.2), then search the `STAT_*.RCP` files in that subdirectory for the run and part number you need. Then look at the `STAT_WNyyyy.RCP` files found in the search to determine whether all files (or only some) on the tape are missing their RCPs.

If all files on an output tape lack RCPs, then do **`cd ~/proman/report`** on the farm and execute **`conv_prepare_rcp WNxxxx <dir> ~/proman`**. Here, `<dir>` should be replaced by the full pathname of the subdirectory of `~/proman/pdbkd` in which the STAT RCP file for the tape you need resides. The last argument specifies a “linktemps” area to hold the RCPs until they are ready to be transferred; always use `~/proman` for this. The RCPs will be generated and transferred to D0FS automatically; check later to confirm that the DSTs have moved out of the buffer area.

If only a few files from a tape need RCPs, execute **`conv_prepare_rcp_single`** instead, with the arguments as indicated above. The RCPs will appear in the directory `~/proman/linktemps`; you should transfer the needed RCPs by hand when they are there (this takes some time). Transfer the DST RCPs to the buffer area on D0FS, and send the STA RCPs to `USR$ROOT:[FMD0.RCPFARMIN]` on D0FS. Finally, send both the DST and STA RCPs to the PROD_DB area on D0FS: for data, the directory is `USR$ROOT22:[PROD_DB.PROD_DB.SGIDATA]`, for Monte Carlo, use `USR$ROOT22:[PROD_DB.PROD_DB.SGIDATA.MC]`. You should also try this method when you have a full tape for which **`conv_prepare_rcp`** failed.

2) If you have RCPs without DSTs, check the log files on the farm in `~/proman/report/log`. Again, you will need to know the output tape for the corresponding STA. Then look at the log files. (Use **`ls *.WNxxxx`** to find the log files.) In particular, the file `copied.WNxxxx` is the script for the zftp transfer of the DSTs, which you can use to find out which directory had the DST before the transfer (it’s also a nice example of how to do zftp transfers), and `copy_log.WNxxxx` is the output from the transfer, which you can check for errors. Check in the directory given in `copied.WNxxxx` to see if the DST is still there. If the file is there, zftp it manually to the buffer on D0FS; if not, delete the RCP from the buffer area. The DST must be in lower-case letters or zftp will not work. If the file is only in uppercase, rename it to lowercase before starting zftp.

Note: The above procedures for handling failed transfers were provided by Mark Galli (FNALV::MGALLI). Please direct questions about them to him.

You should pay particular attention to files in `D0$DATA$BUFFER` with names starting with `WG_<code>_`, where `<code>` is one of the following:

- 'T' if the target directory specified in the RCP is an invalid choice,
- 'N' if the file is an RCP whose corresponding DST never got to the buffer,
- 'E' if the file already exists in the target directory,
- 'F' if the RCP format is bad.

Codes 'D' (for done) and 'C' (for copy) are dealt with by others; you may ignore such files. For the codes listed above, investigate the problem identified by the code, and if you solve the problem, rename the file to remove the `WG_<code>_` prefix.

If duplicates of a file are produced, you will receive email on D0FS::DZERO. Investigate and reply with an indication of which files should be removed (usually the older version).

1.1.9. Monitor CPU activity

You can create a running display of CPU activity and other information from the farm nodes (I/O and worker nodes) on any terminal that runs X-windows and supports TCP/IP network protocol. This is not necessary, but it can be helpful. To start the display, log into fnsfe or fnsfd and type:

```
% setenv DISPLAY <node>:0   {<node> is the name of your X-windows terminal}
% cps_xpsmon -p <prodname>
```

with `<prodname>` chosen to be `sgid0sfX` to get a display for I/O node fnsfX.

It will take a few minutes for the monitor to pop up in a separate window. Once it does, you can control it from that window. The colors are frightening, but the information can be helpful.

1.2. Email (automatic notification of problems)

You will be notified of various conditions via automatic email from the farm. Responsibility for fixing these problems (or notifying an expert where appropriate) rests with the shifters.

The mail messages are sent to unix_proman, an emailing list defined in a file named `.mailrc` and kept in the home directory on fnsfe. You can update the list by editing that file. Please leave dzero@d0fs on the list at all times. You should have your name (at an account you log into regularly) on the list while you are on shift; you may take it off when you are not on shift. In order to minimize confusion, do not add or delete names other than your own unless the person in question has asked you to.

You should archive these and other messages in the MAIL folders of the DZERO account on D0FS. Here is the current list of folders, with an explanation of what goes in them:

FARM_YYMMDD - Messages generated by the farm, or other messages directly related to farm status. The folder dated MMDD runs from Monday (MM/DD at 0:00) through

Sunday (MM/DD+6 at 23:59). Please use the sender's time stamp, not the recipient's, to determine which week messages belong to.

SPECIAL - Any message related to a special project: requests, questions, status messages you send out or receive, etc.

TAPE_PROBLEMS - Messages related to investigation of problems involving unreadable, overwritten, or useless input or output tapes.

HINTS - Messages explaining how to do things on the farm, or copies of messages requesting hints on handling a problem. If a hint is of general interest, you may extract it to a file in the `[.DOCS]` subdirectory of the DOFS::DZERO account. (When doing this, be sure the file name is descriptive and contains the date of the message.)

MINUTES - Any meeting minutes that get sent to DOFS::DZERO.

DOUBLE_ENTRIES - Messages identifying files with two entries in the FATMEN catalog. You should respond with some indication of which ones should be deleted (see section 1.1.8). Keep the messages and any response you make to them in this folder.

WRONG_FILES - Messages containing lists of `WG*` files in the buffer area on D0FS. Messages listing DST files in the buffer without RCP files go here also. Investigate these files as described in section 1.1.8.

Messages that fit none of the above categories may be left in the **MAIL** folder, unless they are clearly irrelevant (e.g., cps/ocs updates for machines not used by the RECO farm), in which case they may be deleted.

Here are some mail messages generated by the farm (with examples in some cases), and what to do when you get them.

1.2.1. P-server(s)

```
Date:
From: dzero@fnsfe.fnal.gov ( Dzero)
To: dzero@d0fs
Subject: P-server(s)

Parallel servers on fnsfe_2 will be restarted
```

This is probably the most common messagae. It signals that the parallel servers on one of the VMs were restarted. It can be ignored unless it happens too frequently, so you should archive these on d0fs and pay attention to the frequency. There is a likely problem if you see repeated messages from one VM (`fnfsX_#`) with little or no processing going on. If you are concerned, check the flow of files in the spooling area (see section 1.1.3). You can also use **tpm_disp** to see how long the VM has been running the current job.

1.2.2. Intape ABEND

```
Date:
From: dzero@fnsfe.fnal.gov ( Dzero)
```

```
To: dzero@d0fs
Subject: Intape Abend
```

```
The previous intape job ended ABEND 12512 Tape WM5986 -1 files
1869 sgi84 fnsfe_2 Mon Mar 21 16:45:27 CST 1994
```

This message signifies failure to mount an input tape for reading (`-1 files`), or failure during read of one of the files on the tape (file count > 0). Check whether the tape has already been resubmitted. If it is a special-project tape, you will probably have to resubmit it by changing its status from `SUBMITTED` to `WAITING` in the `injobs` file. For standard RECO, there is a procedure that automatically resubmits the tape after 48 hours. You only need to resubmit these tapes manually if you are in a hurry to get this particular data processed or if the farm has nothing else to do. In any case, add this tape to the list in the weekly report (see section 4) and archive the message.

You should watch for 3 consecutive failures on the same tape drive: that signifies a failing drive.

If a particular input tape has failed many times, you may want to declare it a bad tape. (See section 2.4.)

1.2.3. Outtape ABEND

```
Date:
From: dzero@fnsfe.fnal.gov ( Dzero)
To: dzero@d0fs
Subject: Outtape Abend
```

```
The previous outtape job ended ABEND Tape WN7750 2 files
/usr/local/home/dzero/proman/pdbkd/1994/mar/19-24 sgi81 fnsfe_1
Thu Mar 24 14:42:50 CST 1994
```

This signifies failure to mount an output tape for writing (file count 0) or a failure during writing (file count > 0). You should not have to do anything to the farm, but again you should watch for 3 consecutive failures on the same tape drive: that signifies a failing drive. Add this tape to the list in the weekly report and archive the message.

1.2.4. Bad file

```
Date:
From: dzero@fnsfe.fnal.gov ( Dzero)
To: dzero@d0fs
Subject: Bad file
```

```
Bad file ALL_062508_71.X_RAW01 on WM3233
```

This message tells you that the inspooler has skipped a file due to read errors. Add this file to the list in the weekly report, and archive the message.

Be aware that for `WMxxxx` tapes, the control program will try to resubmit these bad files since it knows they're missing. If the experts tell you that an attempt to recover some particular bad file is hopeless, you should remove the file name from whichever `tapes_moved.<tape>` file in the `~/proman/history` area has it.

1.2.5. Tape in use on D0FS

```
Date: Wed, 6 Apr 94 10:54:15 -0500
From: dzero@fnsfe.fnal.gov ( Dzero)
To: dzero@d0fs
Subject: Tape in use on D0FS
```

```
Tape VGL127 is in use on the D0FS
```

This message occurs when a tape that is called for in `injobs` can't be mounted because some other process is using it. You will most likely see these tapes appear in **`tpm_disp`** with `IN-TAPE_IN_USE` status. Follow the procedure in section 2.4 for releasing tapes from D0FS.

1.2.6. DST transfer to D0FS failed for WNxxxx

```
Date: Tue, 5 Apr 94 16:12:50 -0500
From: dzero@fnsfe.fnal.gov ( Dzero)
To:  dzero@d0fs
Subject:
```

```
DST transfer to D0FS failed for WN7857 Tue Apr 5 16:12:46 CDT
1994
```

Normal operation calls for DST output to be written to a "linktemps" area on the farm and then transferred to D0FS. Sometimes this fails due to D0FS, network screwups, or other problems. If the time stamp on the message is between 00:00 and 00:20, do nothing and the problem should fix itself. Otherwise, go to `~/proman/exe` directory and execute **`pick_up_all`** (or **`pick_up_trans`**, although that works only on the node you're actually logged into, where the other does all nodes).

The system will retry the failed transfers twice a day (at 5:00 and 20:00).

1.2.7. after_crash executed

This is a message you get after the system reboots. It signifies that the automatic procedure (which is called `after_crash`) that restarts `tpm_submit_job` has been executed. A second message should follow, confirming that `tpm_submit_job` has started. If you get many of these messages in a row, check to make sure `tpm_submit_job` is actually running. If it isn't, or if the `d0reco_prl` do not start, go to `~/proman/exe` and execute **`after_crash`** manually. In any case, archive the message.

1.2.8. No tape drive

```
Date: Wed, 6 Apr 94 17:19:54 -0500
From: dzero@fnsfe.fnal.gov ( Dzero)
```

```
To: dzero@d0fs
Subject: No tapedrive

All drives are busy; fatal error submitting the spooler; page D0
fnsfe_2 Wed Apr  6 17:19:46 CDT 1994
```

You get this message when an inspooler or outspooler needs to mount a tape, but there is no drive available. The most common cause is a bad tape drive, so check the appropriate `drives.fnsfX` file. Archive these messages.

This message has also been seen recently when the `drives.fnsfd` file has been overwritten by a blank file of zero lines. This problem is not yet understood, but you can tell when it has happened if you go to the resource directory and type (**`more`**) the file — you will see nothing at all. The short-term fix is, from fnsfe:

```
% res
% cp sk_drives.fnsfd drives.fnsfd
```

This replaces the blank file with one that has the needed drive information so that processing can continue.

1.2.9. Bad tape drive

```
Date: Wed, 6 Apr 94 17:19:29 -0500
From: dzero@fnsfe.fnal.gov ( Dzero)
To: dzero@d0fs
Subject: Bad Tape Drives

The following drives have their error counts exceeded: sgi83;
page D0
```

You get this message when the error count in `drives.fnsfX` is 5 or more on a particular drive. Follow the procedure in section 2.1 for handling tape drive failures. Be sure to archive this message in `FARM_YYMM` folder.

1.2.10. Wrong project; no current project defined

These signify an improper definition of a project. The problem could be in the `injobs` file, in the `resource` file, or in the `~/proman/project` area where project definitions are specified. It can also happen on the worker nodes. You should check the projects requested in the `injobs` and `resource` files to make sure that they actually exist (compare them to the names on the files in the `~/proman/project` area). Also make sure that there are tapes waiting in `injobs` for every project to which a VM is assigned. If these look okay, wait. The system has many automatic recovery features that may solve the problem. Normal processing should resume on the affected VM and/or worker node(s) shortly.

If the problem persists, call in an expert (S. Kunori) to sort out the problem. Shifters should not edit project files or create new projects without explicit instruction to do so.

1.2.11. Server node fnsfX_# down - investigate

Usually, this represents a network glitch. Check the spooling area (**`sp#`**) to see if the STA files are growing. If they are, there is no real problem. If this message appears frequently, it is likely to be a symptom of a major network problem. If you are suspicious, contact the expert (S. Kunori) and ask him to take a look. Report any other symptoms you see.

1.2.12. Disk space limit exceeded

```
Date: Wed, 6 Apr 94 06:00:17 -0500
From: dzero@fnsfe.fnal.gov ( Dzero)
To: dzero@d0fs
Subject: Disk space limit exceeded
```

```
%90 of the disk space is used on /spool02 of the farmlet
fnsfe_2; Check it. Also  check mounts - maybe Xoper is frozen;
otherwise page D0 primary.
```

This message signifies that too much space is in use in the spooling area named. Processing will slowly grind to a halt unless the disk use falls below 90% used. See Section 2.12 for advice.

You will get a similar message if the `/dzero` area is over 95% used; this will also halt processing. See Section 2.11 for advice on that problem.

1.2.13. Rsh got stuck in _______ ; killed

This means that a remote command got stuck and was cancelled. A few of these are no problem; if you get many, it may signal a serious network problem. Try to **`rlogin`** to the node(s) that sent the message. If you can't do that, call the operator and ask to have the nodes checked (are they on?). Again, if you think there is a serious problem, call in the expert.

1.2.14. Check wrkshell on fnsfxxx

```
Date:
From: dzero@fnsfe.fnal.gov ( Dzero)
To: dzero@d0fs
Subject: Check wrkshell on fnsf128
```

```
More than one wrkshell on fnsf128
```

This message means that there are multiple `wrkshell` processes running on the worker node. Usually, when the problem is detected, the system cleans itself up and then sends the message —but sometimes this fails. You should check by logging into the node in question and doing **`ps -ef | grep dzero`** and killing whatever daughter processes exist, i.e., every `wrkshell`, `d0reco.x`, `inreader.x`, and `outwriter.x`.

1.2.15. RECO hung

This means that `d0reco.x` has been hung for over 30 seconds on the specified node. You should restart RECO on that node (see section 2.5).

1.2.16. Multiple RECOs running on node fnsfXXX

This means that too many processes got started on this node. You need to kill every `inreader.x`, `outreader.x`, and `d0reco.x` process you see on this node. (See section 2.5.)

1.2.17. Many others for which no action is required, including:

No files in raw
WNxxxx returned to blanks
dbl3 server failures
Bad Zebra construct
Job control failed

If you see these, you should just archive them in the `FARM_YYMMDD` folder on D0FS::DZERO. [FYI: at present, the "returned to blanks" message is fictitious. Tapes flagged for returning to blanks are instead listed in a separate file (`blanks_to_be.list`) in the resource directory, so that our tape managers can examine them to diagnose our problems and prevent overwriting of tapes.]

1.3. Log files

The farm produces a number of log files which can be helpful in tracking down problems or determining the status of particular jobs. The log files are kept in subdirectories of the areas `~/proman/pdbkd`, `~/proman/logdb`, and `/proman/wrkdb`; the subdirectories have the format `year/month/6-day`. (Go to one of these areas and do **`ls`**; you'll understand the format.)

It is to your advantage as a shifter to be familiar with the contents of various log files, so you should look in these directories on your own to learn the details. Here, I provide only some general guidance as to the kinds of information you can find in these files.

1.3.1. logdb

These are the kinds of files you will see in this area:

```
sfe~/proman/logdb/1994/mar/19-24% ls

PRL_RECO_FULL_V12_fnsfe_2_4032112.log          inspool_12522.log
PRL_RECO_FULL_V12_fnsfe_2_4032207.log          inspool_12523.log
copy_results_log.WN7726                        inspool_12524.log
copy_results_log.WN7727                        inspool_12525.log
```

```
copy_results_log.WN7728            outspool_WN7726_4031920.log
inspool_12512.log                  outspool_WN7727_4032200.log
inspool_12513.log                  outspool_WN7728_4032202.log
inspool_12518.log                  outspool_WN7729_4032203.log
inspool_12520.log                  outspool_WN7730_4032209.log
inspool_12521.log                  outspool_WN7731_4032210.log
sfe~/proman/logdb/1994/mar/19-24%
```

Here is what some of them contain:

-`PRL_RECO_FULL_V12_<VM>_<date>.log` records the activities of the `d0reco_prl_fnsfX_#` process; a new file is created every time this process is started.

-`inspool_<job_id>.log` records the copying of files from the input tape to the inspool directory. This job ID corresponds to the ID shown in `tpm_disp` (see section 1.1.1).

-`Copy_results_log.<tapename>` confirms copying of DSTs and RCPs to D0FS.

-`Outspool_<tape>_<date>.log` records the copying of output STAs to tape <tape>.

1.3.2. pdbkd

These directories contain the RCP files for the production database. The `STAF_<filename>.RCP` and `DSTF_<filename>.RCP` files (one for each run partition) contain input and output file names, when and where RECO ran, the number of events processed and missed, and a few other things. The `STAT_<tape>.RCP` files list the files on each tape and which tapes the corresponding raw data files can be found on. They also note when and on which VM the job ran, as well as on which tape drive the output tape was written. You will find these files helpful in a number of different situations, particularly when you want to know which tape a given output file was on.

1.3.3. wrkdb

This directory is not on the home disk. It can be found by typing **`cd /proman/wrkdb.`** (Note that there is no ~ in this directory name.) The subdirectories contain the D0RECO log files for each worker node; the logfile is placed in the subdirectory corresponding to the most recent date when `wrkshell` was *started* on the node, which does not usually correspond to the date a particular file was processed. (To find the most recent starting date, **`rlogin`** to a worker node and type **`pf wrk`**. You will see the date this process was last started. Then log out and look in the subdirectory corresponding to that date. If **`pf wrk`** returns a time rather than a date, it means that `wrkshell` was started within the last 24 hours.)

Looking at the logfiles is useful when you suspect RECO is crashing or hanging; you may be able to determine where the problem is and give some advice to the RECO experts who are responsible for fixing problems in the code.

1.4. The Pager

The pager should be carried by one of the farm shifters, mainly so that the operators can reach you when they need to notify you of conditions calling for your attention.

When the operators page you, respond as soon as you reasonably can.

If you see “LO CELL” in the pager window, it is time to change the battery. New batteries may be in a supply cabinet (ask Sonya Wright), or you may procure one from the stockroom if you are authorized to do so (ask Lee Lueking what budget code to use).

2) PROBLEM FIXING

2.1. Tape drive fails

This happens to most drives after they have been operating for a few months. A drive is automatically marked `UNAVAILABLE` (bad) in `drives.fnsfX` when its error count reaches 5. Other signs of a failing drive include: ABEND errors (mount failures) on 3 or more consecutive attempts to mount a tape on that drive, a hung inspooler or outspooler process, or too many files in `~/proman/sta` area. (This last happens when a drive is switched to single density. Call the operators and have them check that.)

When a drive goes bad, the first thing to try is having the drive cleaned. Call the operators in Feynman and ask them to do it. When this is done, reset the error count to zero and status to `READY` in `drives.fnsfX`.

If the drive still has problems, you should do the following:

1. Identify the problematic drive from email messages and/or the `drives.fnsfX` file. From whichever I/O node the drive is on, execute **`cps_umaint tape sgi## broken noswap.`** (Note that **`##`** represents the 2-digit number that is part of the drive name.) When prompted, enter a comment explaining why you think the drive is broken (use `<Ctrl-D>` when done) or create a comment file and select that; try to include enough detail to help the maintenance personnel track down the problem. After you have entered your comment, the tape drive state will be updated to broken in the cps database (when you do **`cps_tape -lt`**, you will see that it has changed) and mail will be sent to farm-admin@fnsg01.fnal.gov and to farm-user-fnsfe@fnsg01 announcing the update.

2. Recombine the inspoolers, if needed (see Section 2.2).

3. Call operator for a report. The production names for D0 are **sgid0sfe** and **sgid0sfd**.

FYI, the person who actually takes care of repairing and replacing bad drives and other hardware is Ken Stox, stox@dcdkc.fnal.gov. It is generally a good idea not to contact him directly; use farm-admin. Also, you should send to farm-users-d0@fnsgi1 a copy of any message to farm-admin.

When a drive is repaired, you will receive a mail message telling you that its status was updated to working. After receiving this message, you should do **`cps_tape -lt`** to make sure the drive is listed in the database as working. If it is, check the `drives.fnsfX` file (in the resource directory) to make sure that its status has been set to `READY` and its current error count to zero. If these have not been done, you should edit the file to make the changes. Then, you can change the inspooler configuration to reflect the new count of available drives.

2.2. Combining inspoolers

This needs to be done whenever fewer than 6 tape drives are available on a physical machine (fnsfe, fnsfd, or fnsff). Since each machine has 7 drives, you do not need to combine inspoolers unless 2 or more drives are broken.

The principle is that two VMs assigned to the same project can share an inspooler (and therefore a tape drive), but each VM must have a separate drive for its outspooler. The following rules for combining inspoolers follow this principle:

- 6 or 7 drives available: Each VM has its own inspooler.
- 5 drives available (2 bad): Two inspoolers must share a drive.
- 4 drives available (3 bad): One VM should be set to `UNAVAILABLE` (you can have all three VMs sharing an inspooler, but this is less efficient).
- 3 drives available (4 bad): One VM must be set to unavailable and the remaining two share an inspooler.
- 2 drives available (5 bad): Two VMs must be set to UNAVAILABLE.
- Fewer than 2 working drives: Data processing is not possible.

In order to combine the inspoolers, find two VMs on the same physical machine that are running the same project. If all three are running different projects, you will have to change the project for one VM in the `resource` file so that two of them are running the same project. When you have decided which two VMs will share an inspooler, go to the `~/proman/resources` directory (on fnsfe) and look at the `inspool_list_N.fnsfX` files there. You want to find one that combines the inspool areas of the two VMs you have decided to combine. If there isn't one, edit one of them (e.g., N=8 or N=9 for `fnsfe`) to match your choice. Then edit the `resource` file, changing the project (if needed) and changing the inspooler number to match N for the two VMs whose inspoolers you are combining.

Now you will need to get the proper inspoolers started. (If you skip this step, the system will eventually sort itself out, but that wastes a lot of time and generates an annoyingly large number of email messages.) Here are the steps to take: First, kill the inspooler processes on the VMs you have just combined (using **`kill -9`** `<pid>`). Then cancel the tape mount(s) requested by these inspoolers: Figure out which tape(s) have been mounted on which drive(s) for the inspoolers you have combined (**`pf inspool`** will help you here). Then type **`cps_tapereply -f -c"fnsfe" sgi##`**. You will then have to log in to D0FS::DZERO and release the tape from the batch queue (see section 2.4). You should also do these things when you un-combine inspoolers after drives have been repaired.

Here is an example of how to tell which tapes have been requested by which inspoolers:

```
% pf inspool
dzero 18798 1 41 19:09:48 ?    3:52 /usr/local/home/dzero/proman
/exe/inspool_fnsfe_5  -inl WM5915 -dev sgi85 -queid
dzero 19462 1  0 19:13:53 ?    1:29 /usr/local/home/dzero/proman
/exe/inspool_fnsfe_8  -inl WM6248 -dev sgi86 -queid
%
```

This command shows you that inspoolers 5 and 8 are running on fnsfe, and that inspooler 5 has a tape on sgi85; inspooler 8 has a tape on sgi86. (Incidentally, for this example, `fnsfe_0` and `fnsfe_2` were sharing `inspool_fnsfe_8`.) You can check the status of the mounts by typing

```
% cps_tape -lt
TAPEDRIVE DEVTYPE      STATUS  ALLOC       TAPE   TAPE_STATUS
sgi80     exabyte_8500 working allocated   WN7875 mounted
sgi81     exabyte_8500 working allocated   WN7876 mounted
sgi82     exabyte_8500 broken unallocated  WN7849 mounted
sgi83     exabyte_8200 broken unallocated  WN7877 mounted
sgi84     exabyte_8500 working allocated   WN7880 mounted
sgi85     exabyte_8500 working allocated   WM5915 mounted
sgi86     exabyte_8500 working allocated   WM6248 mounted
%
```

To cancel the mount for inspooler 5, you would type **cps_tapereply -f -c"fnsfe" sgi85**.

2.3. Releasing tapes held on D0FS

You need to do this when you receive a “Tape in use on D0FS” message, or when you have killed an inspooler.

First, make sure the tape isn't being held for a legitimate reason. (Occasionally, someone else has cause to use the raw data tapes.) If it isn't, log into D0FS and look at the queue STAGE_IN_USE_ON_UNIX_FARM and find the entry number of the job that is holding the tape you need. (The job name will be WMxxxx_WMxxxx_XX_UNIX.) Stop that job using **DELETE/ENTRY**. Back on fnsfe, change the status of that tape from SUBMITTED to WAITING in injobs.

Note: you can try logging into specific D0FS nodes, like D0RSEX, D0TSEX, D0RSUT, etc. if login to D0FS is denied due to “too many users.”

2.4. Marking tapes BAD

You should do this when you notice that a tape has accumulated a large number of error counts in the injobs file. Edit injobs, changing the status of the tape from WAITING to BADTAPE. Send a message to D0FS::COPYMAN identifying the bad tape; include the tapes_moved file for that tape. You should also rename the tapes_moved file for that tape in ~/pro-man/history to something like bad_tapes_moved.WMxxxx to make sure the automatic procedure doesn't resubmit it. Do not attempt to process the tape again unless told to do so by an expert.

2.5. Injobs has nothing WAITING

To make a long story short, this should not happen during the run, unless there has been a lengthy shutdown of the accelerator. If you see only a few or no tapes WAITING in injobs, you should check whether tapes are being vaulted and entered in the catalog properly.

Check by logging into D0FS::DZERO and looking in the directory USR$DISK51:-[FMD0.MOVEDTAPES]. There you should see files called TAPES_MOVED.<date>, which list all the files on all tapes vaulted since the last file was created. You should check for recent dates on these files to verify that vaulting and cataloguing hasn't stopped.

Next, on fnsfe, look in the history area (**`his`** or **`cd ~/proman/history`** gets you there) for `itapes_moved.<date>` files; there should be one corresponding to each file you found on D0FS. Look at the most recent of these files to see the tape labels.

Then, look at the `injobs` file to check if the tape is in the queue. If you see a problem, send mail to D0FS::COPYMAN and to Shuichi Kunori so that they can investigate. They will advise you if tapes need to be added manually.

2.6. DST does not get to D0FS

The normal chain of events after a file is processed calls for the STA to be written to tape and the DST to be copied to the `D0$DATA$BUFFER` area on D0FS, along with an RCP file indicating where the DST in question really belongs. A server process on D0FS looks for these RCP files, reads them, copies the relevant DST to the correct area, then deletes the DST from `D0$DATA$BUFFER`.

So if the DST does not show up in `D0$DATA$DST` (or whatever "target" directory area it should end up in), the first thing to do is check the buffer area. If the DST is still there, check that the RCP file is also there and okay. If so, make sure the target area has space in it. When a target area is full, inform Lee Lueking that files are backing up in the buffer; he is responsible for deciding how to create space.

If the DST hasn't reached the buffer area, it may be that the buffer area is full. (If not, see section 1.1.8.) In this case, it's probable that a target area is full and has caused the backup into the buffer; contact Lee Lueking to get things sorted out.

The next thing to look for is a network problem. These are common, especially when D0FS goes down. Make sure D0FS is alive, then log into fnsfe and kill any `uptest` process you see that is over a day old. This should prod some file transfers into starting.

If there are no stale `uptest` processes, you can try **`pick_up_all`** to start the transfer, or use zftp to transfer the file manually. (In the latter case, you will have to find the DST file on the farm and rename it using lowercase letters before the transfer will work.)

Note that if the DST is not copied, the STA (which has been written to tape) will not be catalogued. This may cause files to be reprocessed, generating duplicates. A process is run to check for this; in time, the farmers may be made responsible for removing the unwanted copy of any duplicate files, but this has not yet been implemented.

2.7. Killing/Restarting a parallel server

You should do this only when a particular VM has been frozen (no files growing) for a long time, or when combining inspoolers. If this is the case, see if the `d0reco_prl_fnsfX_#` process is running. (Use **`pf prl`** to confirm that the process is running for the "hung" VM and to get the process ID.) If the process is running but the VM has been hung for a sufficiently long time, the following steps will kill the process and induce a restart:

First, be sure you are on fnsfe and that `tpm_submit_job` is sleeping. (You can check this by typing **`ps -ef | grep sleep`** and looking for `sleep 800` among the processes. If you don't see `sleep 800`, check that `tpm_submit_job` is running; continue checking for `sleep 800` until you see it.)

To kill the process, edit the resource file, changing the status of the VM whose server you intend to kill to `UNAVAILABLE`. Then, from the I/O node on which the server in question is running, type **`kill -9`** `<pid>`. If you want it to finish its present activity and end gracefully, or if you intend for an immediate restart, do nothing else. However, if you want it to die quickly and not restart at once, you should also kill the `inspool`, `outspool`, `insrv` and `outsrv` processes for that VM if the next "wake" cycle of `tpm_submit_job` doesn't do it. Then, cancel the mounts (see example in section 2.2) and release the tapes from D0FS (section 2.3).

To restart the VM, edit the resource file again (on fnsfe, while `tpm_submit_job` is sleeping), changing the status back to `FARMLET.` The process will be automatically restarted when `tpm_submit_job` wakes up, so after half an hour or so, you should check the area again and make sure all is well.

2.8. Multiple jobs INQUEUED on a VM

This has happened when RECO crashes. You should check out the `wrkdb` logfiles (see section 1.3.3) and notify a RECO expert. If the problem is run-specific, you can set the tapes for that run to `PENDING` in `injobs`; the VM will go on to a different run. When the RECO problem is fixed, be sure to set the relevant tapes back to `WAITING` so they will be processed.

If there are multiple jobs `INQUEUED` on a VM, but no evidence of a RECO crash, you should do the following to clear up the tapes: First, for all but the most recently `INQUEUED` job on the VM in question, check with **`tpm_disp`** to make sure that the tapes have not been resubmitted and processed since then. If they have, do nothing about them. But if they have not (and if it has been over 48 hours since the first submission of a WMxxxx tape), check the status of the tapes in `injobs`; if it is `SUBMITTED`, change it back to waiting and remove it from the queue on D0FS (See Section 2.4) if it is sitting there.

2.9. Restarting D0RECO on a worker node

First, see if RECO really is dead by doing **`rlogin`** to the worker node and doing **`ps -ef | grep dzero`**. You will see `d0reco.x` running and the elapsed CPU time. Repeat this command after 30 seconds and see whether the CPU time increases. If not, kill `inreader.x`, `outwriter.x`, and `d0reco.x`. (If you have another reason to be restarting RECO, you don't need to check whether the CPU time is increasing.)

Kill the processes with **`kill -9`** `<pid>`. They should restart automatically, but it is not a bad idea to check the worker node in question to make sure all is well.

2.10. Problem runs

This is not very common, but you should know what to do, just in case. Some runs cause RECO to crash on every event. The first symptom is that a particular VM stays hung (usually with "total 0") for a long time. Look in the `inspool/` and `raw/` areas (subdirectories in the spooling area); you will see files piling up if RECO is having problems. Next, check one of the worker nodes for that VM to see if `d0reco.x` is running. If RECO is crashing, you will see the process appear and disappear when you do **`pf d0reco`** repeatedly. You should then look at a log file for one of the worker nodes (these are in the `wrkdb` area; see section 1.3.3) to see what problems RECO is having. Send mail to the appropriate experts describing the problem.

If the problem is with the databases (DBL3), it will affect entire runs. It is a good idea to find all the raw data tapes corresponding to the affected run(s) and set them to `PENDING` in `injobs` until an expert informs you that the problem has been fixed. (The quickest way to get all the tape numbers corresponding to a particular run is to call the DAQ expert on shift in the control room and ask; they should know how to do this.)

2.11. Disk space shortage on /dzero

When the free disk space on `/dzero` falls below 5%, all processing halts. Contact an expert (S. Kunori) for advice on cleaning up the disk.

2.12. Disk space shortage in spooling area

When one of the spooling disks is 90% (or more) in use, processing on that VM will eventually grind to a halt. This problem is often a byproduct of unusually large STAs or outspool failures. When the disk space runs low, you will get mail messages. Check that the outspooler for the affected VM is running and that its tape has been mounted. Also, look at outtape ABEND messages for indications that the drive being used by the outspooler is failing. (See section 2.1 for further instructions on tape problems.) If the tape and drive are okay and the problem persists, you can try executing **`pick_up_all`** from the `~/proman/exe` directory. This usually frees up enough space to get the disk use below 90%.

Note: *RESIST* the temptation to rm files from the dst and sta directories; this can cause confusion in the databases and on D0FS. You can try deleting the files from the inspool and raw areas; that may free up enough space to get things going.

3) SUBMITTING SPECIAL JOBS

Do this when you are instructed to do so by Lee Lueking or someone else with authority from OCPB. (Remember that all special jobs requests must be approved by OCPB before they are run.) You will be given a form describing each approved request, which you should keep until the job is completed, then return to Lee. This section describes the steps to take to process the data.

Note regarding regular RECO jobs: This procedure has been automated for WMxxxx tapes (raw D0 data) to be processed with the current version of D0RECO. Automated processes check for new WM tapes in the log of vaulted tapes kept by the operators at Feynman; these tapes are automatically entered in the FATMEN catalog and a `tapes_moved.WMxxxx` file for the tape is generated. The tape is also appended to the end of the `injobs` file. So if someone requests a WM tape to be processed at high priority, all you need to do is to move it to the top of `injobs`.

3.1. Get tapes_moved file

The person requesting the special job is responsible for providing a `tapes_moved.<name>` file for each tape he or she wants processed; the request form should tell you where these file(s) are. You should copy the file(s) requested to the history area. So, go to directory `~/proman/history` (use shorthand command **`his`** if you like) and copy the files from wherever the form says they are.

The `tapes_moved.<name>` file should look like this:

```
sfe~/proman/history% more tapes_moved.VGL063
VGL063 1 TOP_W2E80A03N_VB300_G314SS3_A_07.X_RAW01
VGL063 2 TOP_W2E80A02N_VB300_G314SS3_A_04.X_RAW01
VGL063 3 TOP_W2E80A02N_VB300_G314SS3_A_06.X_RAW01
VGL063 4 TOP_W2E80A02N_VB300_G314SS3_A_08.X_RAW01
VGL063 5 TOP_W2E80A03N_VB300_G314SS3_A_02.X_RAW01
VGL063 6 TOP_W2E80A03N_VB300_G314SS3_A_04.X_RAW01
VGL063 7 TOP_W2E80A03N_VB300_G314SS3_A_08.X_RAW01
sfe~/proman/history%
```

Sometimes the file is empty. This is a way of telling the farm to process every file on the tape. Empty or not, the file does need to be present or the job will not run. Occasionally, someone who wants every file on a tape will not make a `tapes_moved` file, in which case you may create an empty file for that job after checking with the requester.

3.2. Add the request to the injobs file

Go to directory `~/proman/resources` (use shorthand command **`res`** if you like). You can either add the new tape request to `injobs` manually, by editing the file, or use the automated command procedure to do it for you.

For the automated command procedure, type:

`% injob_create`

You will be prompted for the project name, which you should enter in all capitals (remember that UNIX is case-sensitive), and for the tapes to use, which you should enter one tape name per line (two consecutive <CR> signal the end of the list).

3.3. Edit the resource file

In this same directory, you must edit the resource file to change one VM over to run the special project. Be sure the VM you switch is in `FARMLET` status and not `UNAVAILABLE`; otherwise the job will never run. If a VM is already assigned to this special project, you do not need to assign a second one unless the request is urgent.

3.4. What happens next

When the VM you have assigned to the special project finishes its current job, it will pick up the first tape in the `injobs` file for which the special project has been requested. It will try to run the job.

If the job fails the first time, you should edit the injobs file, changing the status of the relevant tapes from `SUBMITTED` to `WAITING` so that the VM will try again. As with any job, repeated failures may signify that the tape is bad. Do not switch the VM back to `RECO_FULL_Vnn` until the job has succeeded or it is decided that the tape is bad.

If the job goes through (gets to `ENDED` status in **`tpm_disp`**), you should trace the output to confirm that processing was completed. Section 3.5 describes one method of doing this.

When the job has finished (all tapes processed or declared bad), send an email message to the requester (cc to Lee Lueking, OCPB chairman), saying that the job has finished or indicating why it could not be completed (bad tape, etc.). Also, write the job status, date, and your initials at the bottom of the request form and return it to Lee. It is up to the requester (not you) to check the quality of the output and inform the OCPB of any problems. You should only redo a job if Lee (or another OCPB authority) authorizes it.

Be sure to change the VM back to `RECO_FULL_Vnn` in the `resource` file when the job is finished. This will guarantee best use of our available CPU and I/O capacity.

3.5. Tracing a tape

When you run a special job, you should confirm that the output does indeed get through the farm and onto D0FS and/or FATMEN before you declare it finished. If you know that the job is `ENDED` according to **`tpm_disp`**, you can do the following to verify the output:

- Note the date of processing from **`tpm_disp`**.

- Go to the appropriate subdirectory of `~/proman/pdbkd` (see section 1.3.2).

- Type: **`grep <INVSN> STAT*.RCP`**, where `<INVSN>` is the name of the input tape you are tracing. The name of the output tape is given in the name of this RCP file. Check that all the files are listed. It is also a sensible idea to type out the STAT RCP files you find with **`grep`** to make sure the format is correct. If no matches are found, you can assume that the output file was not generated. (Check the VM the input tape ran on to see if it is still working on these files. If so, give it more time. Also, if the processing date is near a 6-day subdirectory's boundary, you may also want to check the next subdirectory of `pdbkd`.)

- Go to the `~/proman/report/log` area and look at the `copied.OUTVSN` and `copy_log.OUTVSN` files, where `OUTVSN` is the output tape label you found in the previous step. You can use these files to confirm that every file on the tape was transferred. There are also logfiles for the copying of DST and STA RCP files that you may want to check.

- The final check is to look on D0FS (directory `D0_DATA_MC` for Monte Carlo) or FATMEN (use the file names you found in the STAT RCP files to determine the generic name) to confirm that the file showed up there. It may take a day or so for things to be catalogued, so do allow some time before doing this test.

For more on tracing tapes, see the most recent version of:
`D0FS::USR$ROOT5:[DZERO.DOCS]-TAPE_TRACE(date).DOC`

4) MAKING WEEKLY REPORTS

In addition to the duties outlined above, several reports on farm progress are expected from the shifters on a regular basis. You should assemble these reports **by 1 p.m. Monday** so that they can be passed on to the person reporting for D0 at the all-experimenters' meeting. (Since the shift changes occur on Mondays, it is the responsibility of those whose shift is *ending* to make these reports for the final week of their shift.) The shifter(s) who prepared the report should present it at the production group meeting on Tuesday morning and post the text report to the PRODUCTION folder of D0NEWS. Finally, the shifters going to the farm users' meeting, every other Wednesday afternoon, should take the reports along as they will help in recalling what problems have occurred. The report should include the following elements:

4.1. Number of events processed (graph)

The graph shows the daily progress of the farm over an interval of several months by showing, for every second day, the cumulative number of events processed between the start date (15 Jan 1994) and that day. The graph also gives the total number of events processed and the average number of events per day. Separate totals should be calculated for ALL stream data, special runs processed on the farm, and Monte Carlo; a combined total should be presented as well. The separate totals should be reported at the D0 production meeting on Tuesdays, while at the all-experimenters' meeting, only the combined total is needed.

The procedure to generate the plots using the weekly report is as follows:

1) The script `official_report` is submitted by the farm automatically every Sunday evening. It generates a file called `~/proman/report/log/off_rep.mm-dd-yy`, where the date the report was generated is used in the name.

2) On Monday morning, check that this file exists, is readable, and has output all the way up to the day before. Then execute:

```
% nawk -f report.awk off_rep.mm_dd_yy > report.mm_dd_yy
```

This will generate a file called `report.mm_dd_yy`, which gives the total number of events processed each day.

3) Ftp the report file you generated in step 2 to the `[.REPORT]` subdirectory on the D0FS::DZERO account. Then you will need to do:

```
$ COPY <JAN15_LASTWEEK.DAT> <JAN15_THISWEEK.DAT>
```

where the first file is last week's report and the second file is the updated report to include the most recent week. (If you look in the `[.REPORT]` directory, this should become more clear.) Then,

```
$ APPEND REPORT.MM_DD_YY <JAN15_THISWEEK.DAT>
```

where the REPORT file is the one you just ftp'd and the second file is the one you just created with the COPY command. Edit the `<JAN15_THISWEEK.DAT>` file to remove the overlapped days (everything between the beginning of the month and the end of last week). When you edit, notice that Sunday's total in last week's report was incomplete; you should use the one you got in the new report.

4) Use the REPORT.FOR program sitting in the D0FS area to generate a cumulative total for every second day. (I have found file protection violations when trying to use the FORTRAN compiler on D0FS, so you may need to copy all the relevant stuff to your own account to do this. Remember that you must leave copies of the data files on D0FS each week so that other shifters can find them.) Compile, link, and run the program. You will be prompted for a file name: give your input file name (the edited .DAT file from step 3). The output file is called TOTALS_OUT.DAT.

5) Run PAWX11. [I suspect this isn't set up on D0FS either, so you will have to run it from your own account.] You want to execute the macro TOTAL.KUMAC (copy it from D0FS to somewhere you can run PAW) to generate the plot. The macro will create a postscript version of the plot for you, which you can print. You will need one copy for the D0 representative at the all-experimenters' meeting, and one for you to show at the production meeting.

Note: This procedure causes some information to be lost at the end of the month, due to the way the official_report script works. Until this is changed, you will need to run the script yourself (as described in section 3.5 of Kirill's manual) soon after the first of each month to get the information for the last few days of the previous month.

Due to directory-size limits in UNIX, the weekly report program is not functioning properly. In place of the above, you can generate the plot using daily reports. This also has the advantage that information is not lost at the end of the month. Here are the steps to follow:

1) Go to the ~/proman/report directory and execute **official_report_daily Mmm dd &**, where Mmm dd is the month (3 letters) and date for which you want a report. Do this for each day for which a report is needed (Monday–Sunday). It is a good idea to run some of these during the week. Each daily report job produces an output file, off_rep_daily.ddMmm1994, which has the same format as the weekly report file.

2) On D0FS, in the [.REPORT] subdirectory, copy <JAN15_LASTWEEK.DAT> to <JAN15_THISWEEK.DAT>. Edit the new file by typing in the total number of events in each daily report file you generated in Step 1.

3) Follow steps 4 and 5 from the weekly report.

4.2. The written report

This is the file PRODUCTION_REPORT.MM_DD_YY (the date is the Monday the report period begins) in the [.LOG] subdirectory of the D0FS::DZERO account. It should include the following elements:

4.2.1. Dates covered and shifters' names

Indicate also who was responsible on which days.

4.2.2. Weekly summary

Monday morning, write a few sentences to highlight major problems, changes, plans, etc. Remember that the report will be made public, so avoid negative commentary on individuals or their systems.

4.2.3. Tape drives

Note when tape drives went bad, were replaced, repaired, cleaned, whether inspoolers had to be combined due to shortage of drives. You should also include any drives that were reported bad during a previous week and fixed during the week your report covers.

4.2.4. Special projects

List all special requests for the week, along with a status report, making it clear whether the job is in the queue, in progress, finished, failed miserably (what is being done about that?), etc. If there were projects outstanding from the previous week, include them also.

4.2.5. Current node configuration

Monday morning, type the resource file and enter the project assigned to each VM. If a VM is unavailable, indicate that.

4.2.6. Number of tapes processed

Look at the official report output files (daily or weekly). Count the number of input tapes processed each day and add them up.

4.2.7. Last run number processed

Look through the official report output files again, and find the highest run number from the previous week's listings.

4.2.8. Tapes in injobs

This is a count of the backlog. To get the total, type

```
% res
% more injobs | grep -c WAITING
```

The response will be a number: the total number of tapes waiting. To get the breakdown by project, add another "pipe" to the UNIX command, e.g.,

```
% more injobs | grep WAITING | grep -c FULL
```

will give you the count of tapes waiting for full (standard) D0RECO. (You can do the piping in either order, but be sure that the **-c** is in the last part, otherwise you won't get a sensible response.) Simillar commands will get you the count for the other projects: the individual project counts should add up to the total. Put all this information in the report.

4.2.9. Tape ABENDS, Bad files

You should collect this information during the week from the e-mail messages. A good way to do this is to have two windows open to the D0FS::DZERO account when you are reading new mail there every day. Read mail in one window, and edit the weekly report file in the other window. Whenever you come to one of these messages, use the mouse to transfer the information to the report file (which is being edited at the time). Then move the message to the appropriate folder. If you do this daily, you will find it less tedious than sorting through a week's accumulation of farm mail on Monday in a frantic search for these messages.

4.2.10. Miscellaneous

Here you can make a note of various things that go wrong: system crashes, disk mounts lost, resource files that disappear, processes that get hung for days, past or upcoming system work, etc. Again, it is wise to avoid negative commentary. Simply state what happened and what (if anything) has been done to fix it.

4.3. Changing log directories

Because UNIX cannot handle directories with over 500 files, it is important to change the log directory from time to time. At present, this is done on the first of each month.

First, make sure that no copy jobs are running by doing:

`%` **`ps -ef | grep copy`**

This should show no jobs running.

Then rename (UNIX command **`mv`**) `~/proman/report/log` to `~/proman/report/logmmmyy`, where `mmm` is the three-letter abbreviation for the month (Jan, Feb, etc.) and `yy` is the year.

Finally, create a new `~/proman/report/log` directory.